第四章：形态学

中国科学技术大学
电子工程与信息科学系

主讲教师：李厚强 (lihq@ustc.edu.cn)
周文罡 (zhwg@ustc.edu.cn)
李　礼 (lil1@ustc.edu.cn)
胡　洋 (eeyhu@ustc.edu.cn)

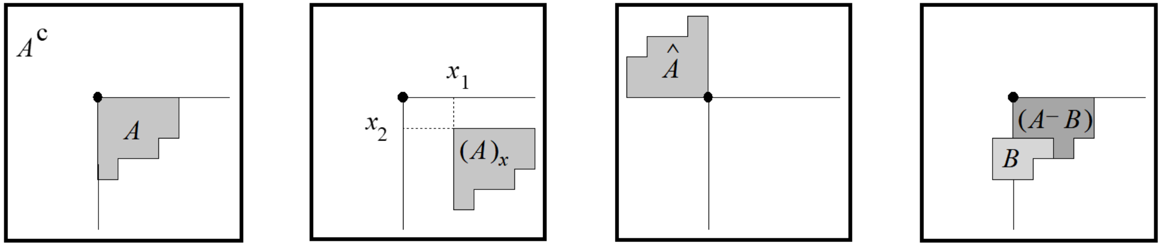
形态学

- 形态学
  - 二值形态学
    - ✓ 基本定义
    - ✓ 基本运算
    - ✓ 实用算法

二值形态学

□ 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：
- 并集：
- 交集：



(a) (b) (c) (d)

- 补集： $A^c=\{x|x\notin A\}$
- 位移： $(A)_x=\{y|y=a+x,a\in \mathrm{A}\}$
- 映像： $\hat{A}=\{x|x=-a,a\in \mathrm{A}\}$
- 差集： $A-B=\{x|x\in A,x\notin B\}=A\cap B^c$

二值形态学基本运算

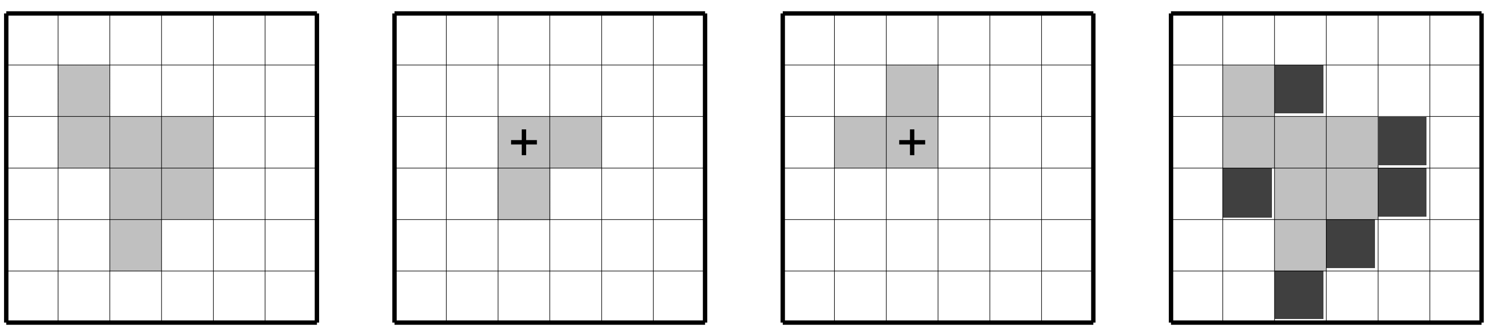
- 集合运算
  - A为图象集合，B 为结构元素（集合）
  - 数学形态学运算是用 B 对 A 进行操作
  - **结构元素要指定1个原点（参考点）**

膨胀和腐蚀

- 膨胀
  - 膨胀的算符为⊕

$$A \oplus B = \{x | [(\hat{B})_x \cap A] \neq \emptyset\}$$

集合A　　结构元素B　　B的映象　　集合$A \oplus B$



(a)　　(b)　　(c)　　(d)

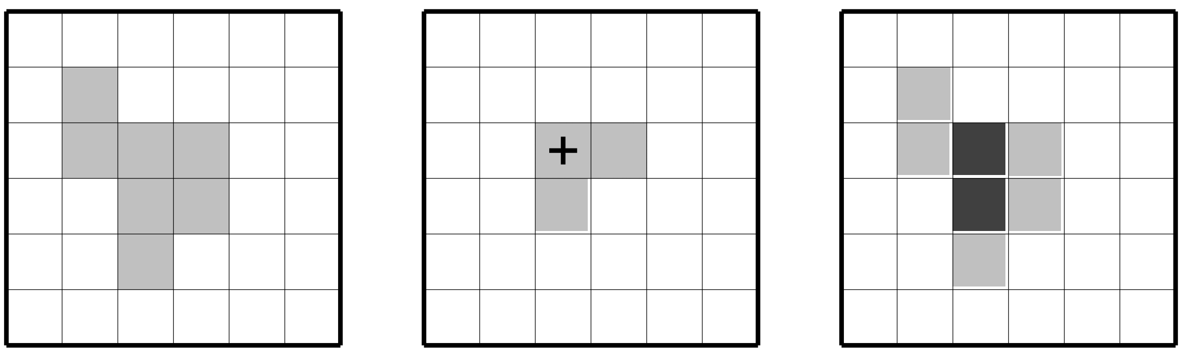
膨胀和腐蚀

□ 腐蚀

- 腐蚀的算符为⊖

$$A \ominus B = \{x | (B)_x \subseteq A\}$$

集合A　　结构元素B　　集合$A \ominus B$



(a)　　(b)　　(c)

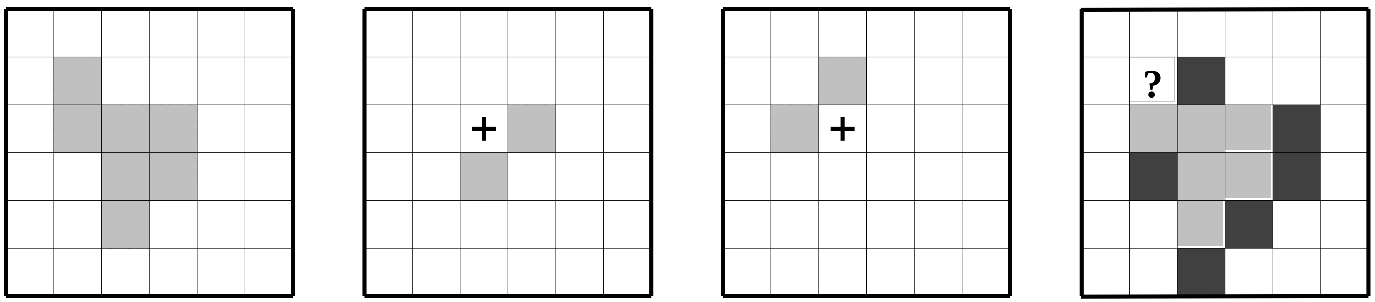
膨胀和腐蚀

- 原点不包含在结构元素中时的膨胀和腐蚀
  - 原点包含在结构元素中
    - 膨胀运算：$A \subseteq A \oplus B$
    - 腐蚀运算：$A \ominus B \subseteq A$
  - 原点不包含在结构元素中
    - 膨胀运算：$A \not\subset A \oplus B$
    - 腐蚀运算：$A \ominus B \subseteq A$，或 $A \ominus B \not\subset A$

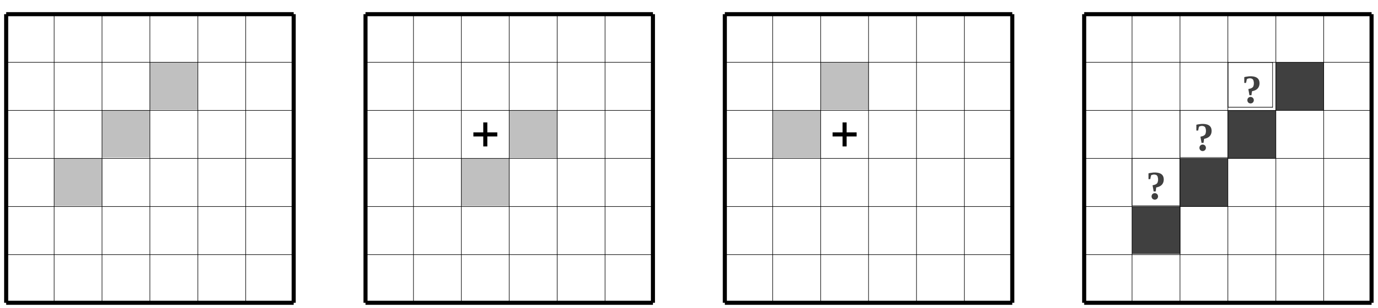
膨胀和腐蚀

- 原点不包含在结构元素中时的膨胀运算

$$A \not\subset A \oplus B \qquad A \oplus B = \{x | [(\hat{B})_x \cap A] \neq \emptyset\}$$



A在膨胀中自身完全消失了

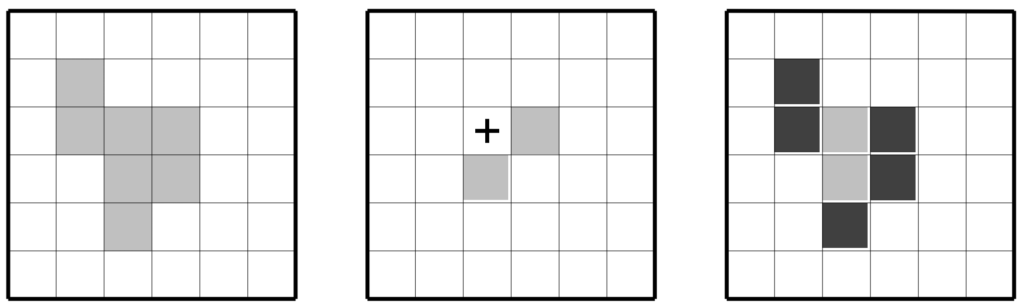


膨胀和腐蚀

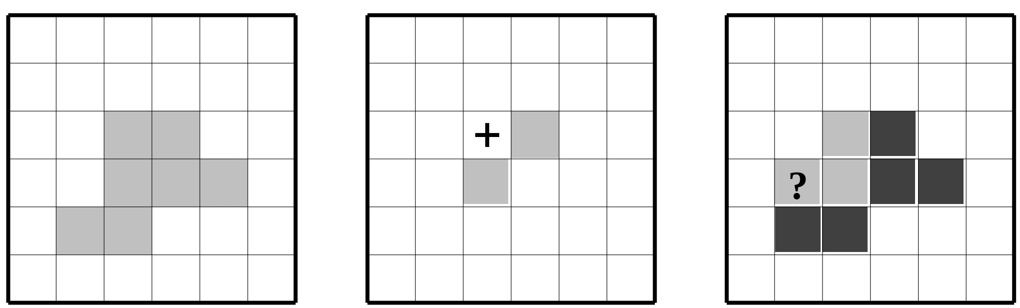
□ 原点不包含在结构元素中时的腐蚀运算

$A \ominus B \subseteq A$

$$A \ominus B = \{x | (B)_x \subseteq A\}$$



$A \ominus B \not\subset A$



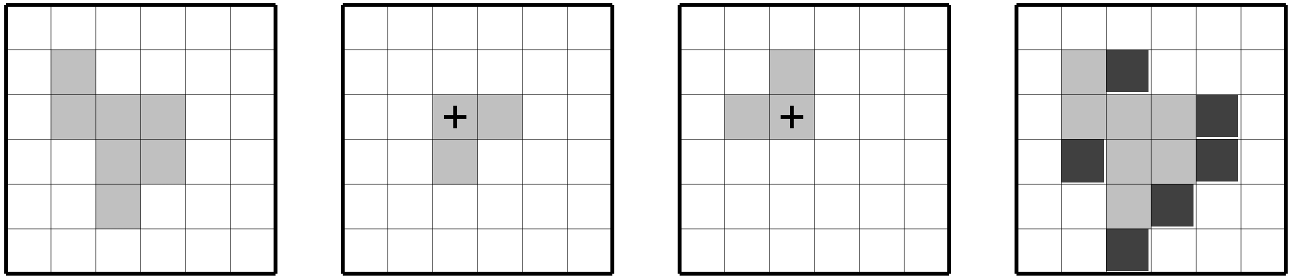
膨胀和腐蚀

- 用向量运算实现膨胀和腐蚀

$$A \oplus B = \{x | x = a + b, 对于任意 a \in A 和 b \in B\}$$

$$A = \{(1, 1), (1, 2), (2, 2), (3, 2), (2, 3), (3, 3), (2, 4)\}$$

$$B = \{(0, 0), (1, 0), (0, 1)\}$$



$$A \oplus B = \{(1, 1), (2, 1), (1, 2), (2, 2), (3, 2), (4, 2), (1, 3), (2, 3), (3, 3), (4, 3), (2, 4), (3, 4), (2, 5)\}$$

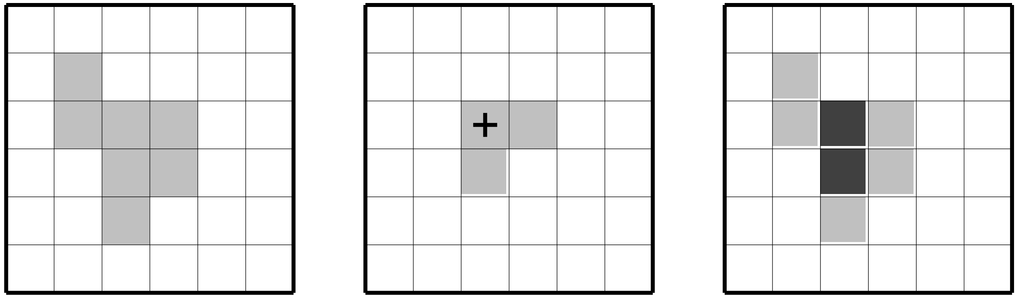
膨胀和腐蚀

- 用向量运算实现膨胀和腐蚀

$$A \ominus B = \{x|(x+b) \in A\text{对每一个}b \in B\}$$

$$A = \{(1,1),(1,2),(2,2),(3,2),(2,3),(3,3),(2,4)\}$$

$$B = \{(0,0),(1,0),(0,1)\}$$



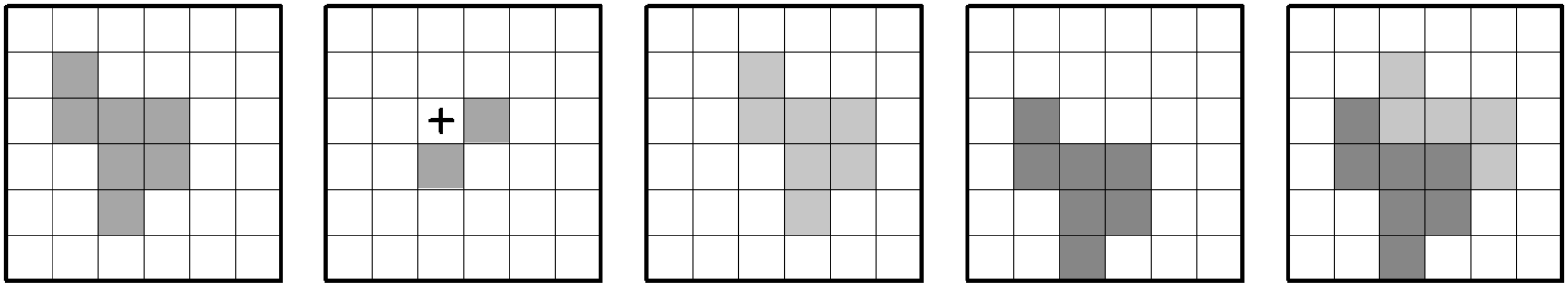
$$A \ominus B = \{(2,2),(2,3)\}$$

膨胀和腐蚀

- 用位移运算实现膨胀和腐蚀

按每个b来位移A并把结果或（OR）起来

$$A \oplus B = \bigcup_{b \in B} (A)_b$$



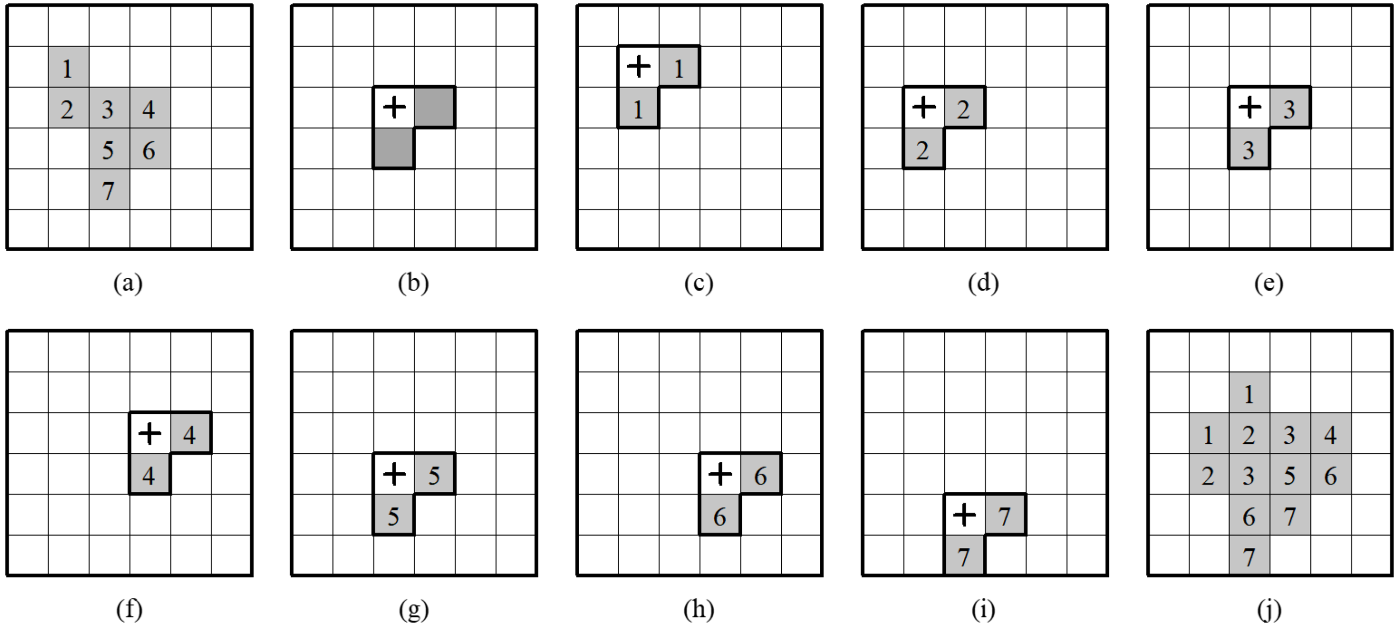
(a) (b) (c) (d) (e)

膨胀和腐蚀

- 用位移运算实现膨胀和腐蚀

按每个a来位移B并把结果或（OR）起来

$$A \oplus B = \bigcup_{a \in A} (B)_a$$

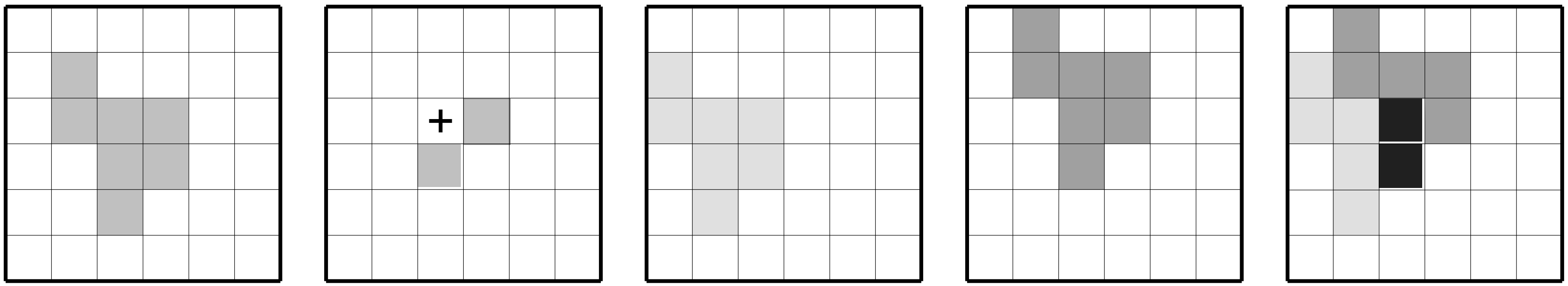


膨胀和腐蚀

- 用位移运算实现膨胀和腐蚀

按每个b来负位移A并把结果交（AND）起来

$$A \ominus B = \bigcap_{b \in B} (A)_{-b}$$

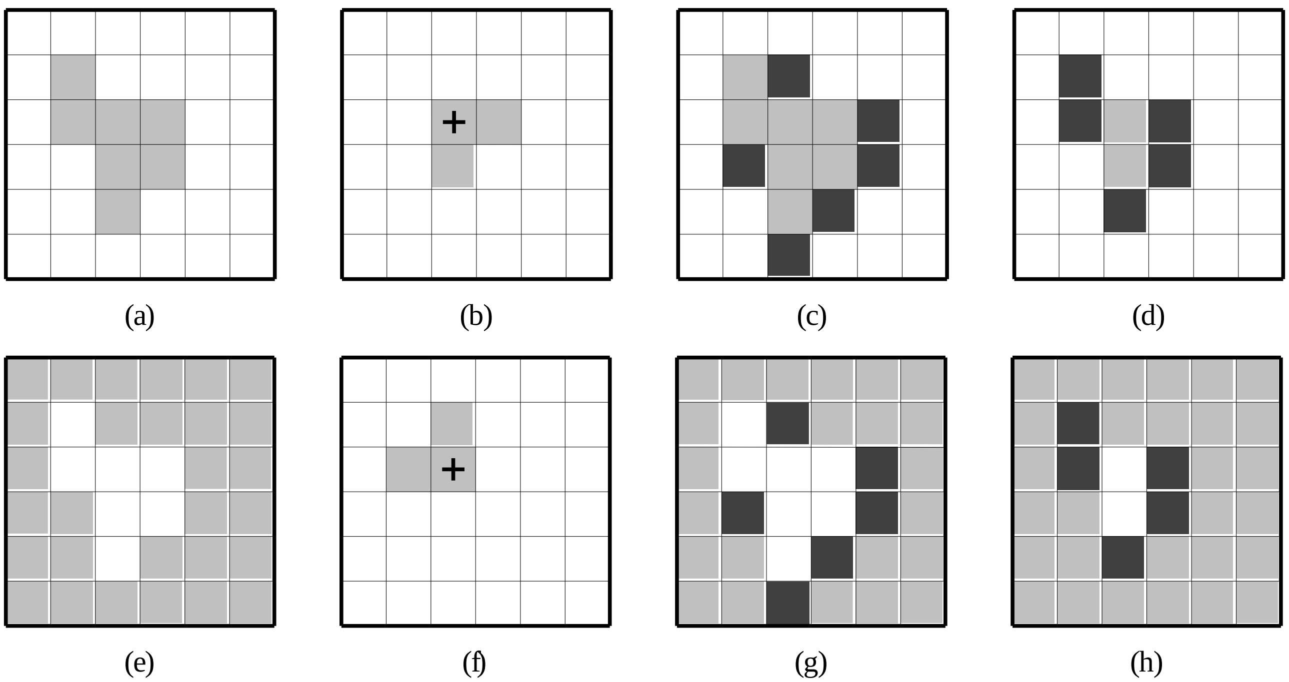


(a) (b) (c) (d) (e)

膨胀和腐蚀

□ 膨胀和腐蚀的对偶性

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



膨胀和腐蚀

- 膨胀和腐蚀的对偶性

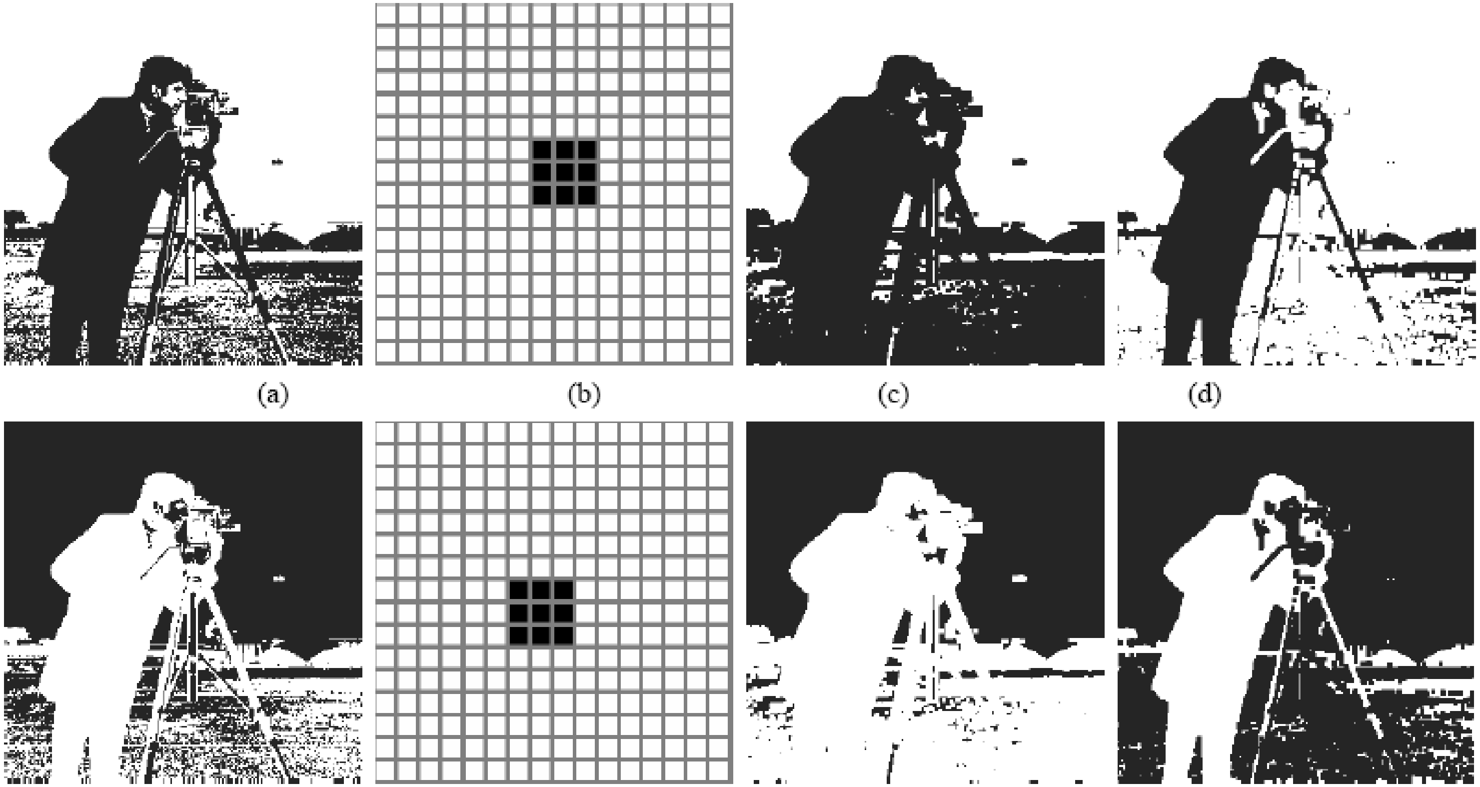


图 14.2.12 膨胀和腐蚀的对偶性验证实例

开启和闭合

开启和闭合定义

- 膨胀和腐蚀并不互为逆运算
- 它们可以级连结合使用
- 开启：先对图象进行腐蚀然后膨胀其结果

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

- 闭合：先对图象进行膨胀然后腐蚀其结果

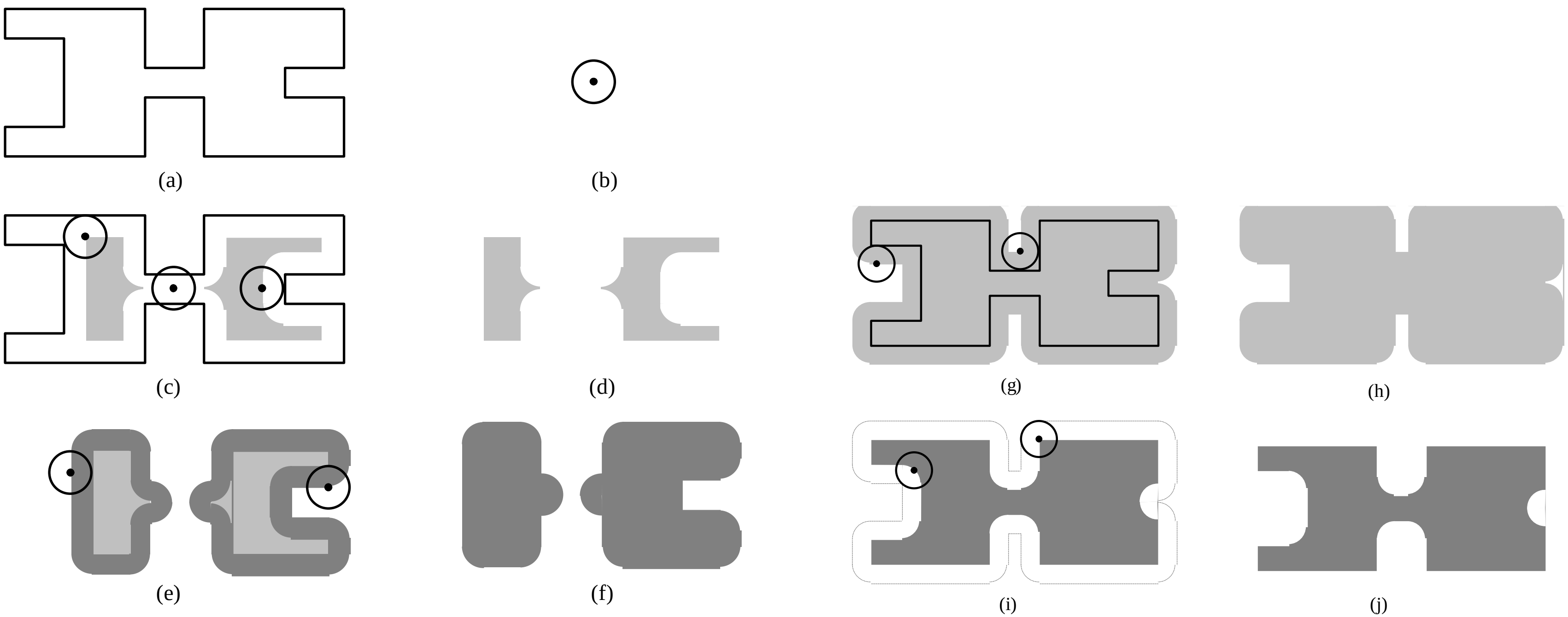
$$A \cdot B = (A \oplus B) \ominus B$$

- 开启和闭合不受原点是否在结构元素之中的影响

开启和闭合

开启和闭合定义

- 开启运算可以把比结构元素小的突刺滤掉
- 闭合运算可以把比结构元素小的缺口或孔填充上



开启

闭合

开启和闭合

- 开启和闭合定义

原图

(a) (b)

图 14.2.14 开启和闭合实例

开启和闭合

- 开启和闭合的对偶性
  - 开启和闭合也具有对偶性

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

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

$$(A \circ B)^c = [(A \ominus B) \oplus B]^c = (A \ominus B)^c \ominus \hat{B} = A^c \oplus \hat{B} \ominus \hat{B} = A^c \cdot \hat{B}$$

$$(A \cdot B)^c = [(A \oplus B) \ominus B]^c = (A \oplus B)^c \oplus \hat{B} = A^c \ominus \hat{B} \oplus \hat{B} = A^c \circ \hat{B}$$

开启和闭合

开启和闭合与集合的关系

操 作	并 集	交 集
开 启	$\left(\bigcup_{i=1}^{n} A_i\right) \circ B \supseteq \bigcup_{i=1}^{n} (A_i \circ B)$	$\left(\bigcap_{i=1}^{n} A_i\right) \circ B \subseteq \bigcap_{i=1}^{n} (A_i \circ B)$
闭 合	$\left(\bigcup_{i=1}^{n} A_i\right) \cdot B \supseteq \bigcup_{i=1}^{n} (A_i \cdot B)$	$\left(\bigcap_{i=1}^{n} A_i\right) \cdot B \subseteq \bigcap_{i=1}^{n} (A_i \cdot B)$

开启和闭合

- 开启和闭合的几何解释

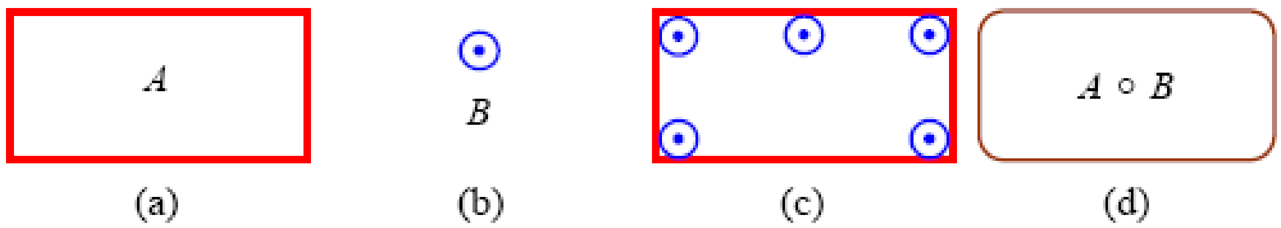


图 14.2.15 开启的填充特性

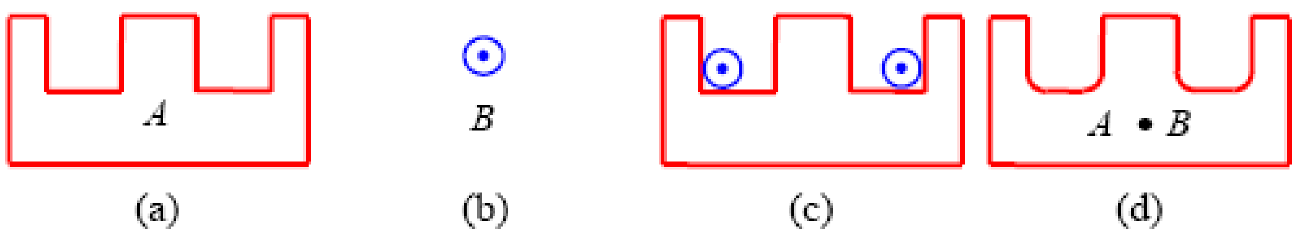


图 14.2.16 闭合的几何解释

击中-击不中变换

- 击中-击不中变换
  - 形状检测的一种基本工具
  - 对应两个操作，所以用到两个结构元素
  - 设A为原始图象，E和F为一对不重合的集合

$$A \Uparrow (E,F) = (A \ominus E) \cap (A^c \ominus F) = (A \ominus E) \cap (A \oplus F)^c$$

E：击中结构元素

F：击不中结构元素

击中-击不中变换

- 击中-击不中变换

$$A \Uparrow (E,F) = (A \ominus E) \cap (A^c \ominus F) = (A \ominus E) \cap (A \oplus F)^c$$

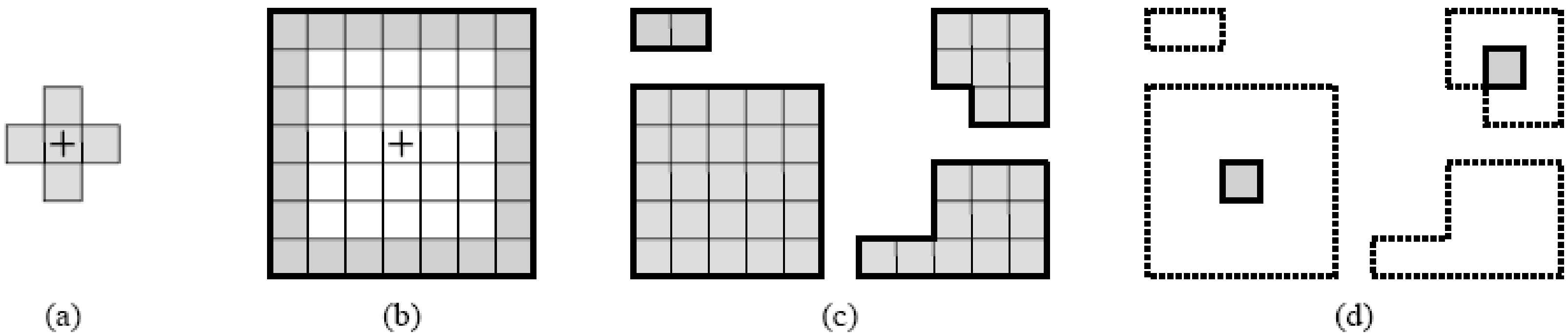


图 14.3.1 击中-击不中变换示例

(a): 击中结构元素　　(b): 击不中结构元素

(c): 原始图像　　(d): 变换结果

击中-击不中变换

- 击中-击不中变换 ((e)和(f)来自于别的变换)

击中变换：[1 1 1]

击不中变换：$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

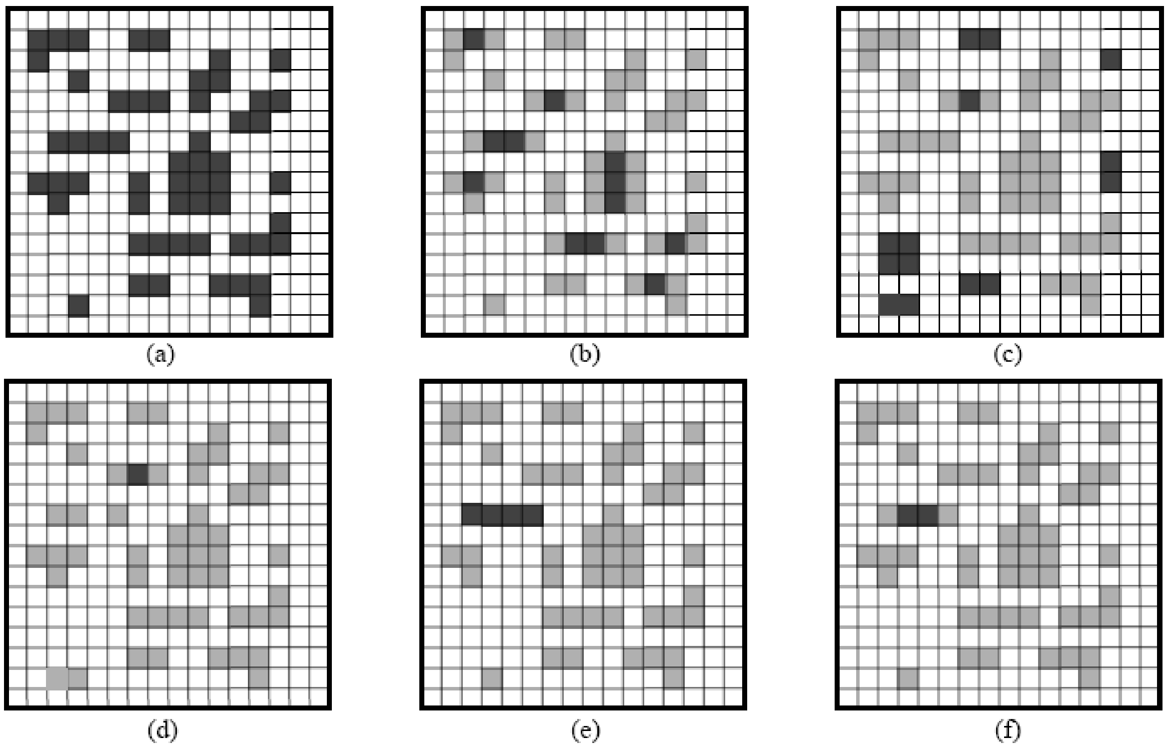
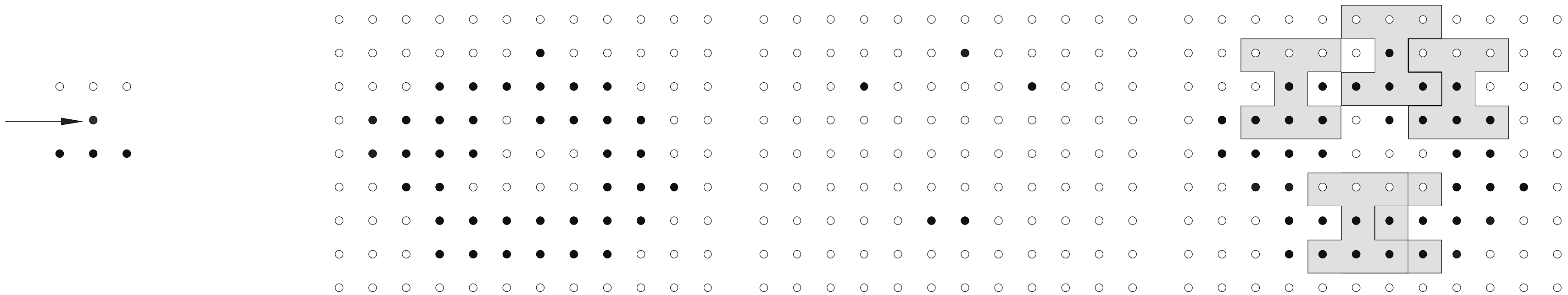


图 14.3.2　利用击中-击不中算子以提取包含水平方向上有连续 3 个像素的线段

击中-击不中变换

- 击中-击不中算子中的击中模板与击不中模板不重合，可以被结合成一个结构元素，1对应击中模板，0对应击不中模板，X表示不关心的像素
- 击中-击不中变换中的结构元素
  - $A \Uparrow B$的结果中仍保留的目标象素对应在A中其邻域与结构元素B对应的象素

$$A \Uparrow B = (A \Theta B_o) \cap (A^c \Theta B_b)$$



(a) (b) (c) (d)

组合运算-I

区域凸包

- 令 $B_i(i=1,2,3,4)$ 代表4个结构元素，$X_i^0=A$构造：

$$X_i^k=\left(X_i^{k-1}\Uparrow B_i\right)\cup A \qquad i=1,2,3,4 \text{ 和 } k=1,2,\ldots$$

- 令 $D_i=X_i^{conv}$，上标“conv”表示在$X_i^k=X_i^{k-1}$意义下收敛，A的凸包可表示为：

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

组合运算-I

- 区域凸包 （X表示其值可为任意）

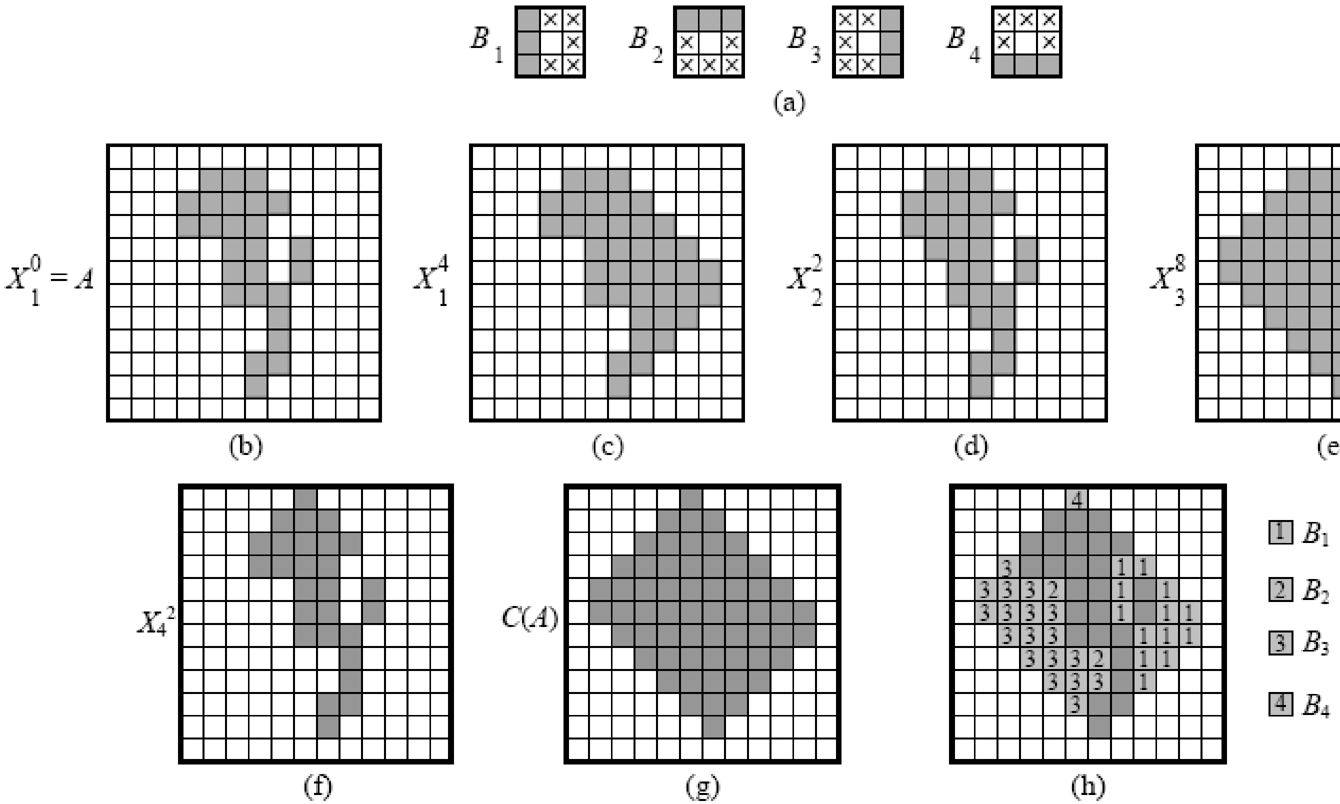


图 14.3.5 构造凸包的示例

组合运算-II

- 细化
  - 用结构元素B细化集合A记作A ⊗ B
  - 借助击中-击不中变换定义

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

  - 定义一个结构元素系列

$$\{B\} = \{B_1, B_2, \dots, B_n\}$$

$$A \otimes \{B\} = A - ((\cdots((A \otimes B_1) \otimes B_2) \cdots) \otimes B_n)$$

组合运算-II

☐ 细化

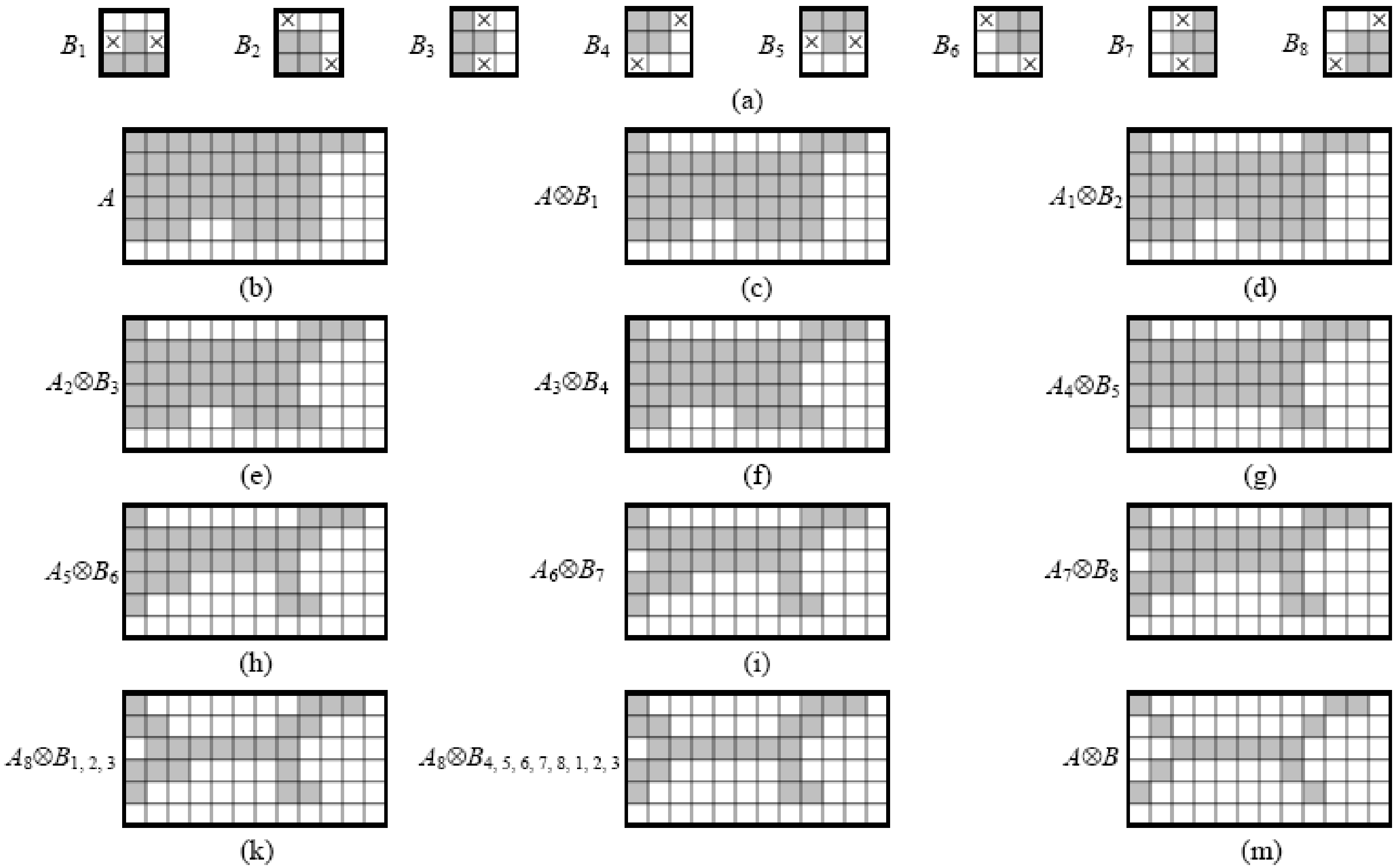


图 14.3.6 细化示例

组合运算-III

- 粗化
  - 用结构元素B粗化集合A记作A $\circledast$ B

$$A \circledast B = A \bigcup (A \Uparrow B)$$

  - 定义为一系列操作

$$A \circledast \{B\} = ((\cdots((A \circledast B_1) \circledast B_2)\cdots) \circledast B_n)$$

粗化从形态学角度来说与细化是对应的，实际中可先细化背景然后求补以得到粗化的结果。换句话说，如果要粗化集合A，可先构造$C = A^c$，然后细化C，最后求C^c。

组合运算-III

□ 粗化

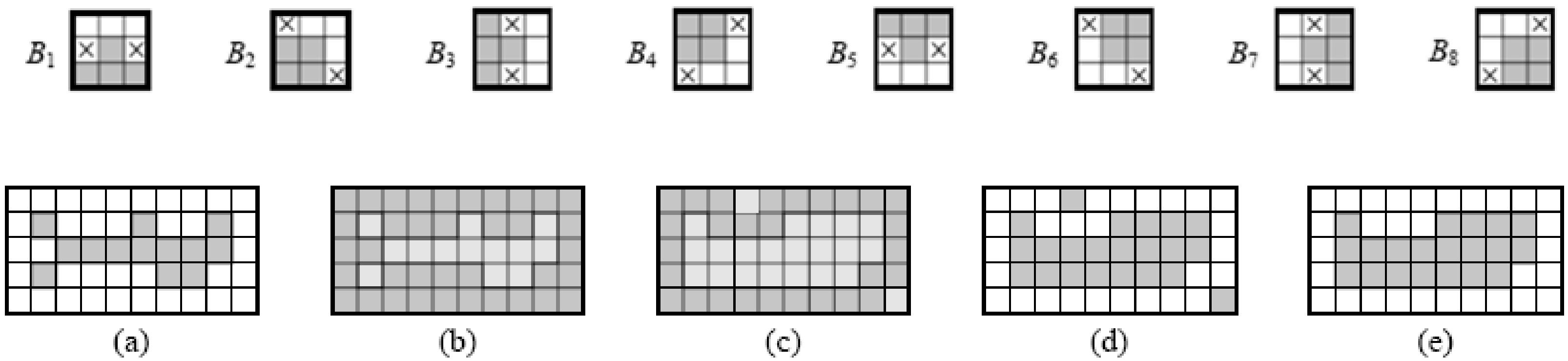


图 14.3.7 利用细化进行粗化

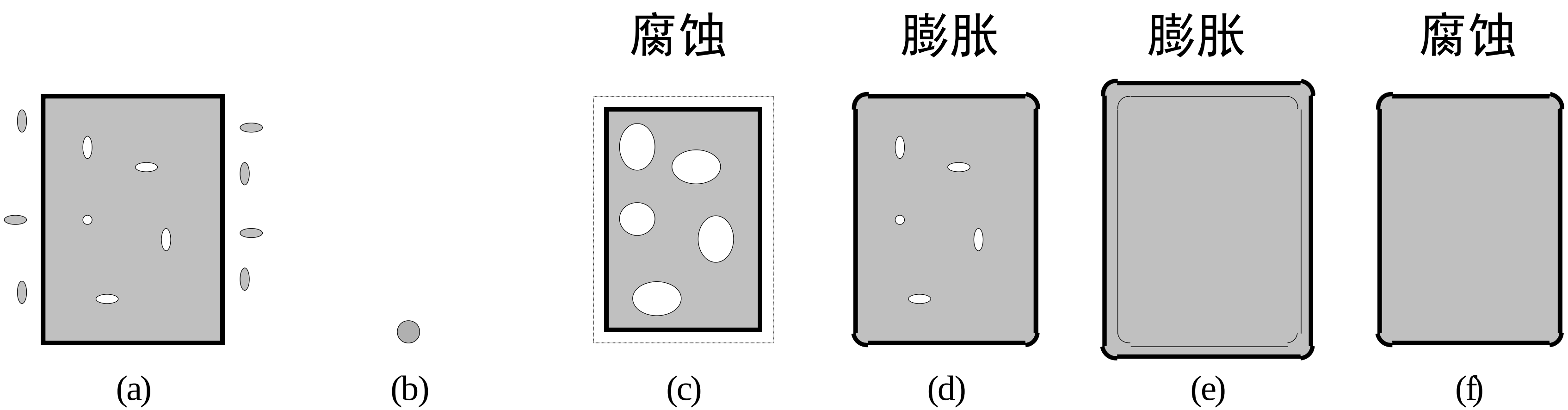
二值形态学实用算法

- 噪声滤除
- 目标检测
- 边界提取
- 区域填充
- 连通组元提取
- 区域骨架提取

二值形态学实用算法-I

- 噪声滤除
  - 先开启后闭合

$$\{[(A \ominus B)\oplus B]\oplus B\} \ominus B = (A \circ B)\cdot B$$

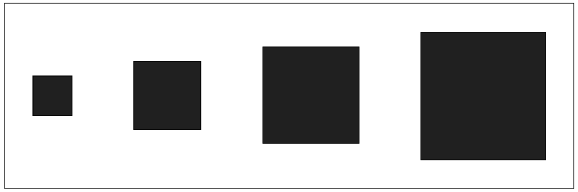


二值形态学实用算法-II

- 目标检测（击中击不中变换）
  - 3 × 3，5 × 5，7 × 7和9 × 9的实心正方形

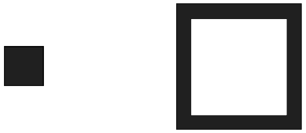
3 × 3 实心正方形

9 × 9 方框

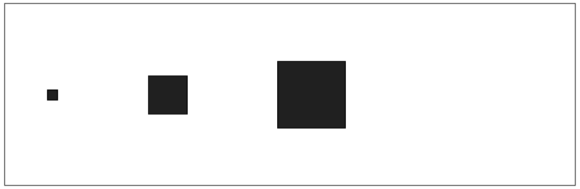


(a)

（b）：E
（c）：F



(b) (c)



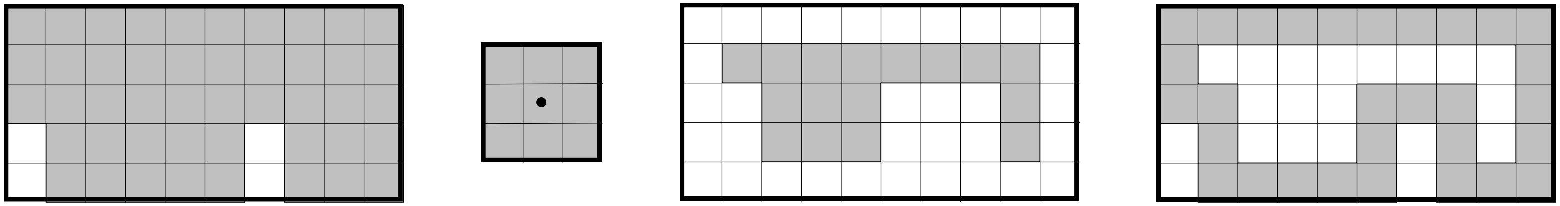
(d)

二值形态学实用算法-III

□ 边界提取

- 先用1个结构元素B腐蚀 A，再求取腐蚀结果和A的差集就可得到边界 β(A)

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



(a) (b) (c) (d)

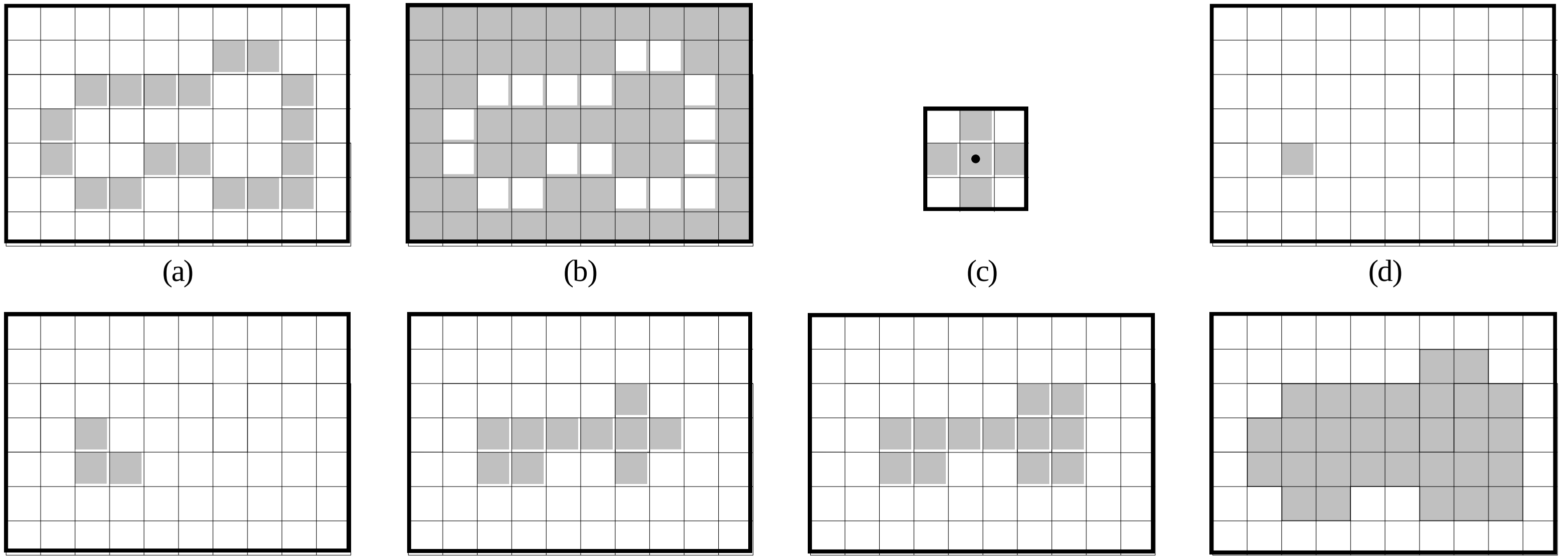
结构元素是8-连通的，而所得到的边界是4-连通的

二值形态学实用算法-IV

□ 区域填充

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

取内部一个点，按照模板膨胀，取交集，迭代多次；最后与（a）取并集



(a) (b) (c) (d) (e) (f) (g) (h)

结构元素是4-连通的，而原填充的边界是8-连通的

二值形态学实用算法-V

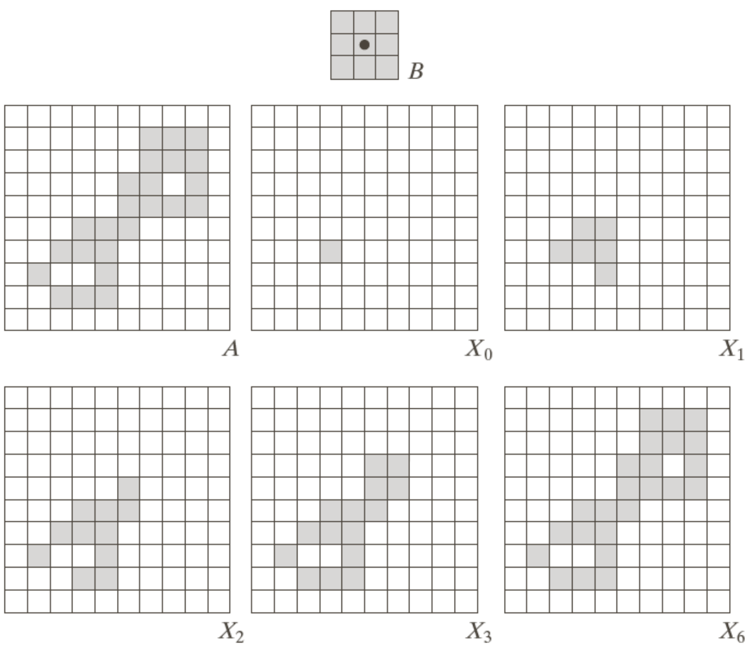
□ 连通组元提取

- 设 Y 为集合A中的一个连通组元，已知Y上的一个点记为阵列X_0。如下迭代过程可完成这一目的：

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

- 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



二值形态学实用算法-VI

- 区域骨架提取

$$S(A)=\bigcup_{k=0}^{K}S_k(A) \qquad S_k(A)=(A\ominus kB)-[(A\ominus kB)\circ B]$$

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

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

- 也可以用骨架重构A

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

二值形态学实用算法-VI

□ 区域骨架提取

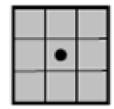
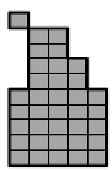
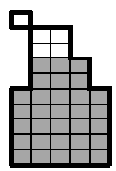
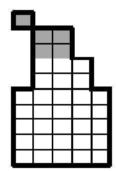
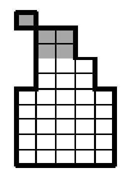
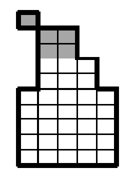
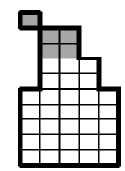
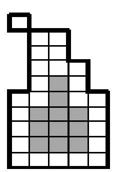
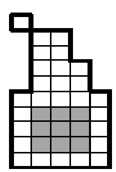
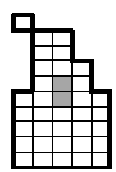
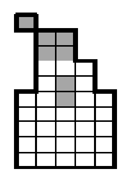
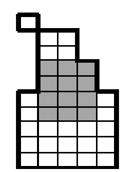
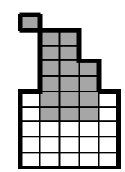


表 14.4.1　区域骨架抽取示例

列	1	2	3	4	5	6	7
运算		$A \ominus kB$	$(A \ominus kB) \circ B$	$S_k(A)$	$\bigcup_{k=0}^{K} S_k(A)$	$S_k(A) \oplus kB$	$\bigcup_{k=0}^{K} [S_k(A) \oplus kB]$
	$k=0$						
	$k=1$						
	$k=2$