

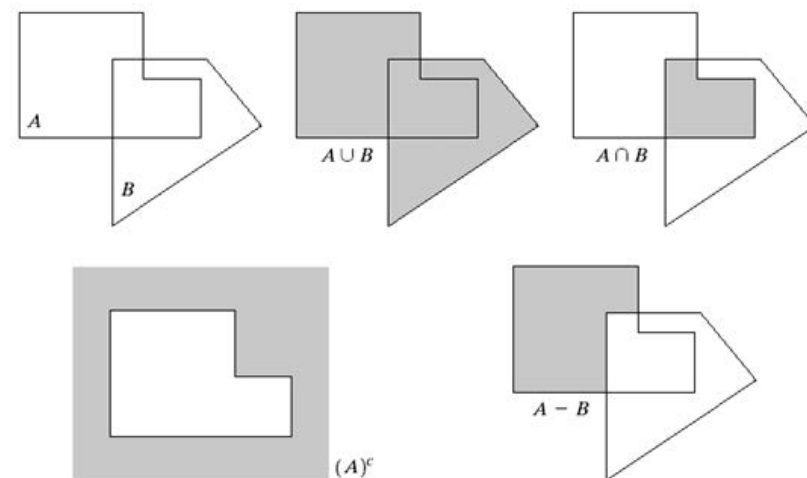
形态学图像处理

《数字图像处理》 第7讲

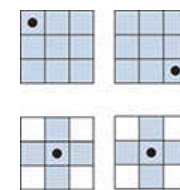
数学形态学

- 形态学(morphology)是研究动物和植物的形态和结构的学科
- 数学形态学基于集合论来调整或描述图像中物体的形状
- 集合表示图像中的对象
 - 在二值图像中，所有白色像素的坐标
 - 在灰度图像中，所有像素的坐标，及灰度值
- 作用
 - 图像预处理或后处理（滤波，简化形状）
 - 提取形状特征（抽取骨骼，细化，粗化，凸包，物体标记）

6种集合操作



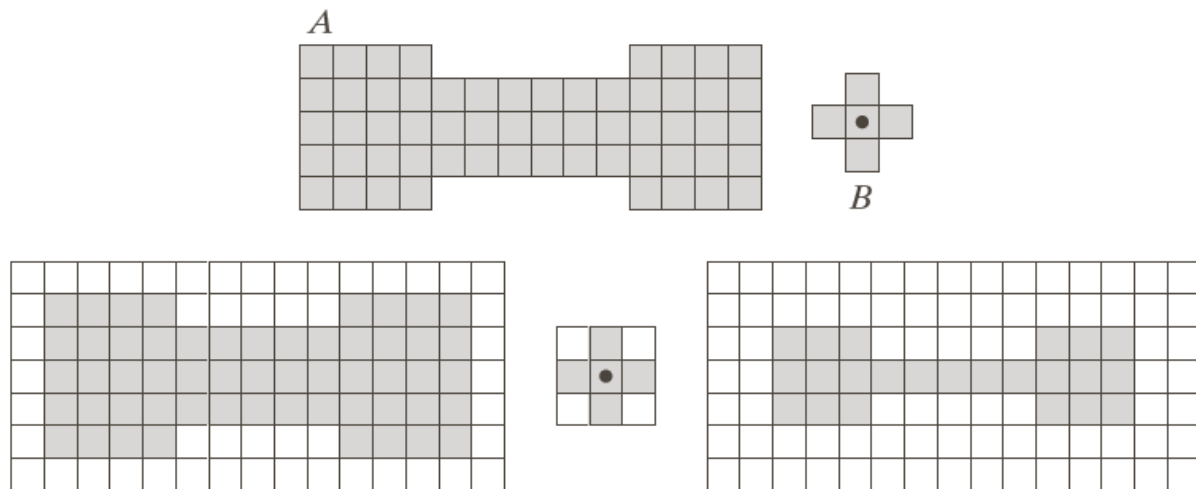
- 并集: $A \cup B = \{w \mid w \in A \text{ 或 } w \in B\}$
- 交集: $A \cap B = \{w \mid w \in A \text{ 且 } w \in B\}$
- 补集 $A^C = \{w \mid w \notin A\}$
- 差集: $A - B = \{w \mid w \in A \text{ 且 } w \notin B\} = A \cap B^c$
- 集合的转置: $\hat{B} = \{w \mid w = -b, b \in B\}$
- 集合的平移: 将集合B平移一个向量 $z = (z_1, z_2)$



$$(B)_z = \{b + z \mid b \in B\}$$

形态学操作

- 形态学操作定义在两类集合上
 - 目标元素，前景像素集合（周围一圈仅仅为了padding）
 - 结构元 structuring elements, SE，由前景和背景像素一起定义
 - 用来描述感兴趣的形状
- 操作举例
 - 结构元在图像上滑动
 - 当结构元完全包含于图像时
 - 结构元前景和图像前景重叠
 - 输出图像在原点位置标记为前景
 - 否则标记为背景



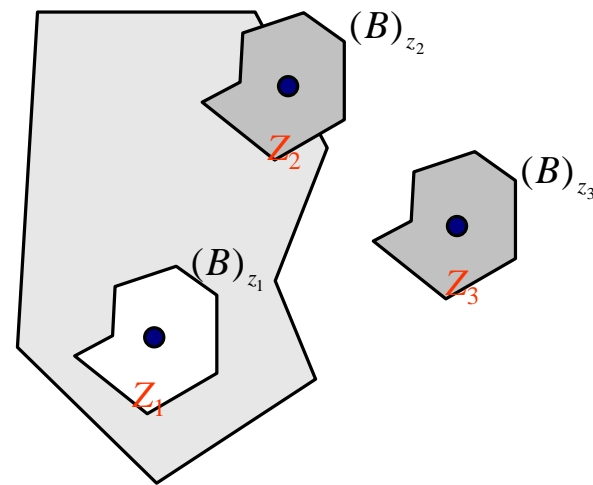
腐蚀操作 Erosion

- A和 B 是图像平面中的集合，A 被结构元素 B 腐蚀定义为

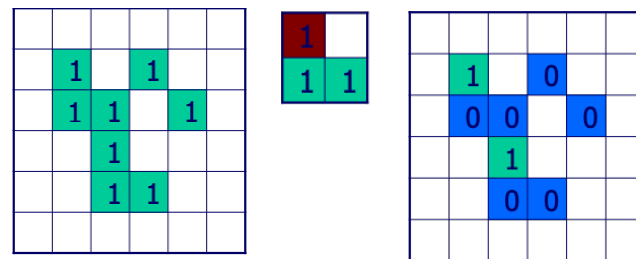
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

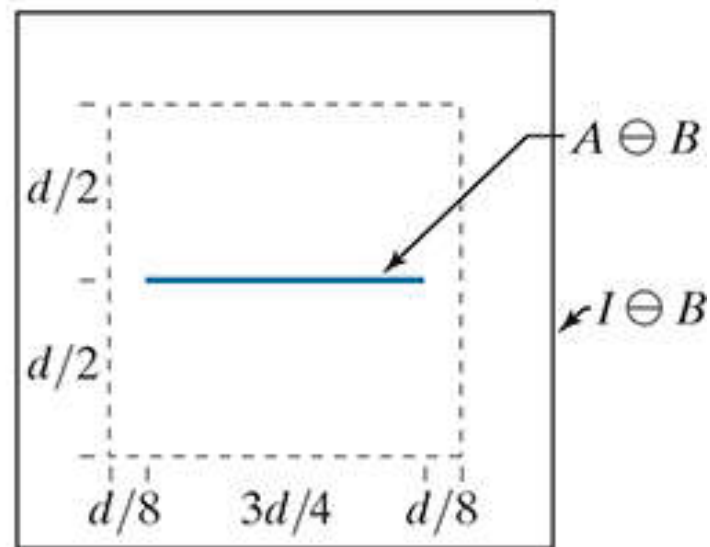
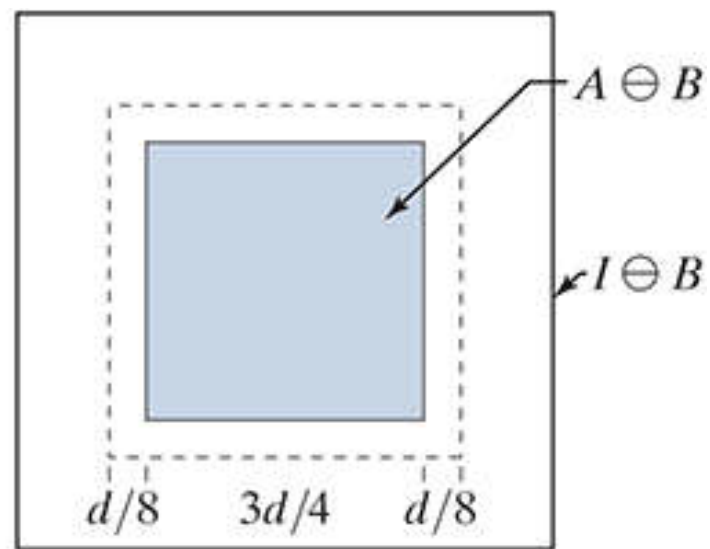
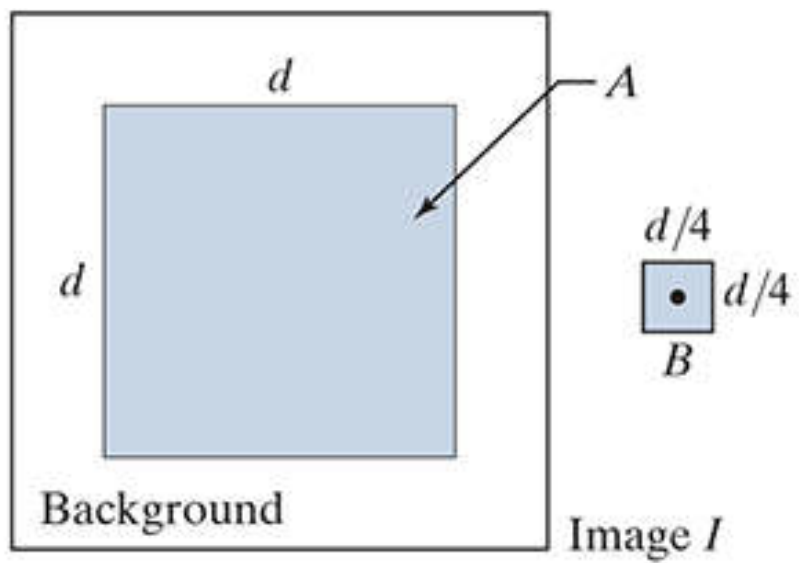
- 将 **B** 平移 z 后，**B** 仍在 **A** 中的所有 z 的集合
- 平移量 z 等于 **B** 的原点的位移（原点的新位置）

$$(B)_z = \{b + z \mid b \in B\}$$



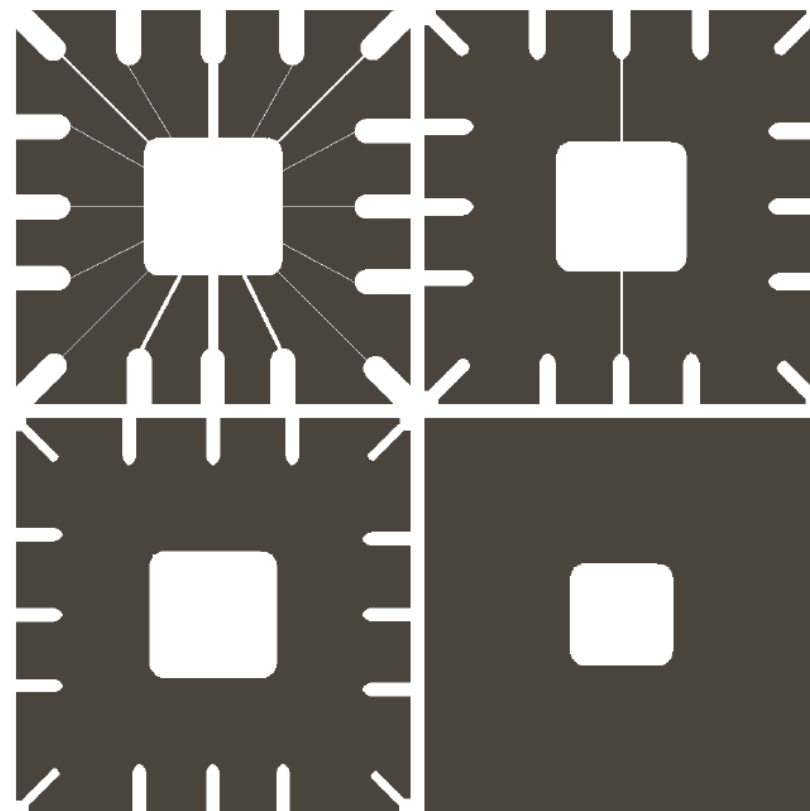
- 实际做法
 - 将 **B** 看作在集合上滑动的空间卷积核
 - 当滑动后的 **B** 完全属于 **A** 时
 - 将 **B** 的原点所在位置作为结果的前景输出





腐蚀操作 Erosion

- 486x486的焊线模板
- 11x11, 15x15, 45x45的结构元腐蚀
- 腐蚀缩小或细化了二值图像中的物体
- 可以将腐蚀看成是形态学滤波操作
- 将小于结构元的图像细节滤除(去除)



膨胀

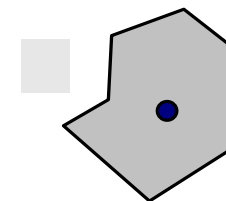
- Dilation

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

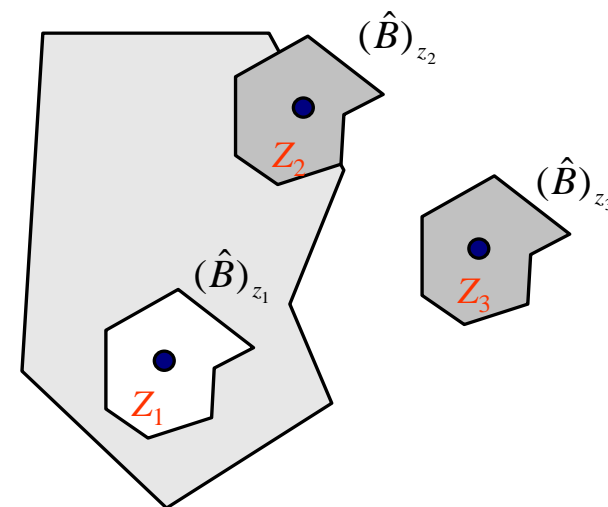
- 将 B 转置后平移到 z
- 与 A 至少有一个元素重叠的所有 z 的集合

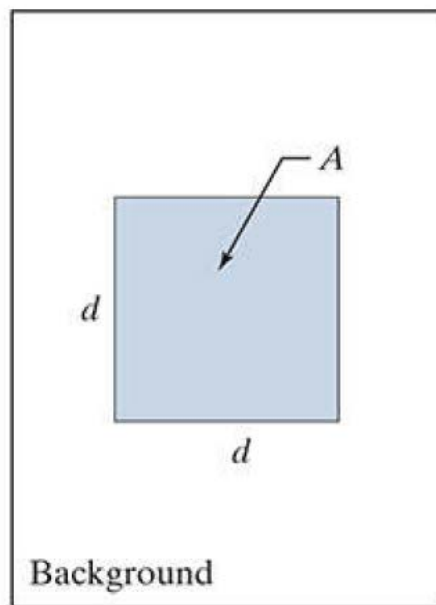
- 实际做法

- 将 B 以自身原点转置 \hat{B}
- 将 \hat{B} 看作在集合上滑动的空间卷积核
- 当滑动后的 \hat{B} 和 A 有交集
- 将 \hat{B} 的原点所在位置作为结果的前景输出

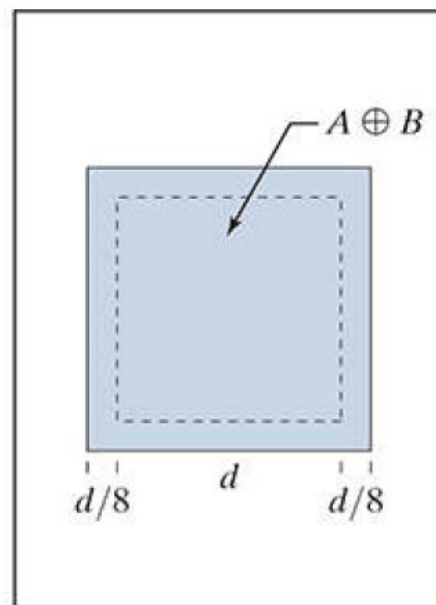
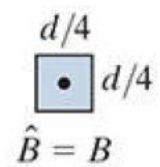


结构元素

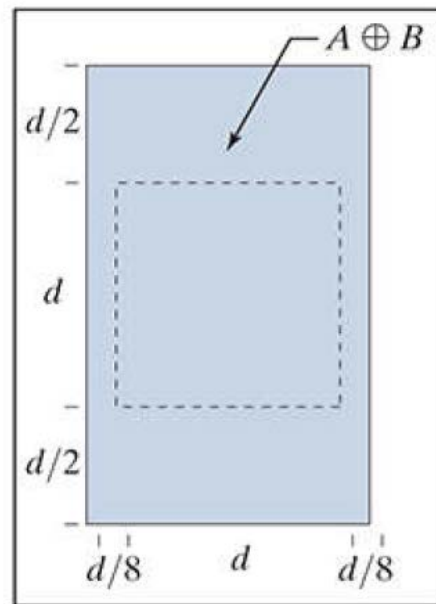
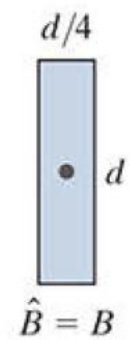




Image, I



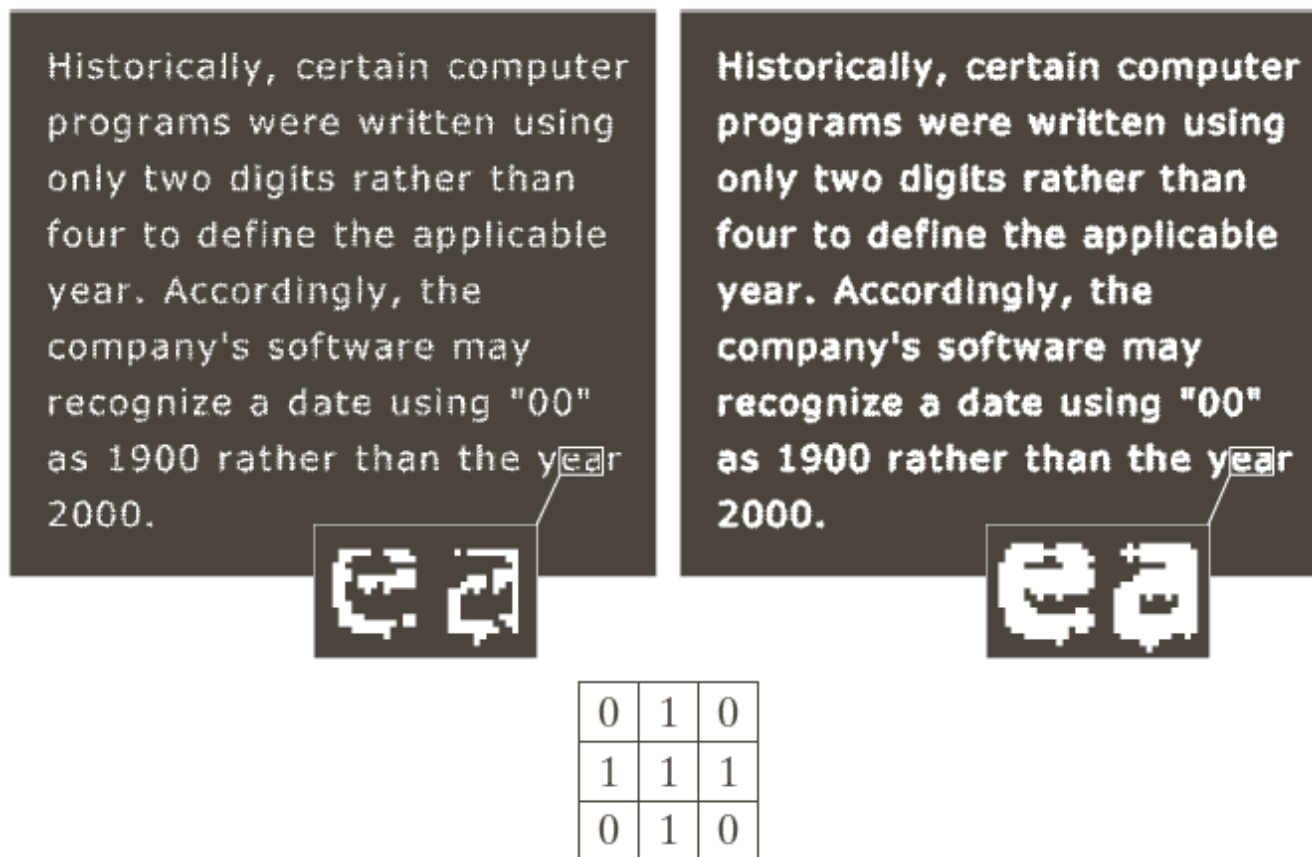
$I \oplus B$



$I \oplus B$

膨胀

- 膨胀则会“增长” 二值图像中的物体
- 裂缝桥接





小结

- 腐蚀使图像中物体区域缩小
 - 可以把小于结构元素的物体(毛刺、小凸起)去除
 - 选取不同大小的结构元素，可以去掉不同大小的物体
 - 如果两个物体之间有细小的连通，当结构元素足够大时，可以将两个物体分开
- 膨胀使图像中物体区域扩大
 - 将与物体接触的背景点合并到该物体中
 - 可用来填补物体中的空洞，或桥接小的裂缝

开运算(Opening)

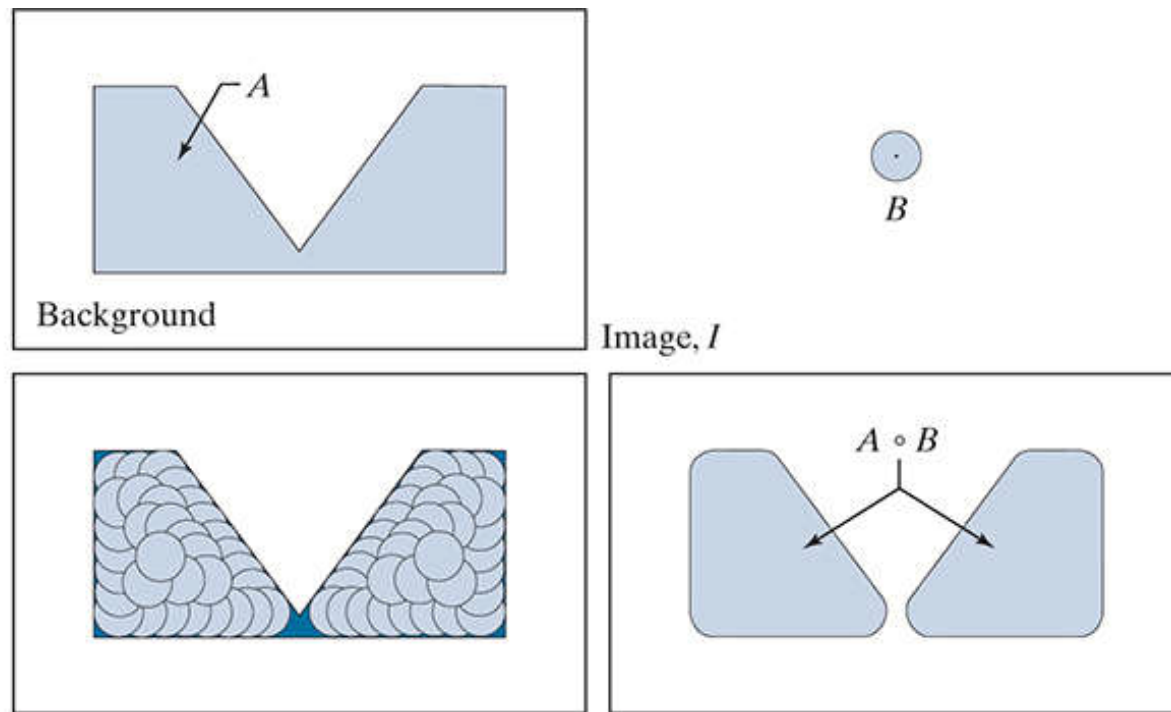
- 使用结构元素 B 对集合 A 进行开运算，表示为 $A \circ B$

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

- 先腐蚀，再膨胀
- 等价表示
 - 在 A 的内部移动 B ，所达到的区域

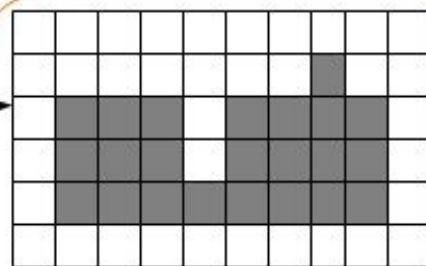
$$A \circ B = \cup \{ (B)_z \mid (B)_z \subseteq A \}$$

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

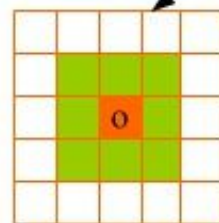




目标图像A，其中白色部分
代表背景，灰色代表目标X

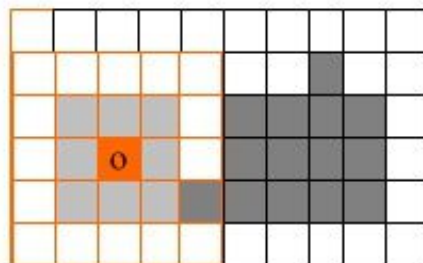


为结构元素B，其中原点
位置为橘黄色标注

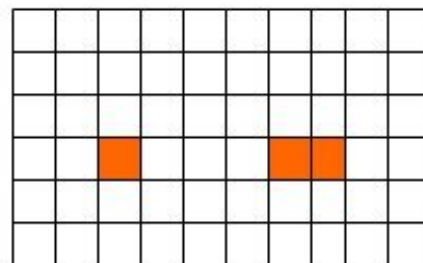


此过程为
开运算

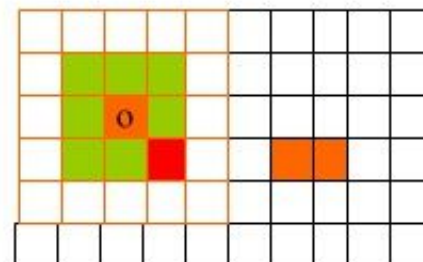
(1) 进行腐蚀操作：
进行遍历处理



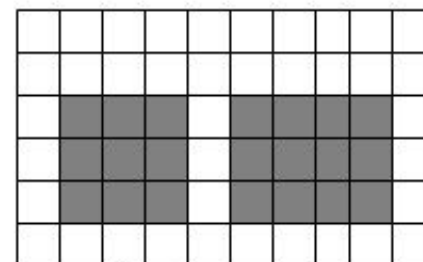
经过腐蚀后的结果



(2) 进行膨胀操作：
进行遍历处理



经过膨胀后的结果



闭运算(Closing)

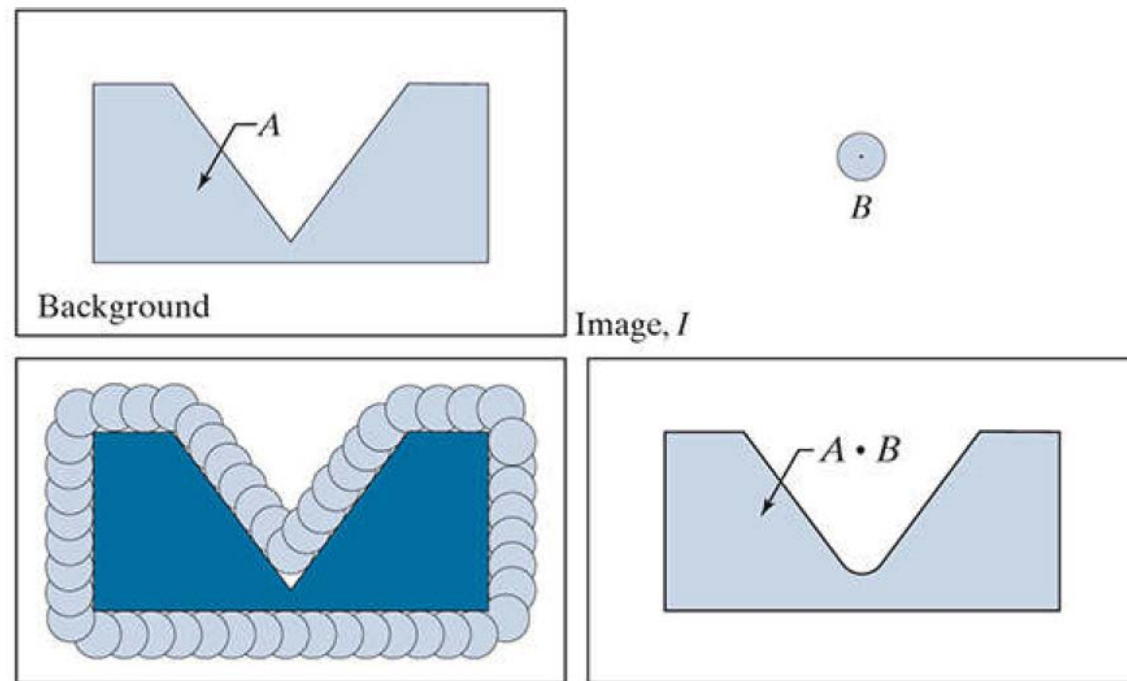
- 使用结构元素B对集合A进行闭运算，表示为 $A \bullet B$

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

- 先膨胀，再腐蚀
- 等价表示

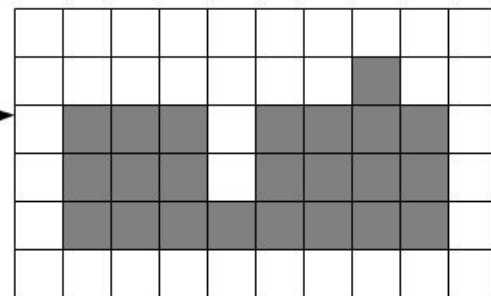
$$A \bullet B = [\cup \{(B)_z \mid (B)_z \cap A = \emptyset\}]^c$$

- 在A的外延移动B，保持不相交
- 对结果区域取补

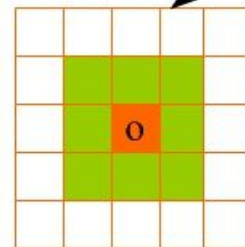




目标图像A，其中白色部分代表背景，灰色代表目标X

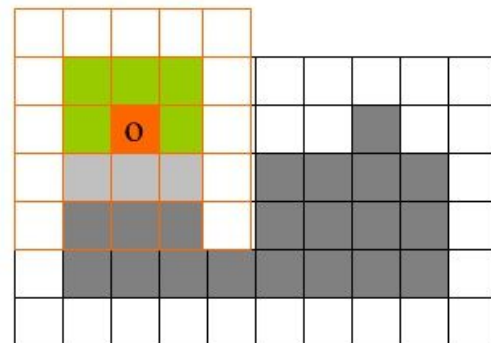


为结构元素B，其中原点位置为橘黄色标注

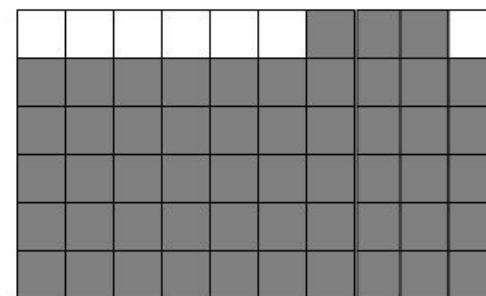


此过程为闭运算

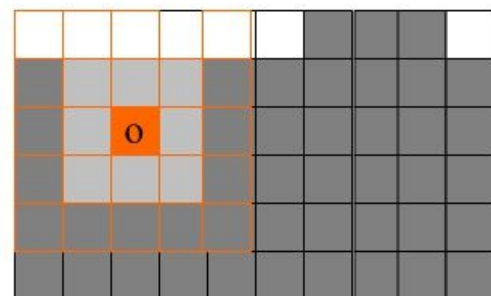
(1) 进行膨胀操作：
进行遍历处理



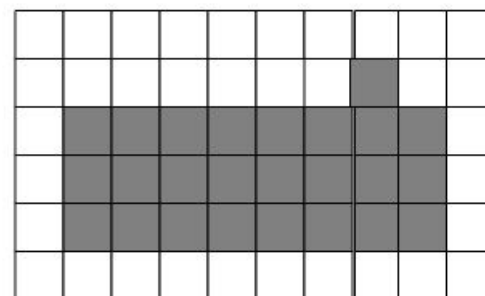
经过膨胀后的结果



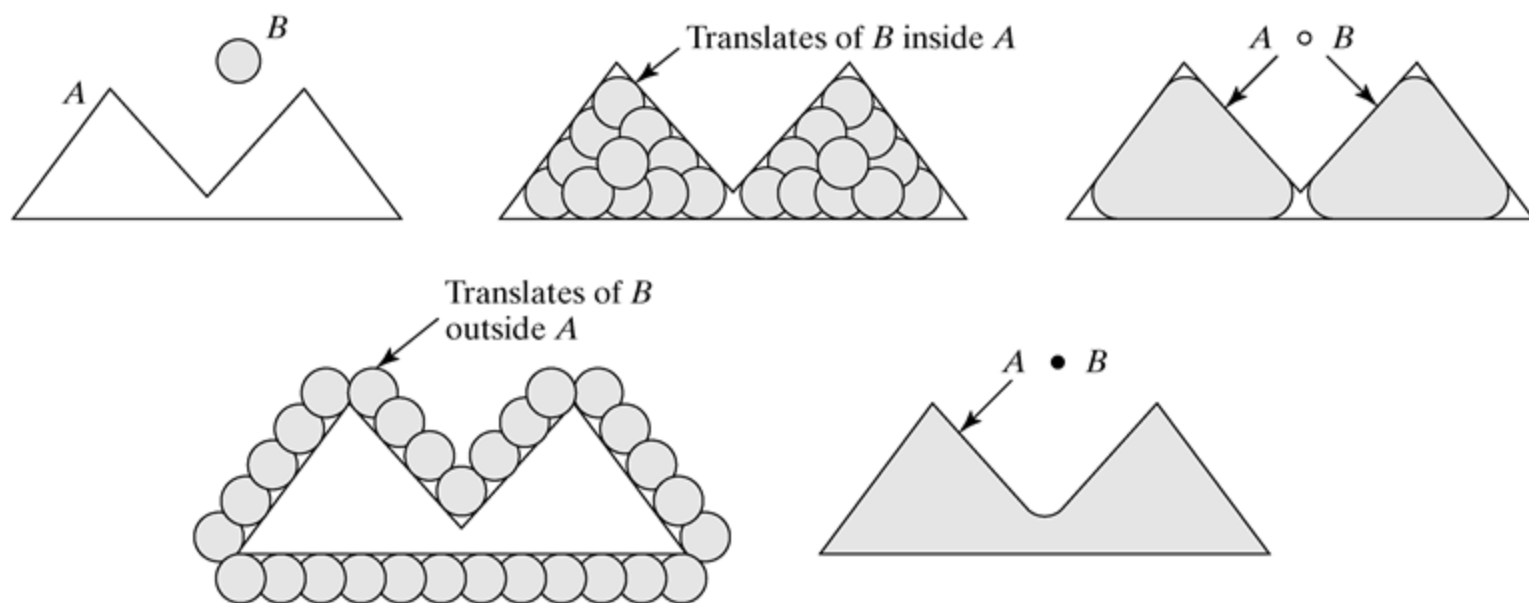
(2) 进行腐蚀操作：
进行遍历处理



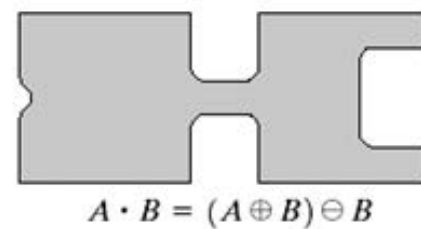
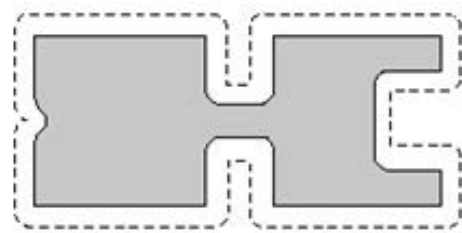
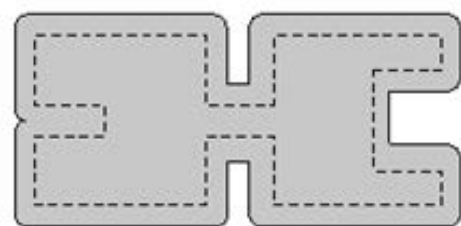
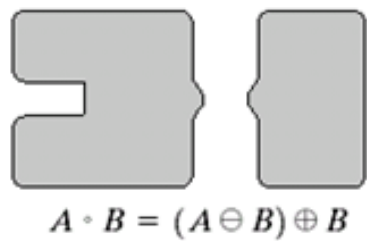
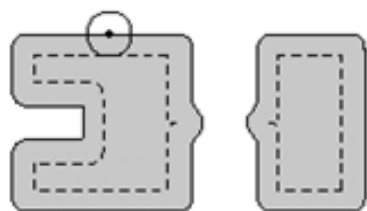
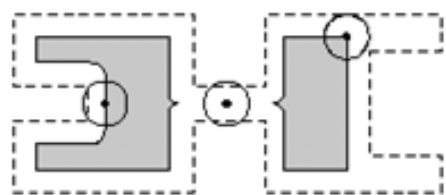
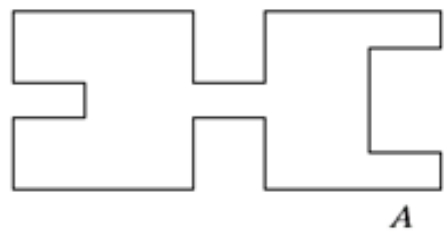
经过腐蚀后的结果



举例



举例

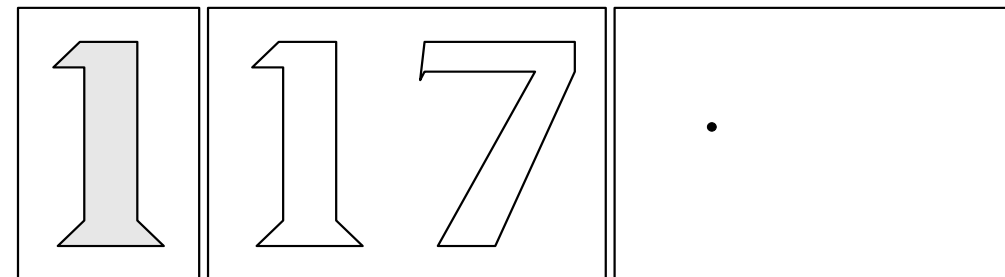


小结

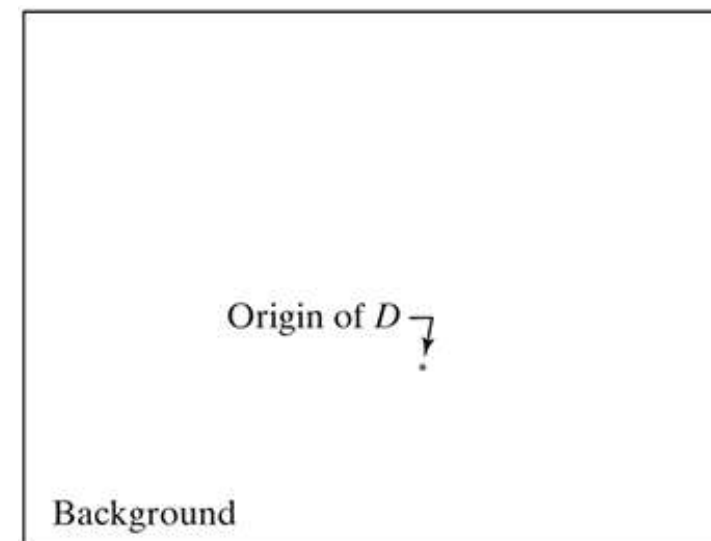
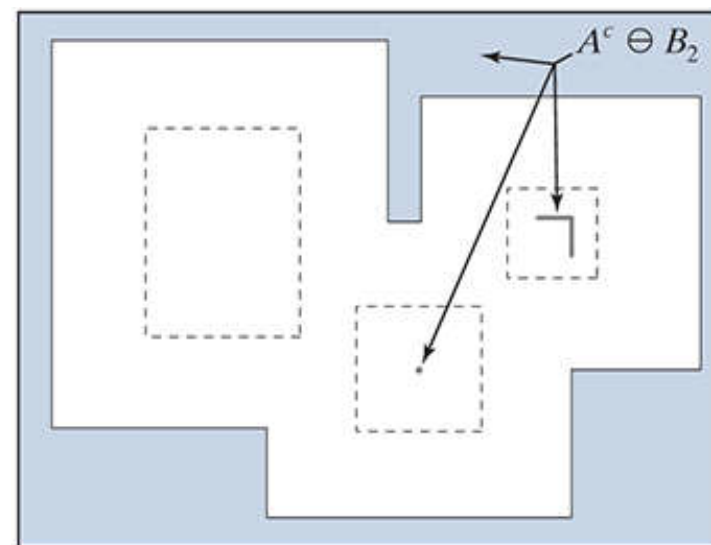
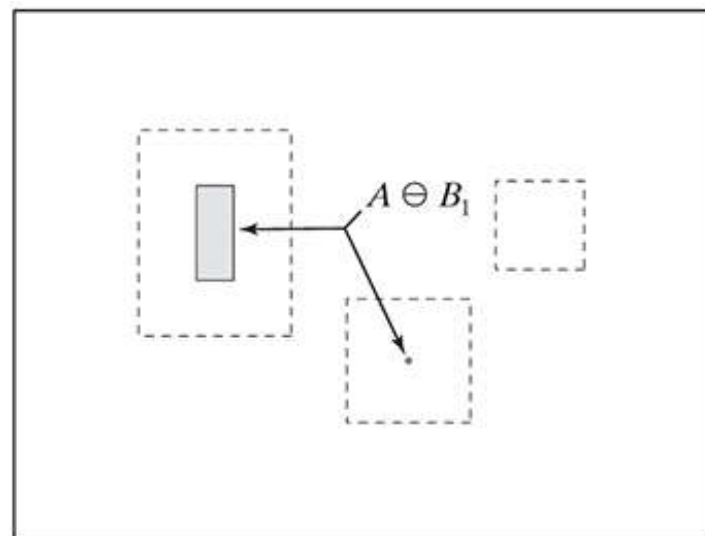
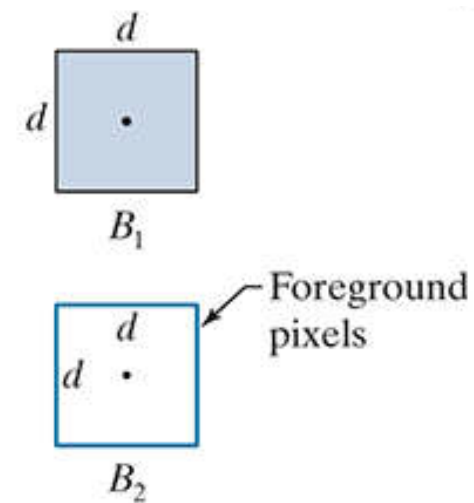
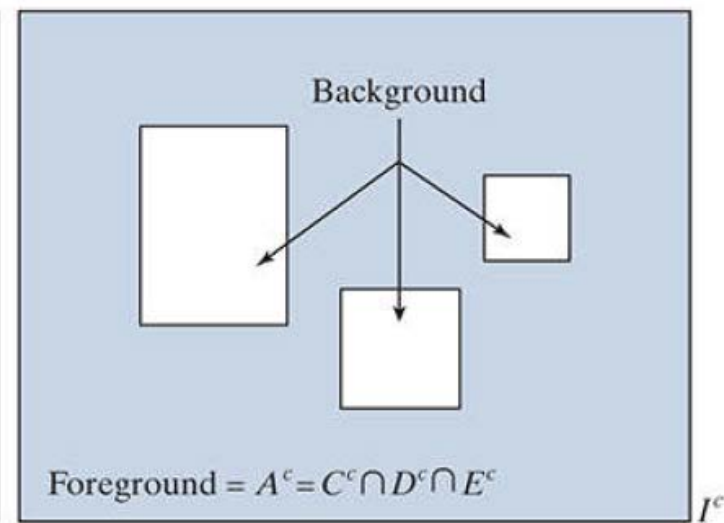
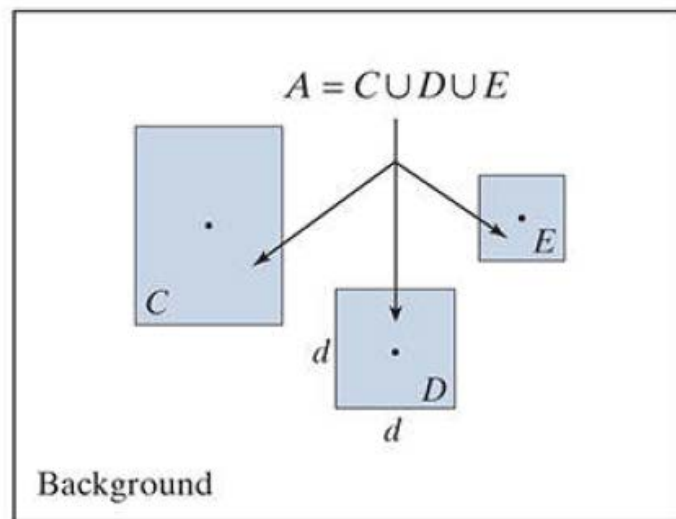
- 开运算一般能
 - 断开狭窄的间断
 - 消除细的突出物，使物体的轮廓变得光滑
- 闭运算同样使轮廓线更为光滑，但与开操作相反的是，它通常
 - 弥合狭窄的间断和长细的鸿沟
 - 消除小的孔洞，填补轮廓线中的断裂
- 和膨胀、腐蚀效果相似，但能保持总的形状不变

击中-击不中变换 Hit-or-Miss

- 形状检测的方法，可以用来做模板匹配
- 设 A 是被研究的图像， B 是结构元素
- B 由不相交的部分 B_1 和 B_2 组成
 - $B = B_1 \cup B_2$, 且 $B_1 \cap B_2 = \emptyset$
- 在 A 中对 B 进行的击中-击不中变换
 - 在 A 中检测 B_1
 - 在 A^c 中检测 B_2 (在 A 中击不中)
 - 相当于两个腐蚀结果求交集



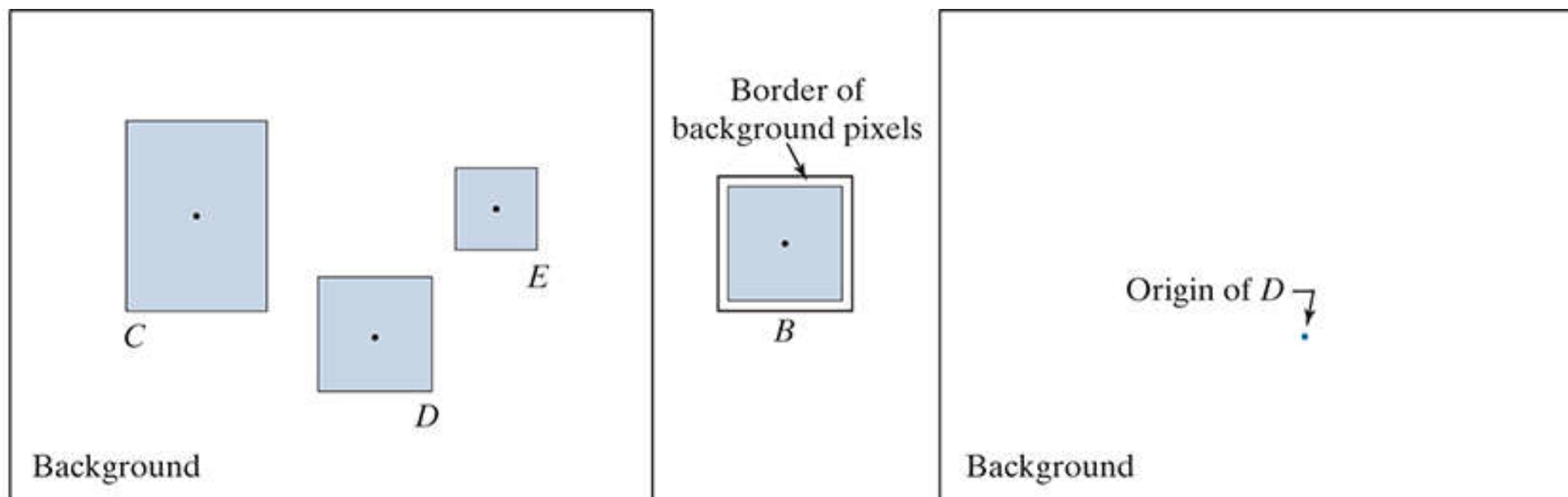
$$\begin{aligned} I \star B_{1,2} &= \{z \mid (B_1)_z \subseteq A \text{ and } (B_2)_z \subseteq A^c\} \\ &= (A \ominus B_1) \cap (A^c \ominus B_2) \end{aligned}$$



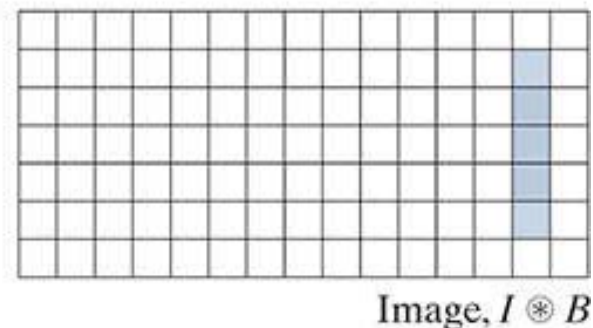
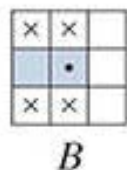
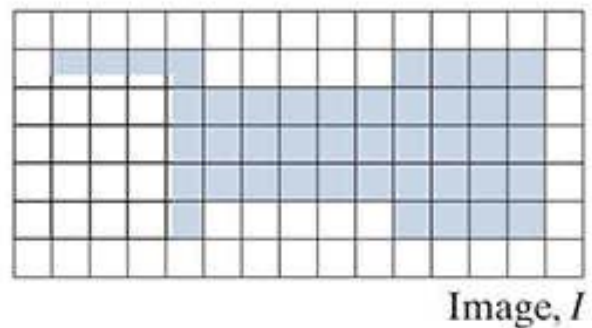
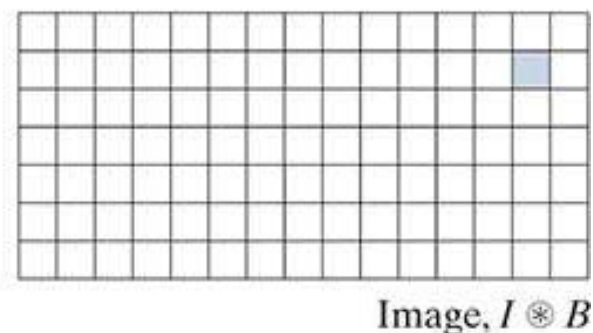
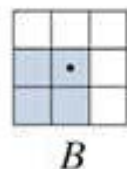
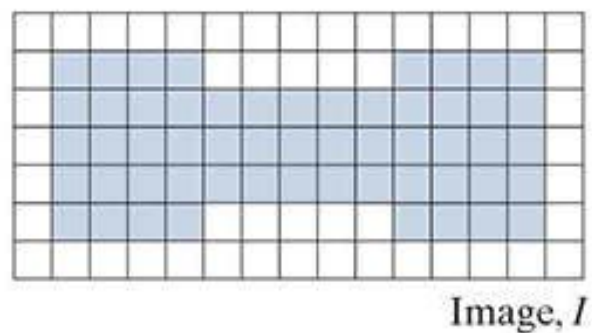
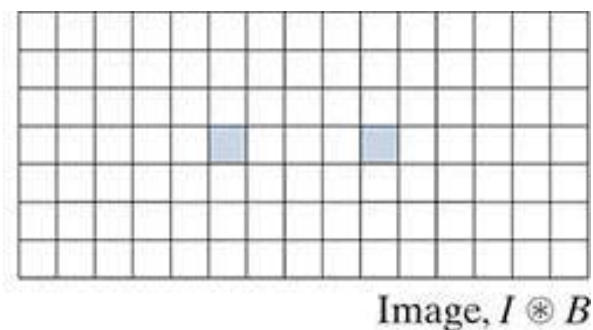
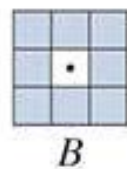
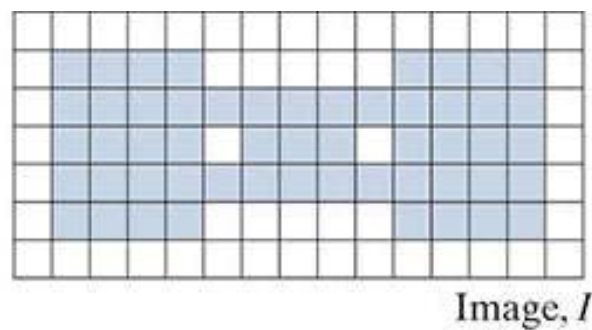
只用一个结构元实现击中-击不中变换

- 同时处理结构元中的前景和背景

$$I \circledast B = \{z \mid (B)_z \subseteq I\}$$



考虑背景的击中-击不中变换

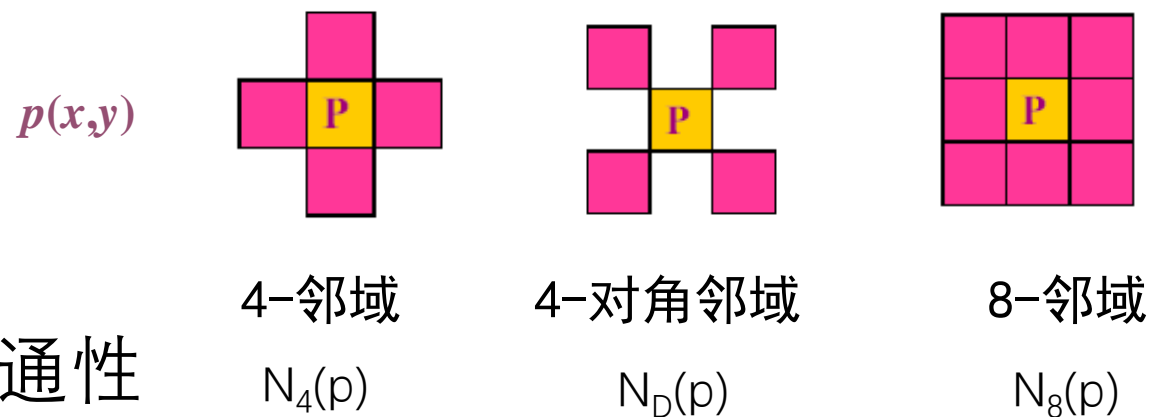


应用

- 边界提取(Boundary Extraction)
- 空洞填充(Region Filling)
- 连通分量提取(Extraction of Connected Components)

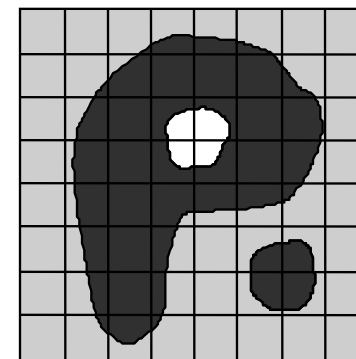
像素的邻域

- 像素 p 与其周围像素的空间位置关系



- 像素之间的连通性

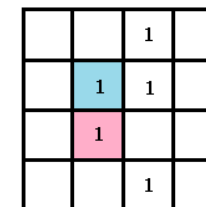
- 空间位置是否相邻
- 像素值是否相似
 - 定义一个像素值相似集合 V
 - 比如, 二值图 $V=\{1\}$, 灰度图 $V=\{8, 9, \dots, 15, 16\}$



3种类型的邻接性

- 4-邻接

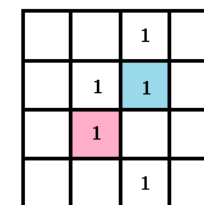
- 2个像素p和q在V中取值, 且q在 $N_4(p)$ 中, 则它们为4-邻接



4邻接

- 8-邻接

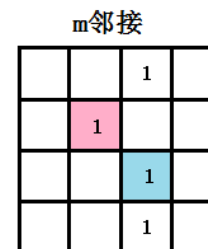
- 2个像素p和q在V中取值且q在 $N_8(p)$ 中, 则它们为8-邻接



8邻接

- m-连接 (混合连接)

- 2个像素p和q在V中取值, 且满足下列条件之一, 则它们为m-邻接
- q在 $N_4(p)$ 中
- q在 $N_D(p)$ 中, 且 $N_4(p) \cap N_4(q)$ 是空集

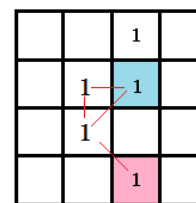


通路

- 一条从坐标 (x_0, y_0) 的像素p, 到具有坐标 (x_n, y_n) 的像素q的通路(路径), 是指具有坐标

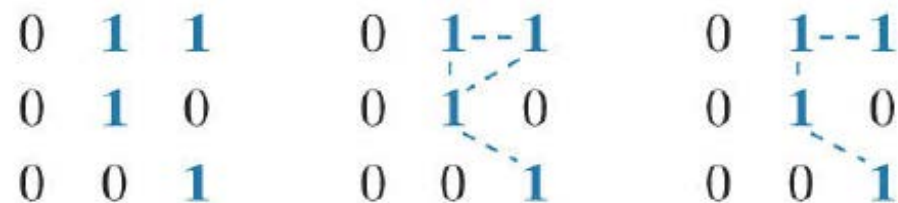
$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$$

- (x_{i-1}, y_{i-1}) 和 (x_i, y_i) 相邻接



- 闭合通路

- 如果 $(x_0, y_0) = (x_n, y_n)$



- 通路类型

- 视 (x_{i-1}, y_{i-1}) 和 (x_i, y_i) 的邻接类型对应为4、8或m通路

一些概念

- 令S代表一个像素子集
- p和q在S中是连通的
 - 如果p和q之间存在一个通路
 - 通路像素都在S中

1	0	0	0	1
1	1	1	0	0
1	0	0	0	0
0	1	0	0	1
0	0	1	1	0

(a)

1	0	0	0	2
1	1	1	0	0
1	0	0	0	0
0	3	0	0	5
0	0	4	4	0

(b)

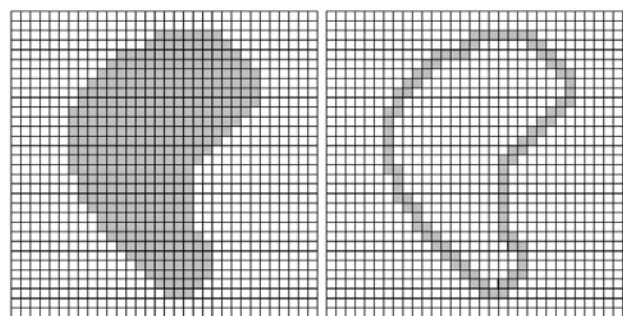
1	0	0	0	2
1	1	1	0	0
1	0	0	0	0
0	1	0	0	1
0	0	1	1	0

(c)

- 连通分量 connected component
 - S中连通到像素p的所有像素集合
- 连通集
 - 如果S仅有一个连通分量，则集合S叫做一个连通集 (connected set)
 - 在图像中，如果R是连通集，则称R为一个区域

一些概念

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	1	0
0	1	1	0	0	0	0	1	0
0	1	1	1	0	0	0	1	0
0	1	1	1	0	0	0	1	0
0	0	0	0	0	0	0	0	0



- 边界
 - 区域R的边界(边框、轮廓)是像素的集合
 - 构成边界的像素至少有一个邻点不在区域R中
 - 内边界 (区域内, 有可能不闭合)
 - 外边界 (背景中, 闭合)
- 和边缘的关系
 - 边界通常是一条闭合通路, 是“整体”概念
 - 边缘常表示由灰度值剧烈变化处的像素构成的像素子集, 是“局部”概念

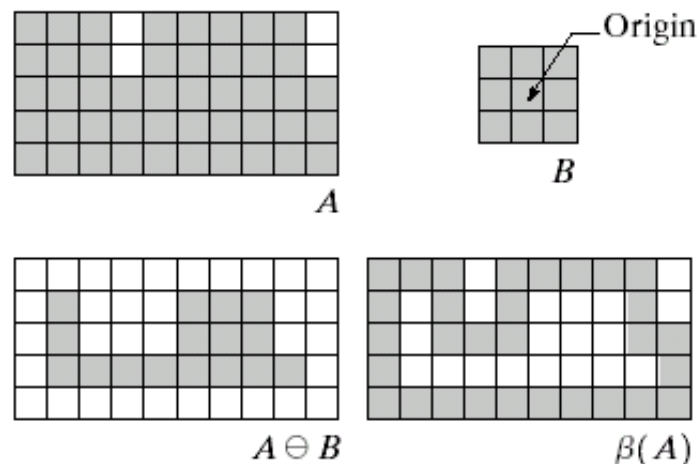
边界提取

- Boundary Extraction
- 边界
 - 一个区域 R 的边界(也称为边缘或轮廓)是区域中像素的集合, 构成边界的像素至少有一个邻点不在区域 R 中
- 先用 B 对 A 腐蚀, 而后再用 A 减去腐蚀结果

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

a b
c d

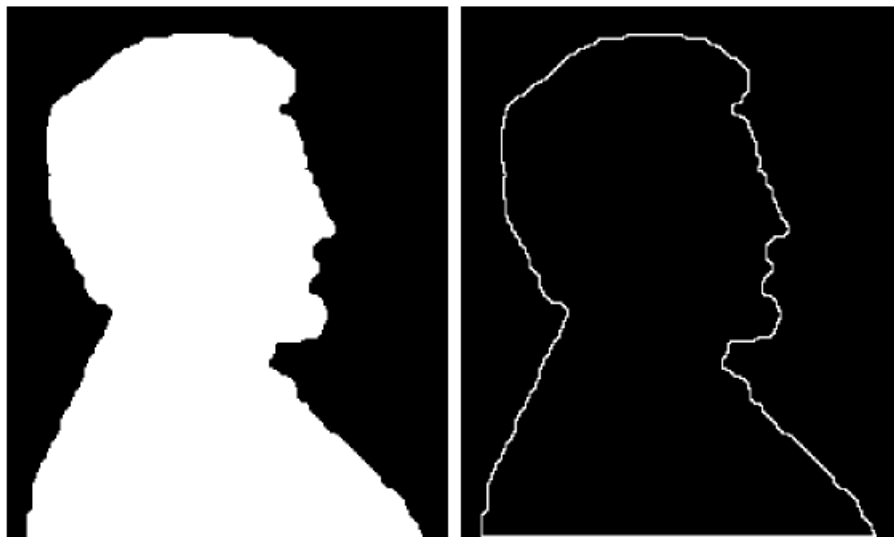
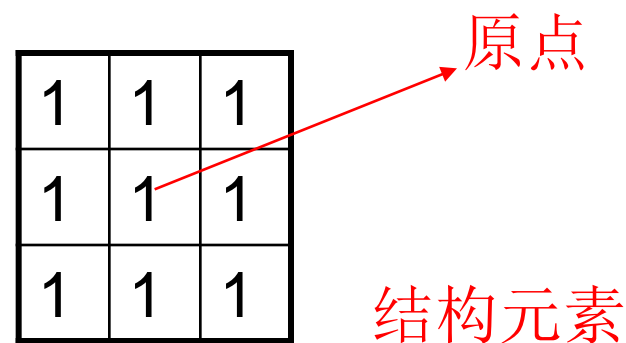
FIGURE 9.13 (a) Set A . (b) Structuring element B . (c) A eroded by B . (d) Boundary, given by the set difference between A and its erosion.



边界提取

- 1表示为白色； 0表示为黑色

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



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).

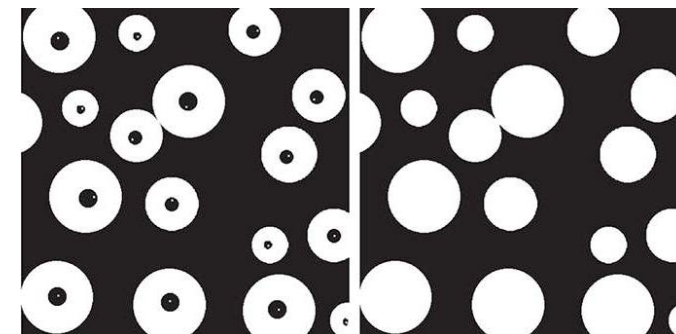
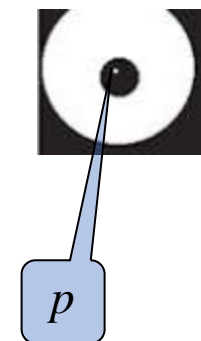
孔洞填充

- I 表示一个集合，元素均是8 连通边界点
- 边界内部包含一个背景区域（孔洞，黑色值）
- 从孔洞内的一个点 p 开始，用“1”填充整个区域

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

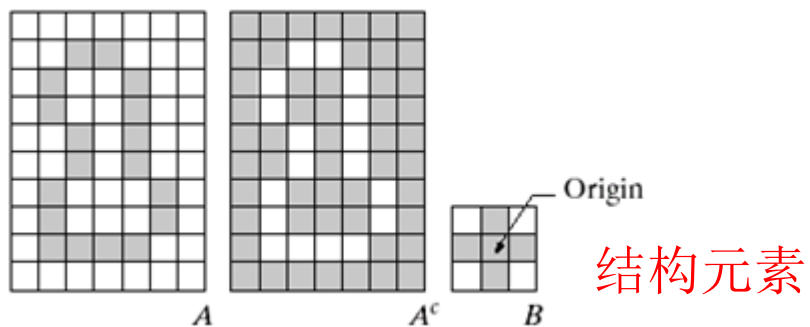
- 以边界为限制，序列式条件膨胀

- 1) $X_0 = p$ ，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
- 2) 如果对上述公式的膨胀不加限制，结果将填充整个区域。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



孔洞填充

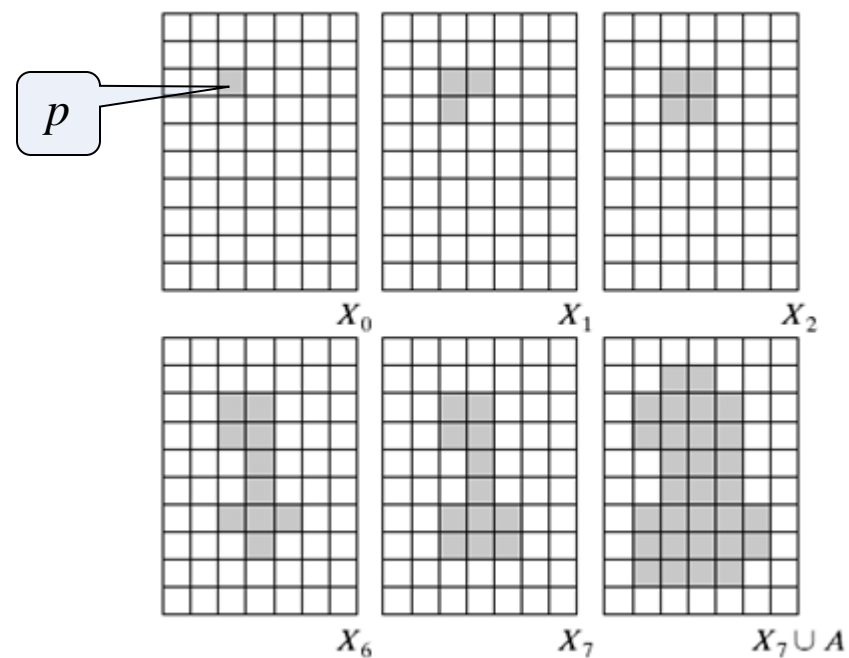
A 表示一个包含子集的集合，其子集的元素均是区域的8连通边界点



结构元素

非边界(背景)点标记为0，边界点1

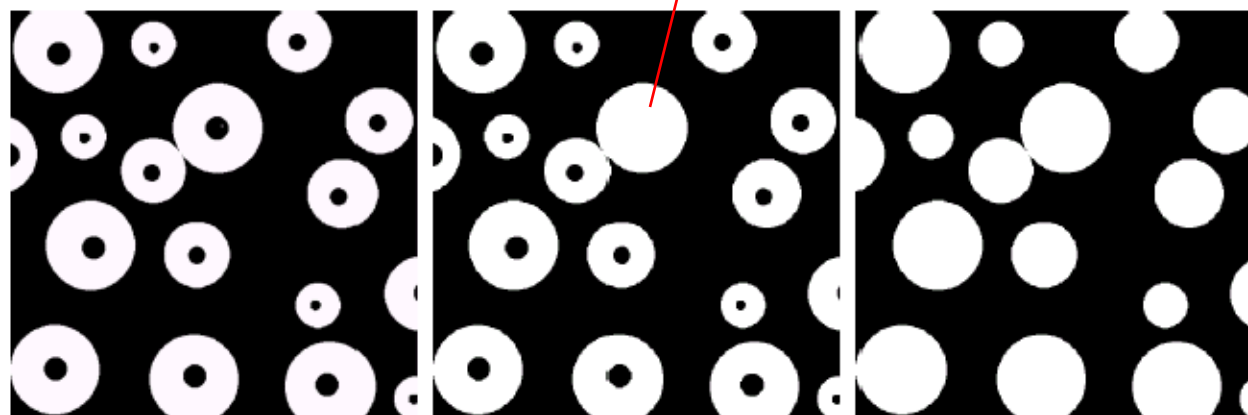
目标：从边界内的一点开始，用1填充整个区域



孔洞填充

- 通过区域填充消除白色圆圈内的黑点

第一个区域填充的结果



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.

黑色点是背景点还是区域内部的点是已知的

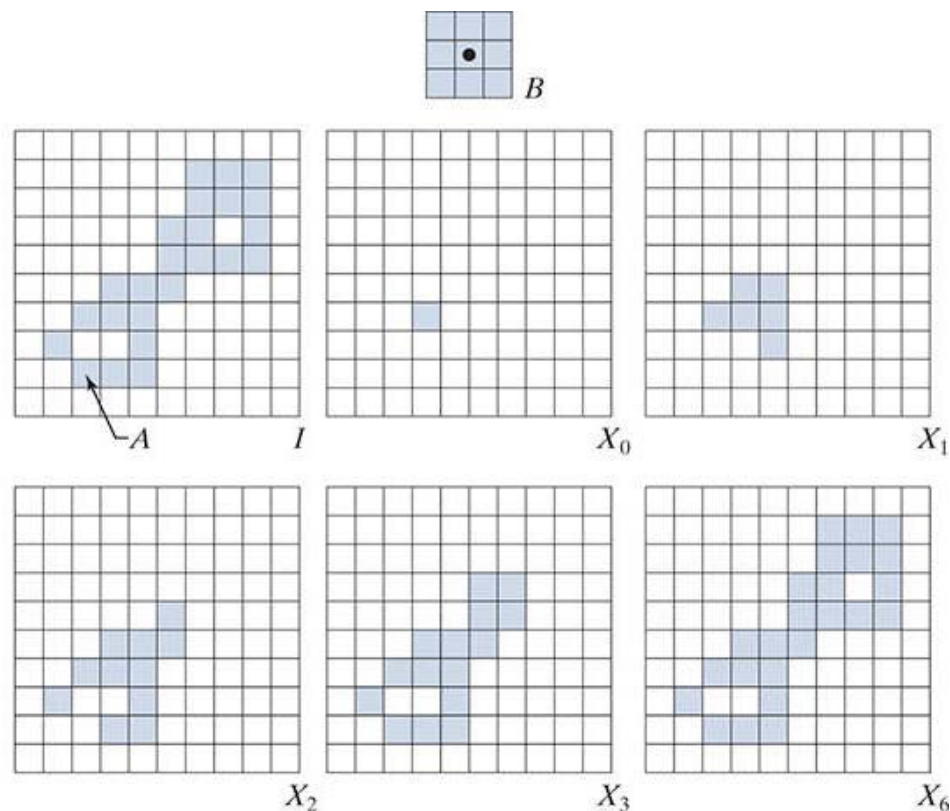
连通分量提取

- 令 X 表示集合 I 中的一个连通分量
- 并假设 X 中的一个点 p 是已知的

- 连通分量 X 的所有元素用下式得到

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

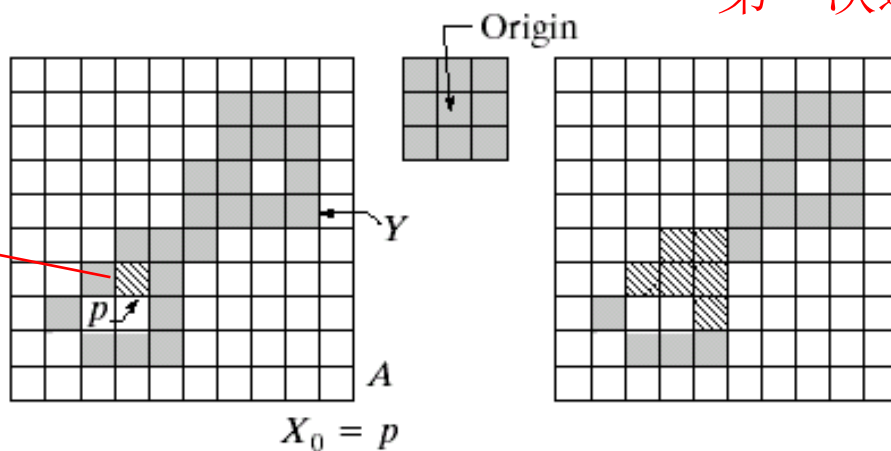
- 序列式条件膨胀
- $X_0 = p$, 如果 $X_k = X_{k-1}$, 算法收敛。令 $Y = X_k$, 即得到连通分量



8连通的结构元素

第一次迭代的结果

起始点



第二次迭代的结果

最终结果

a b c
d e

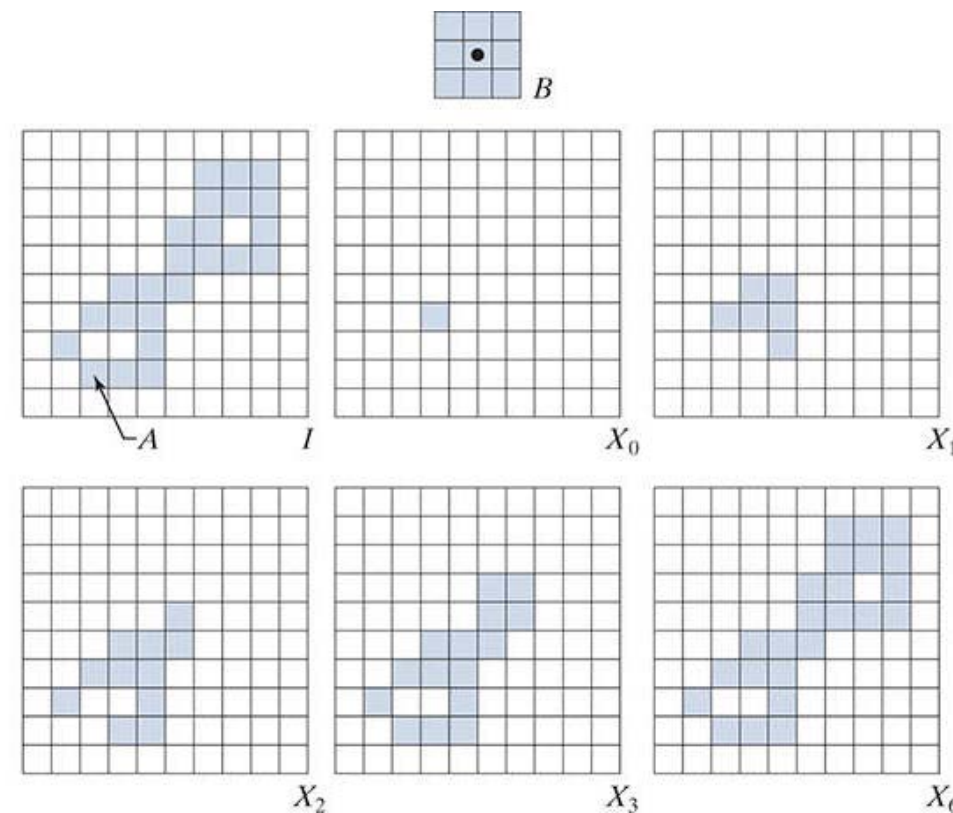
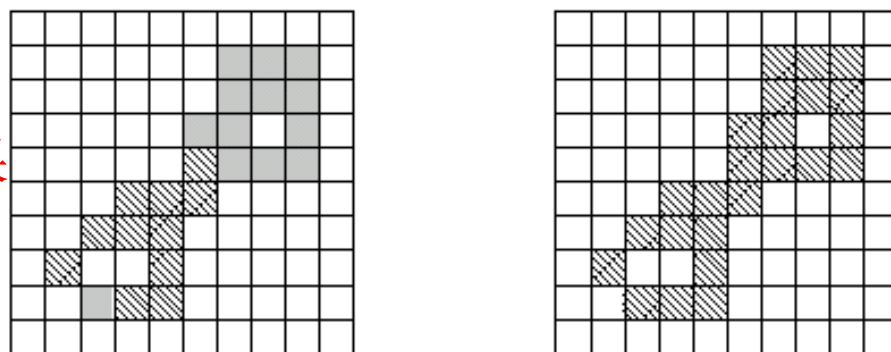
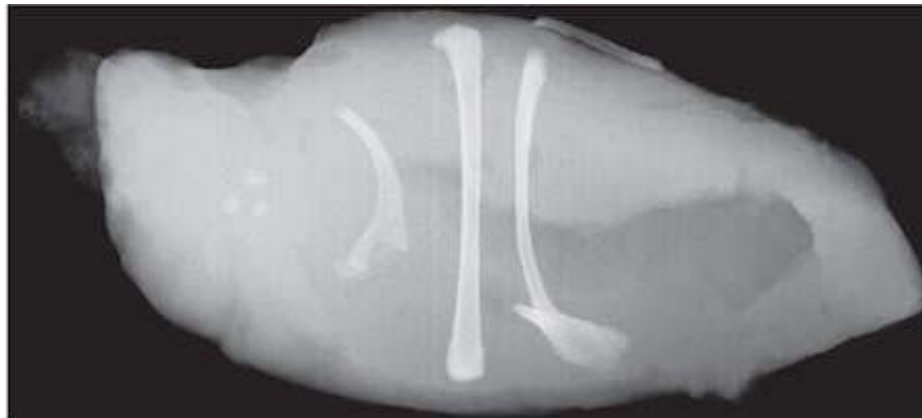


FIGURE 9.17 (a) Set A showing initial point p (all shaded points are valued 1, but are shown different from p to indicate that they have not yet been found by the algorithm). (b) Structuring element. (c) Result of first iterative step. (d) Result of second step. (e) Final result.

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

连通分量提取

包含碎骨的鸡
胸X光图像



使用阈值将骨头从背
景中提取出来



用5*5的结构元素对阈
值处理后的图像进行
腐蚀，消除细节，保
留大尺寸物体



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

提取连通分量，识别大尺寸分量

作业

- 实验7：形态学实验
实现膨胀、腐蚀、边界提取操作；