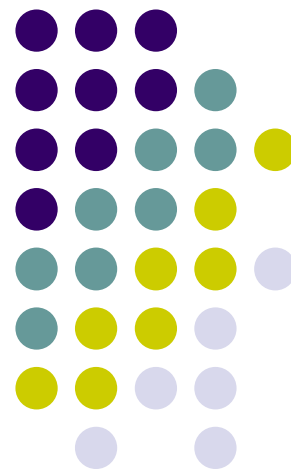


数字图像处理

第八讲 形态学处理



提纲



- 预备知识
- 腐蚀和膨胀
- 开操作和闭操作
- 击中或击不中变换
- 基本形态学算法
 - 边界提取、孔洞填充
 - 连通分量提取、凸包
 - 细化、粗化
 - 骨架、裁剪

引言



- 形态学 (morphology)
 - 生物学的一个分支
 - 研究动植物的形态和结构
- 数学形态学 (mathematical morphology)
 - 提取表示区域形状的图像成分
 - 边界、凸包、骨架
 - 输入：图像
 - 输出：图像中提取的属性

预备知识



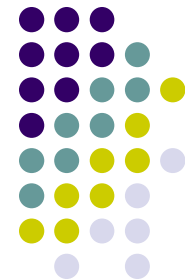
- 集合论
 - 描述形态学的数学语言
 - 集合：表示图像中的对象
 - 例如，二值图像中的所有白色像素
- 二值图像
 - 集合：属于2维整数空间 Z^2
 - 元素：二元组 (x, y)
 - 表示白色像素的坐标
- 灰度图像、 Z^3

基本集合操作



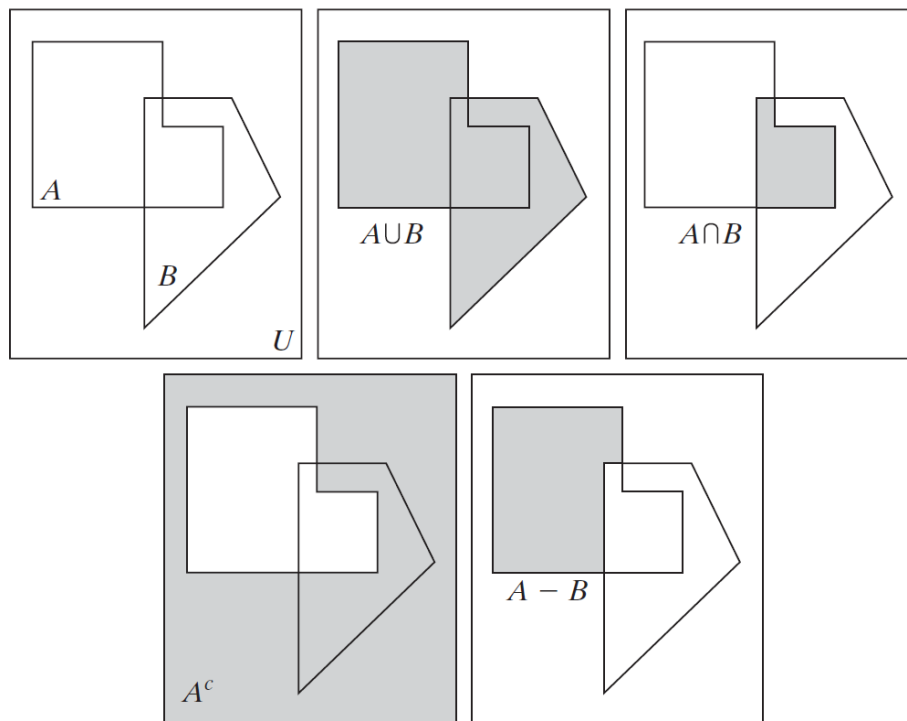
- $a = (a_1, a_2)$ 是 A 的元素: $a \in A$
- a 不是 A 的元素: $a \notin A$
- 空集: \emptyset
- 全集: U
- A 是 B 的子集: $A \subseteq B$
- 集合 A 和 B 的并集: $A \cup B$
- 集合 A 和 B 的交集: $A \cap B$
- 集合 A 和 B 互斥: $A \cap B = \emptyset$

基本集合操作

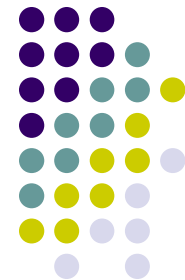


- 集合 A 的补集: $A^c = \{w | w \notin A\} = U - A$
- 集合 A 和 B 的差:

$$A - B = \{w | w \in A, w \notin B\} = A \cap B^c$$

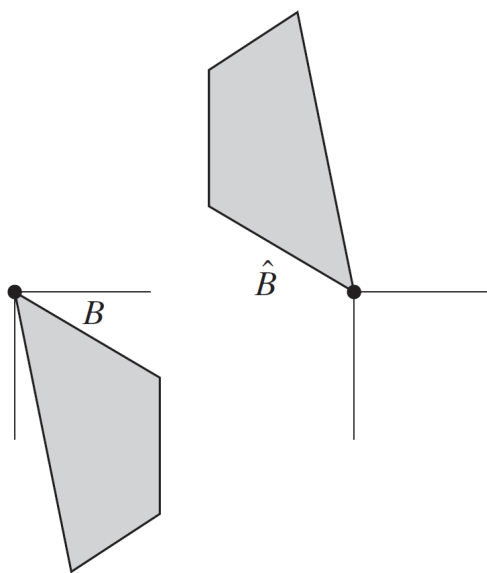


集合操作



- 集合的反射

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



集合操作

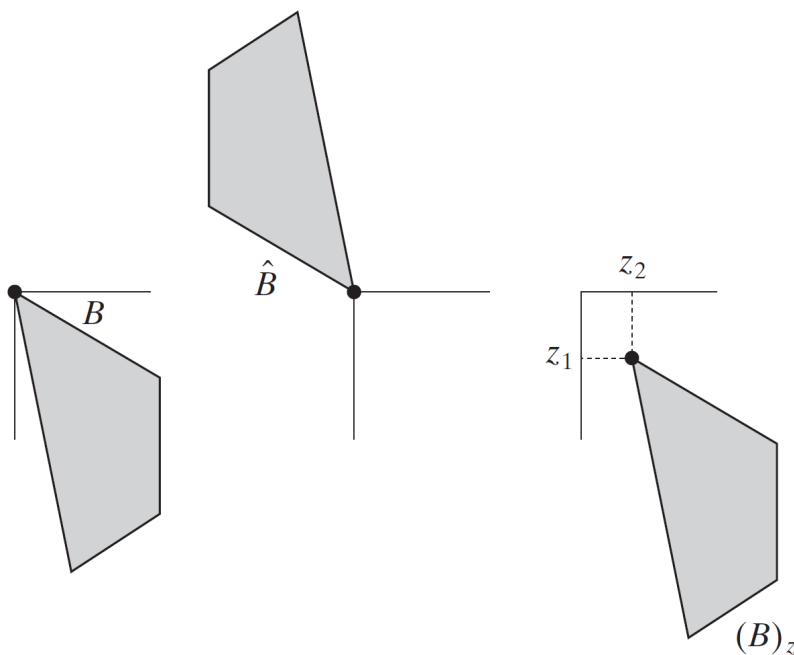


- 集合的反射

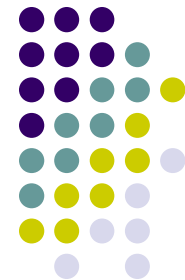
$$\hat{B} = \{w | w = -b, \text{ for } b \in B\}$$

- 集合的平移

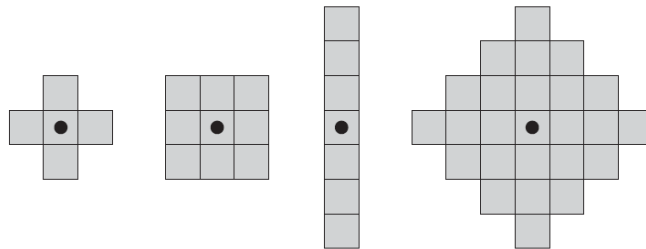
$$(B)_z = \{c | c = b + z, \text{ for } b \in B\}$$



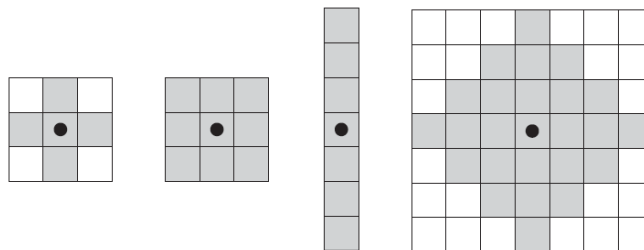
基于结构元的操作



- 结构元 (structuring elements)
 - 用于研究图像性质的小集合或子图像



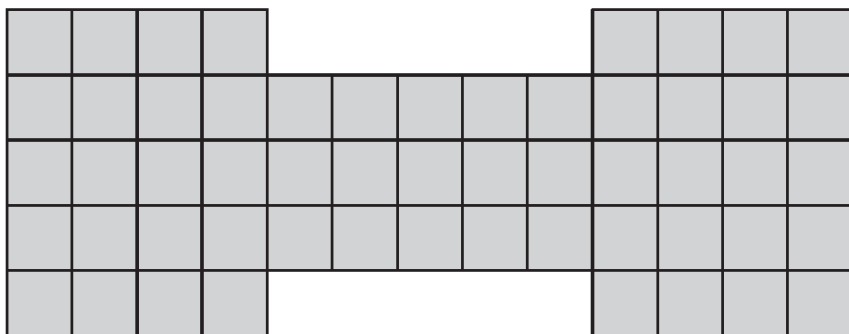
- 黑点表示结构元的原点
- 通常用矩形表示
 - 填充背景



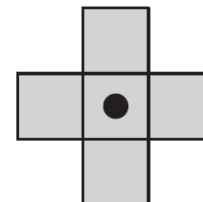
基于结构元的操作



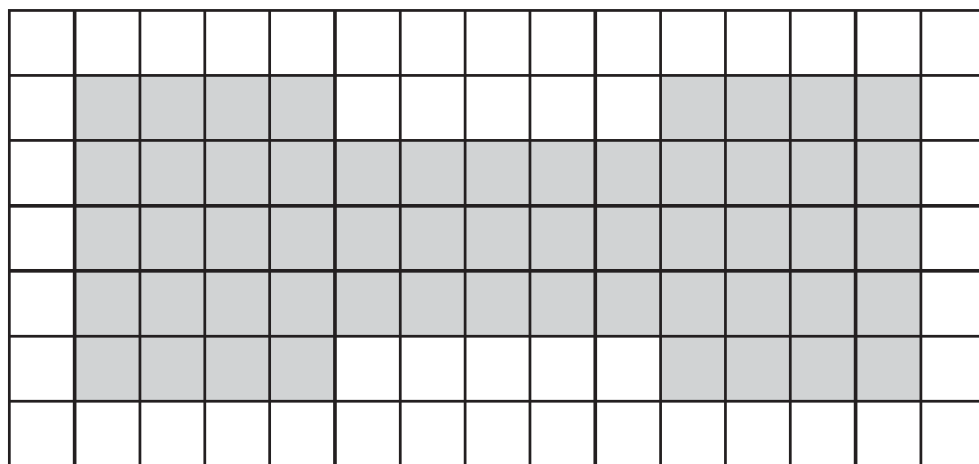
集合A



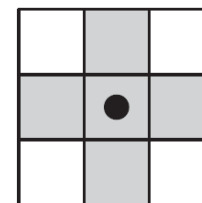
结构元B



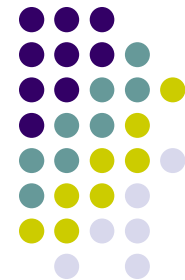
● 填充成矩形



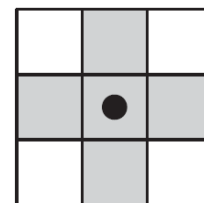
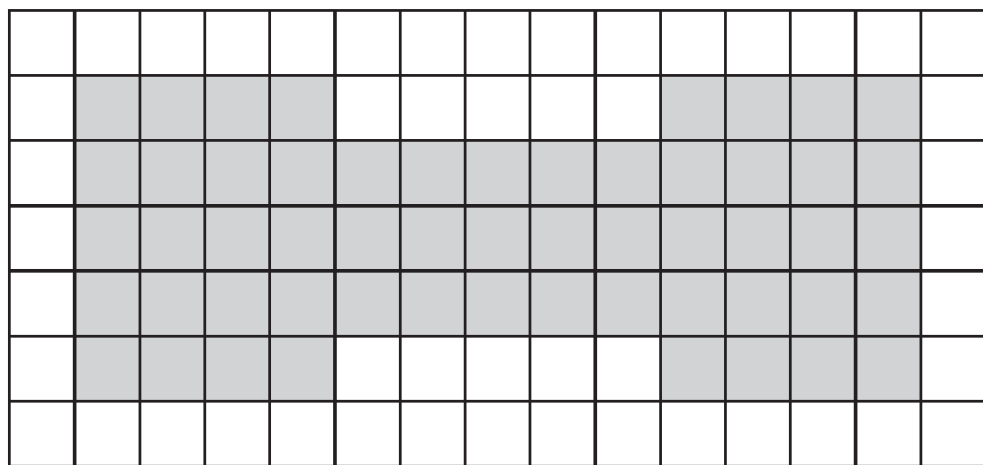
添加边框以
容纳结构元



基于结构元的操作



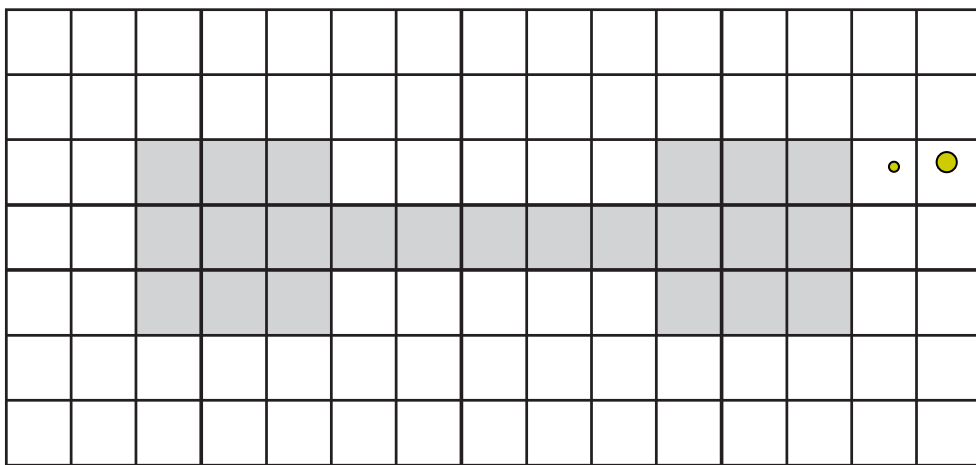
- 利用结构元构造一个新集合 C
 1. 用结构元 B 覆盖集合 A
 2. 在当前位置（ B 的原点），如果 A 完全包含 B ，则当前位置属于 C



基于结构元的操作



- 利用结构元构造一个新集合 C
 1. 用结构元 B 覆盖集合 A
 2. 在当前位置（ B 的原点），如果 A 完全包含 B ，则当前位置属于 C
 3. 移动结构元 B ，使其原点访问 A 中的所有元素



提纲



- 预备知识
- 腐蚀和膨胀
- 开操作和闭操作
- 击中或击不中变换
- 基本形态学算法
 - 边界提取、孔洞填充
 - 连通分量提取、凸包
 - 细化、粗化
 - 骨架、裁剪

腐蚀



- 集合 B 对集合 A 的腐蚀 (erosion)

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

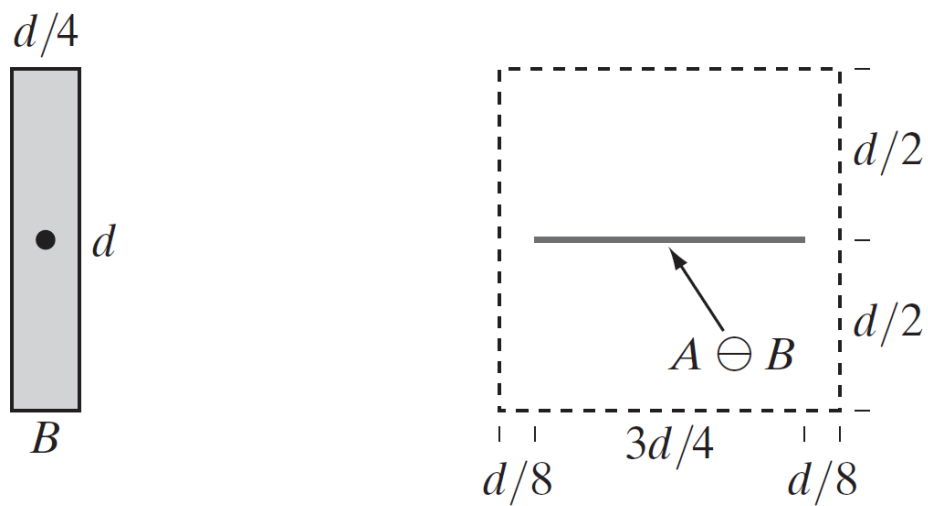
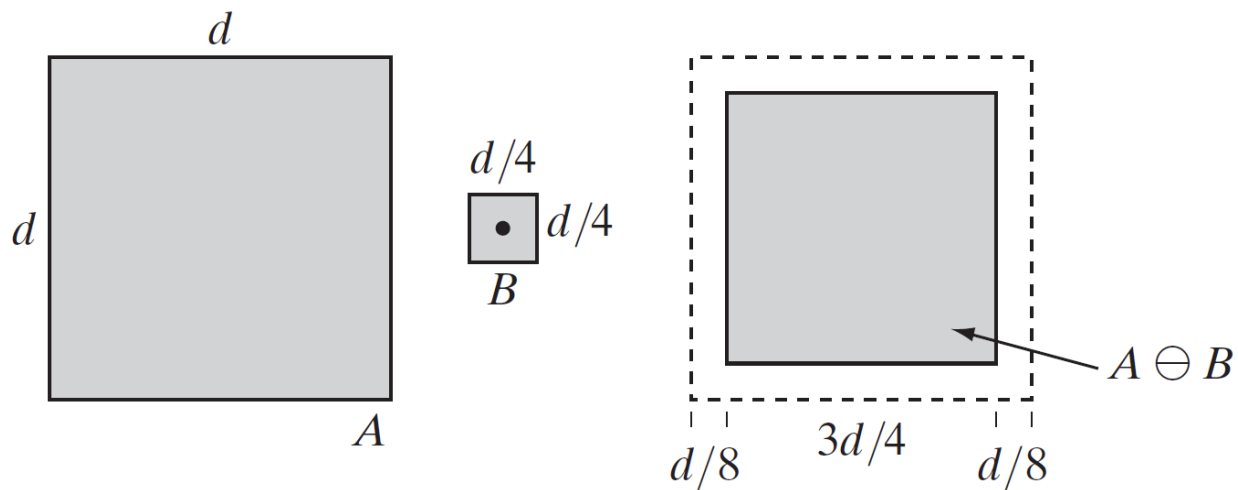
- $(B)_z$ 表示把集合 B 平移到坐标 z
- 通常假设集合 B 为结构元
- $(B)_z$ 意味着把 B 的原点平移到 z

- 等价定义

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

- A^c 表示集合 A 的补集

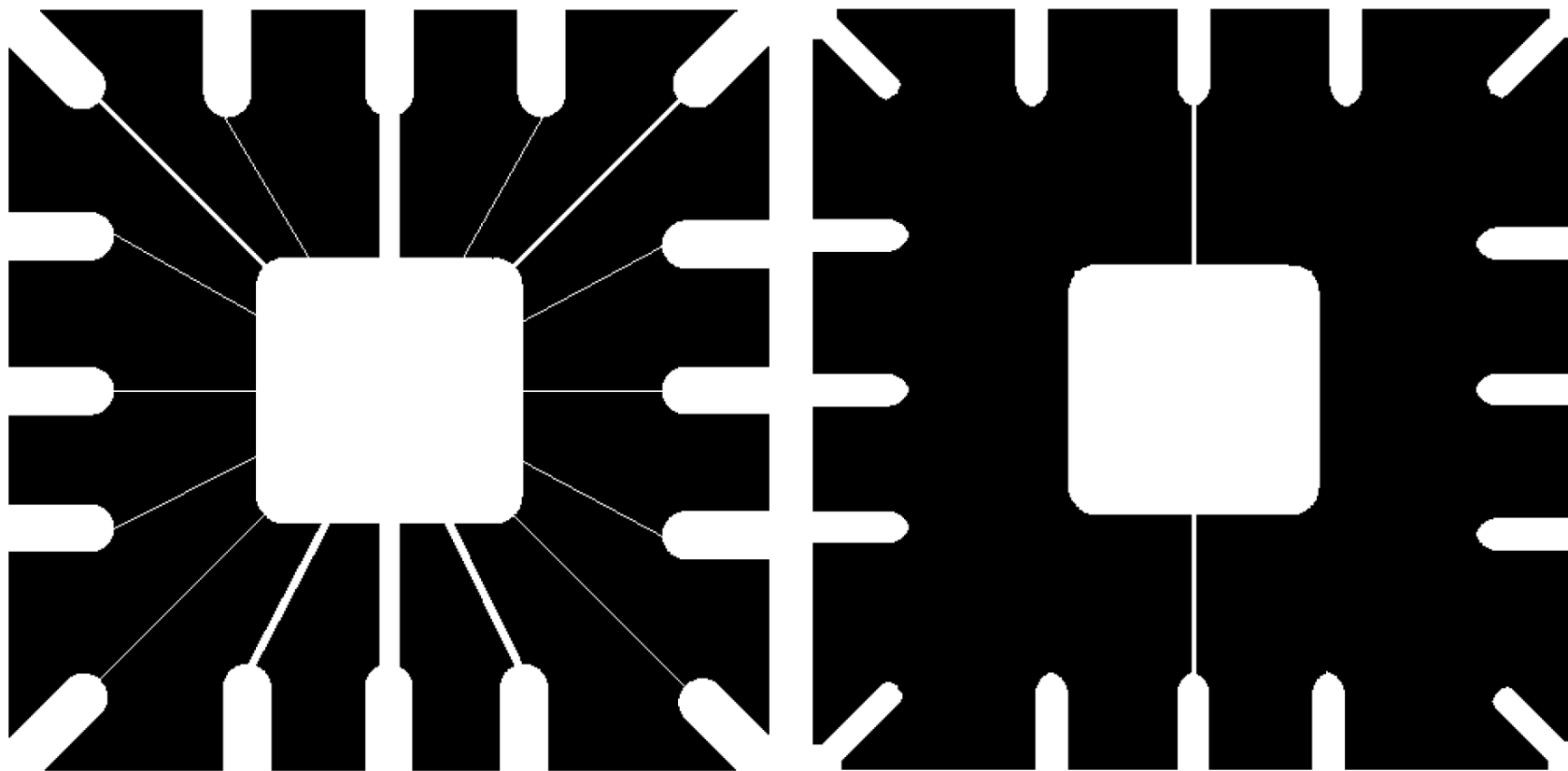
举例



举例



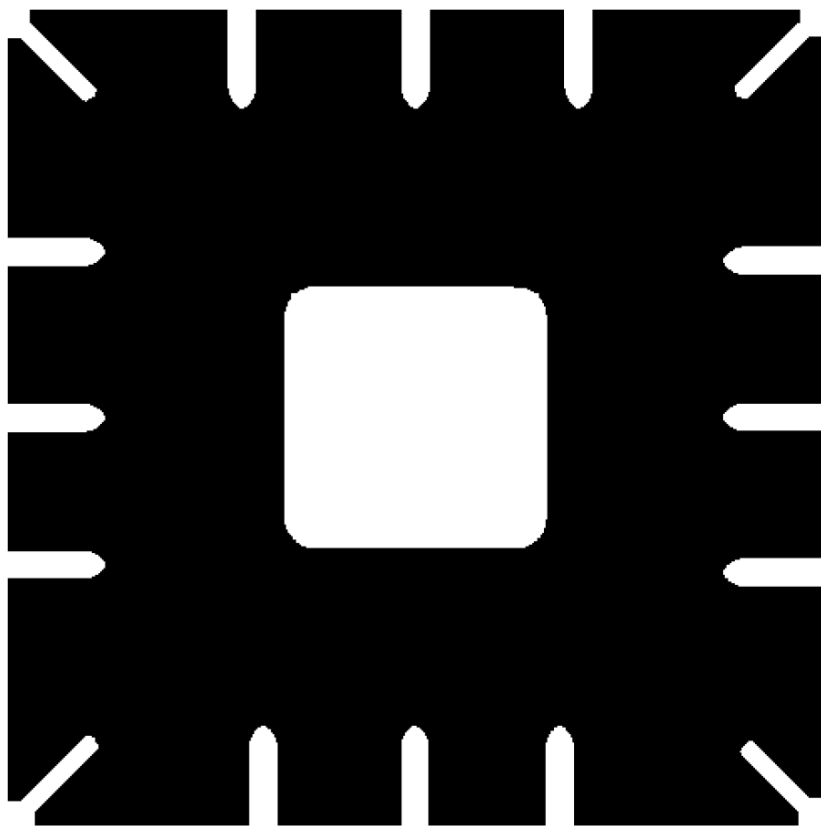
- 去掉连接线



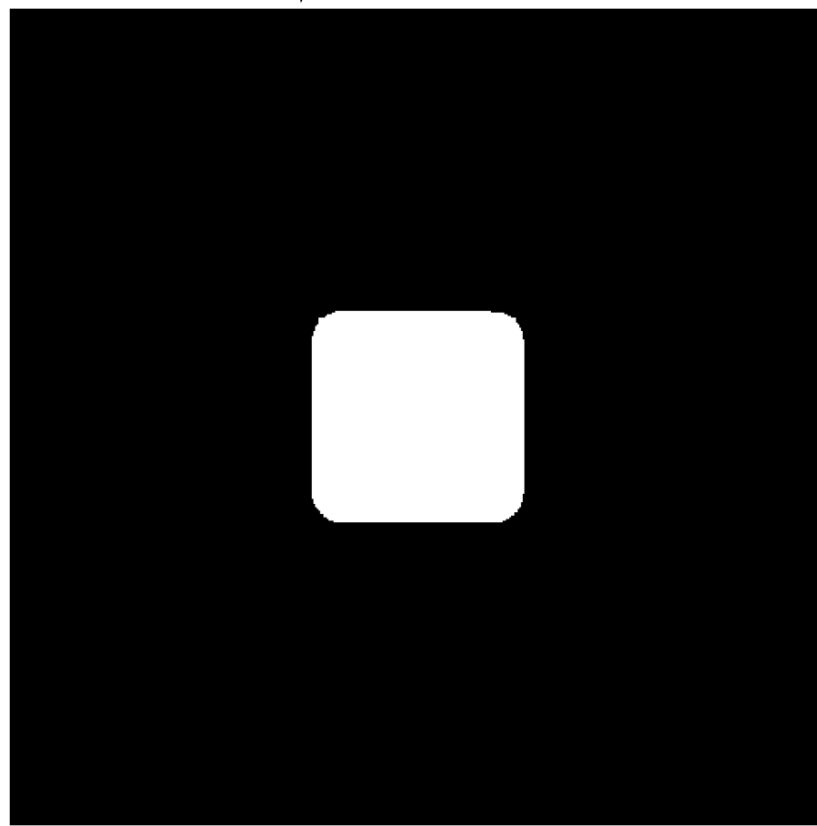
用 11×11 的方框腐蚀

举例

- 去掉连接线



用 15×15 的方框腐蚀



用 45×45 的方框腐蚀

膨胀



- 集合 B 对集合 A 的膨胀 (dilation)

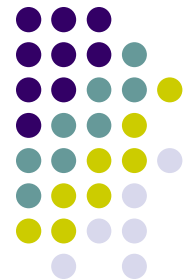
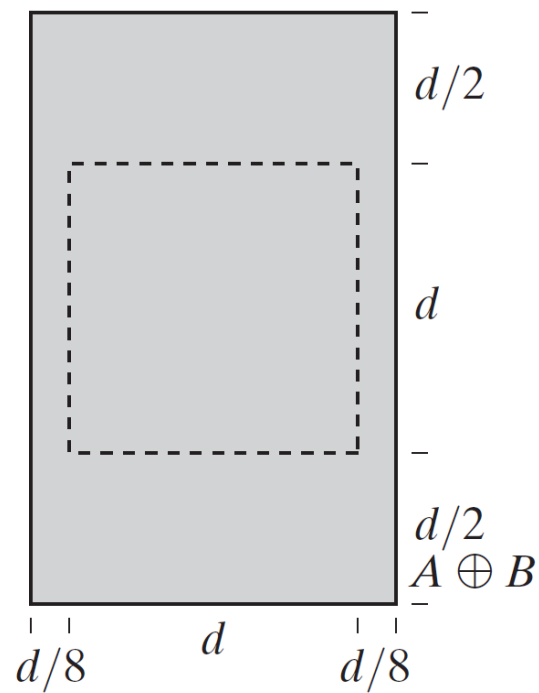
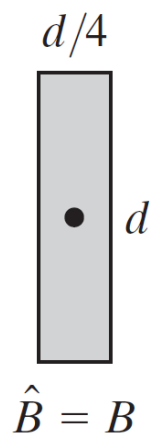
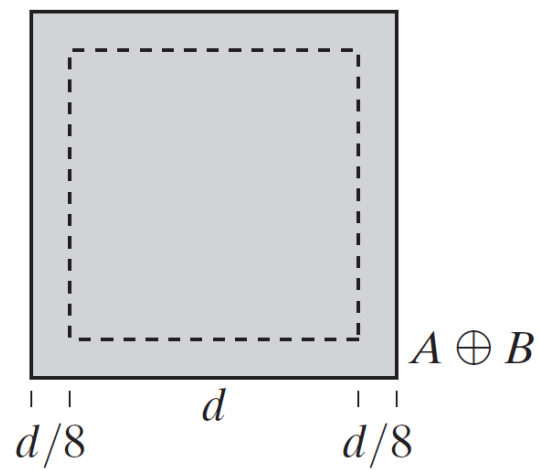
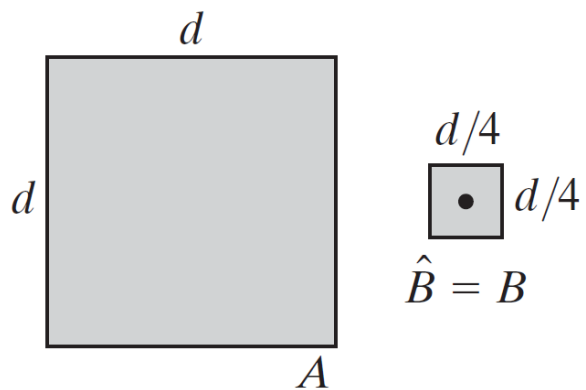
$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\}$$

- \hat{B} 表示集合 B 的反射
- $(\hat{B})_z$ 表示把集合 \hat{B} 平移到坐标 z
- 通常假设集合 B 为结构元

- 等价定义

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

举例



举例

比低通滤波器
更简单、直接

输出仍然是
二值图像

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.



0	1	0
1	1	1
0	1	0

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.



最长间距是2个像素

对偶性



- 公式

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

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

- 证明

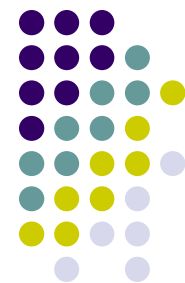
$$(A \ominus B)^c = \{z | (B)_z \subseteq A\}^c$$

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

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

$$= A^c \oplus \hat{B}$$

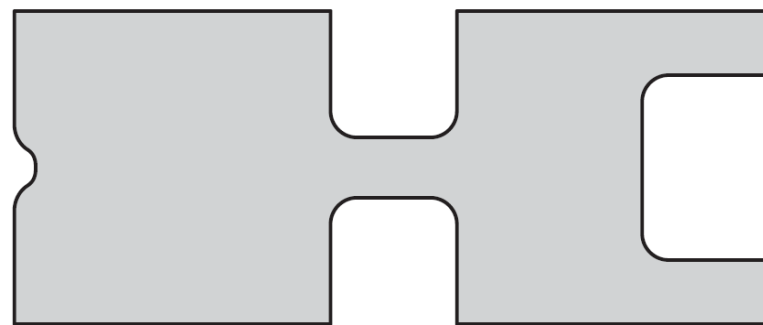
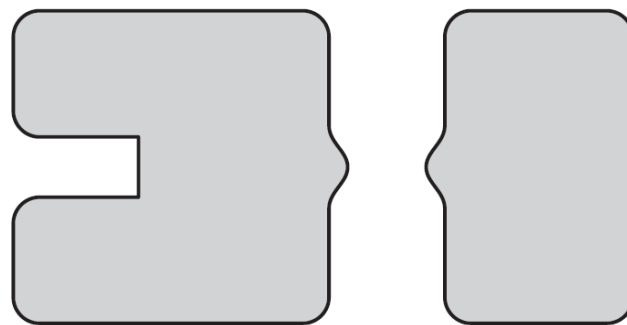
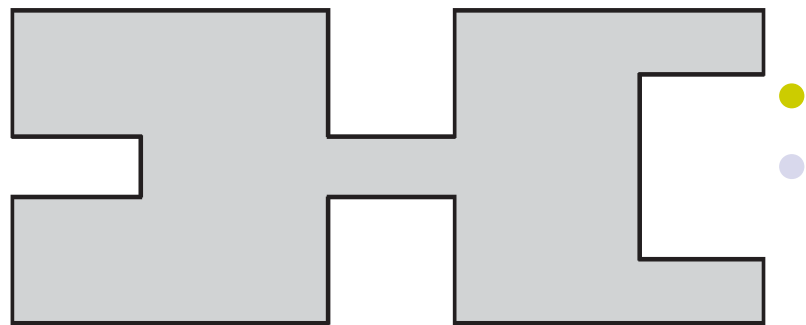
提纲



- 预备知识
- 腐蚀和膨胀
- 开操作和闭操作
- 击中或击不中变换
- 基本形态学算法
 - 边界提取、孔洞填充
 - 连通分量提取、凸包
 - 细化、粗化
 - 骨架、裁剪

开操作和闭操作

- 开操作 (opening)
 - 平滑物体的轮廓
 - 断开窄的连接
 - 消除细的突出
- 闭操作 (closing)
 - 平滑部分轮廓
 - 熔合窄的间断和长沟壑
 - 消除小孔洞
 - 填补轮廓中的缝隙



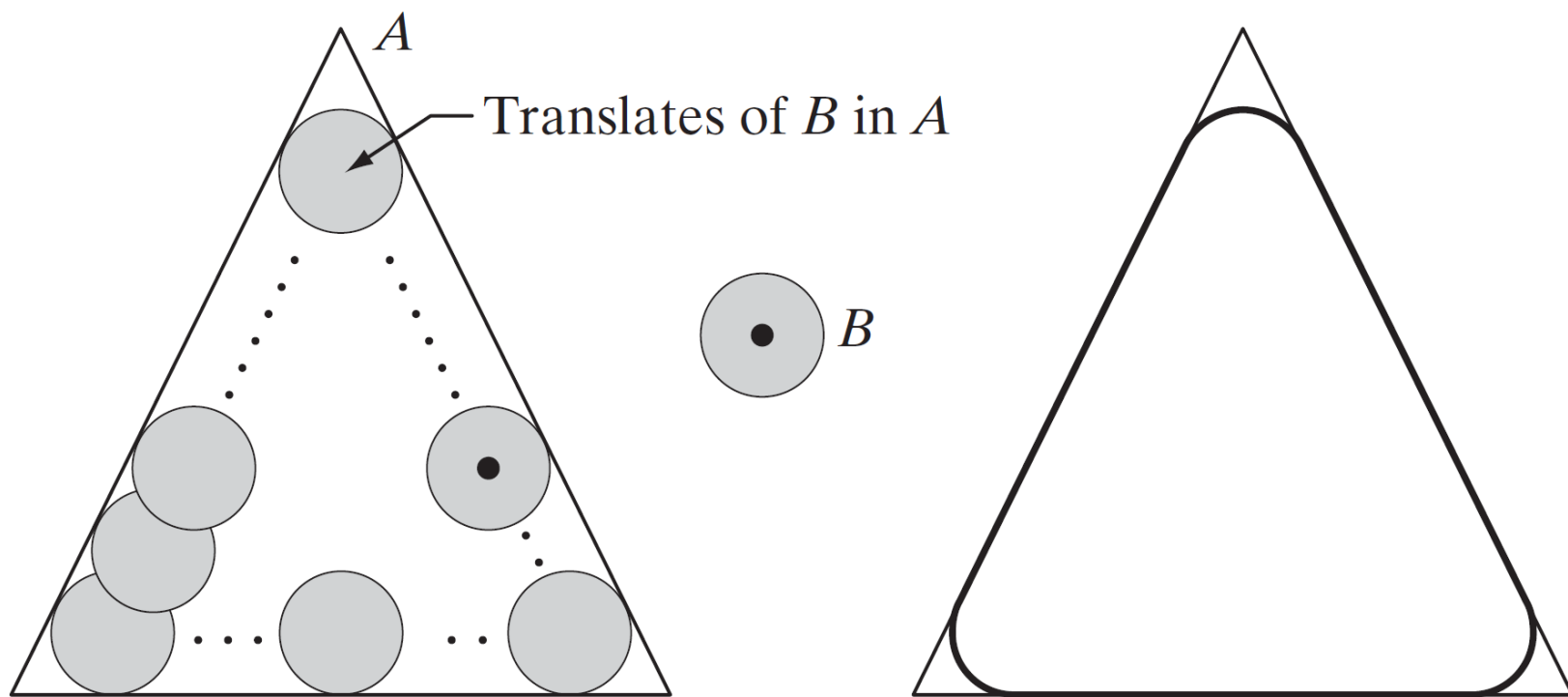
开操作

- 结构元 B 对集合 A 的开操作

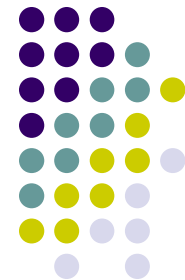
$$A \circ B = (A \ominus B) \oplus B$$

- 先用 B 腐蚀 A ，然后再用 B 对结果进行膨胀

在 A 的边界内侧
滚动 B ， B 的最
远点决定了轮廓



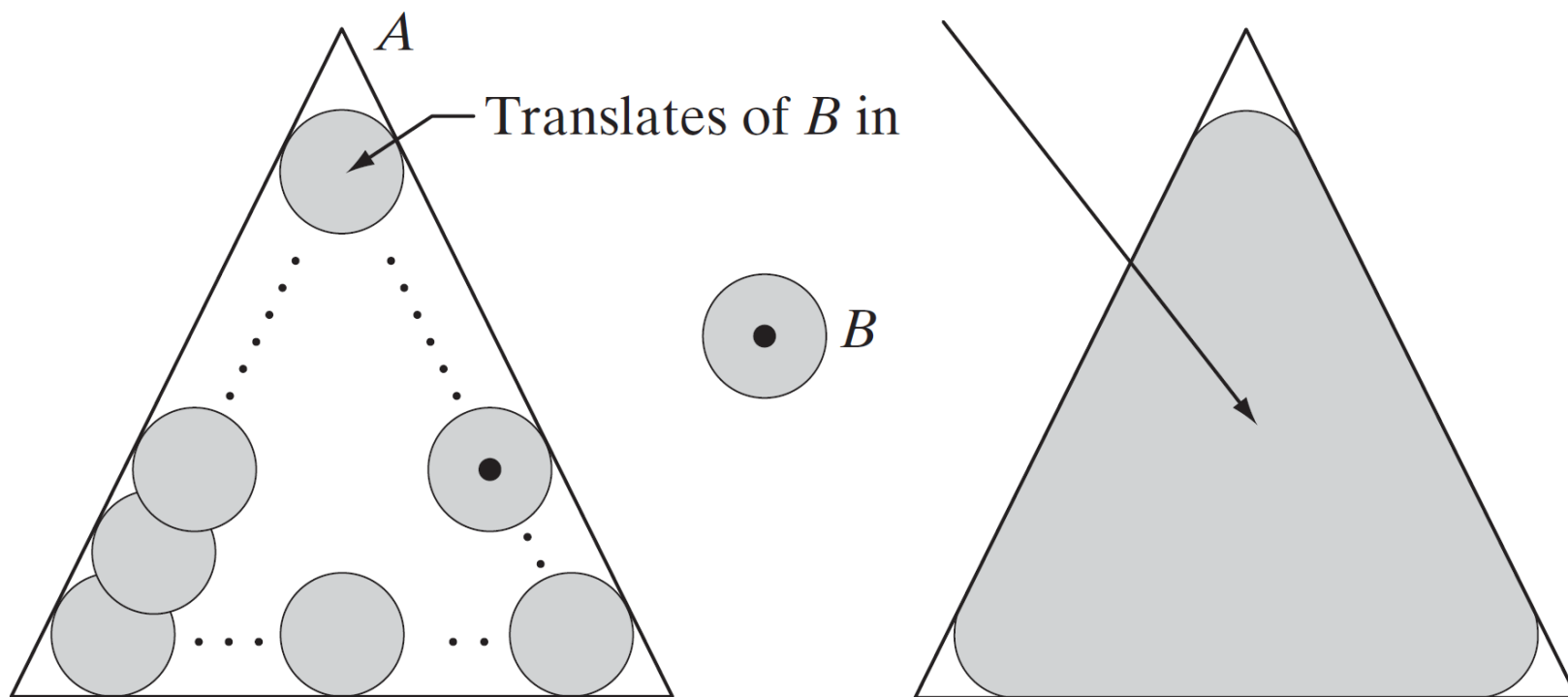
开操作



- 结构元 B 对集合 A 的开操作

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

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

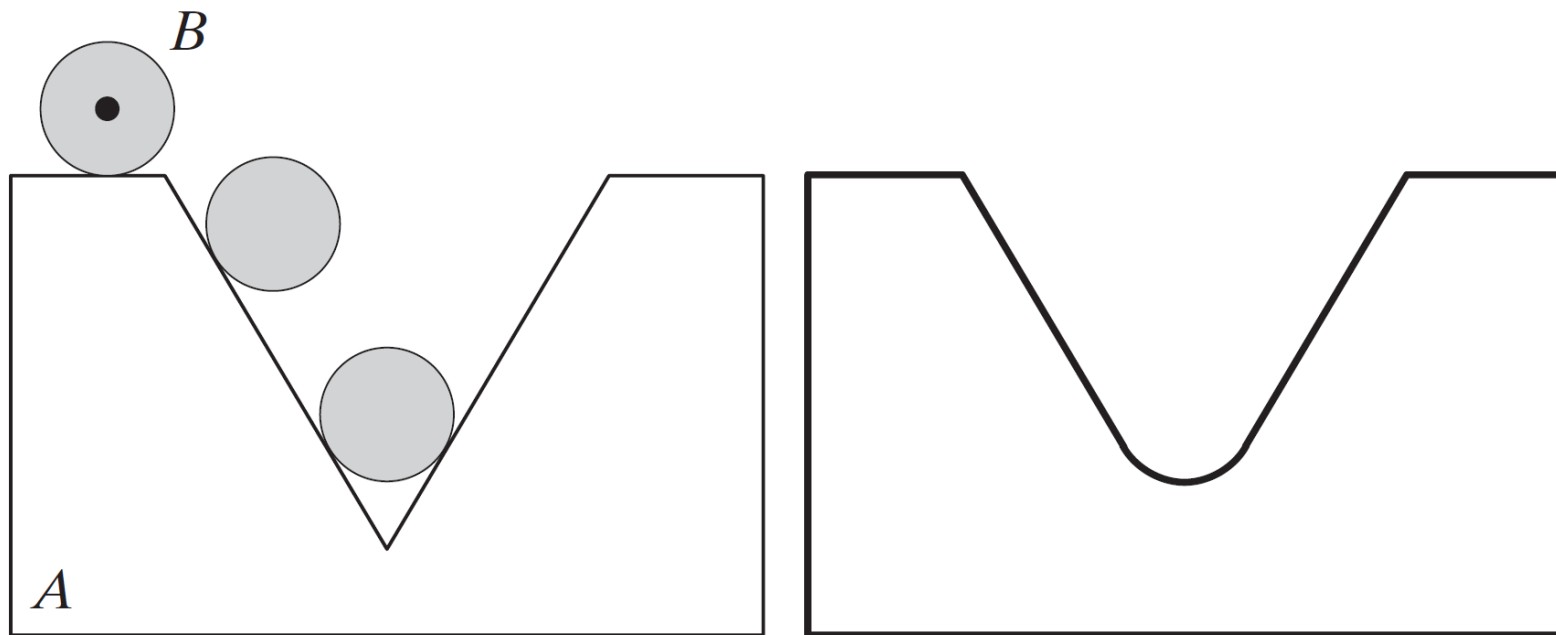


闭操作

- 结构元 B 对集合 A 的闭操作

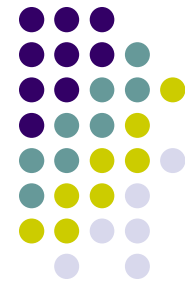
$$A \bullet B = (A \oplus B) \ominus B$$

- 先用 B 膨胀 A ，然后再用 B 对结果进行腐蚀



在 A 的边界外侧
滚动 B ， B 的最
近点决定了轮廓

闭操作

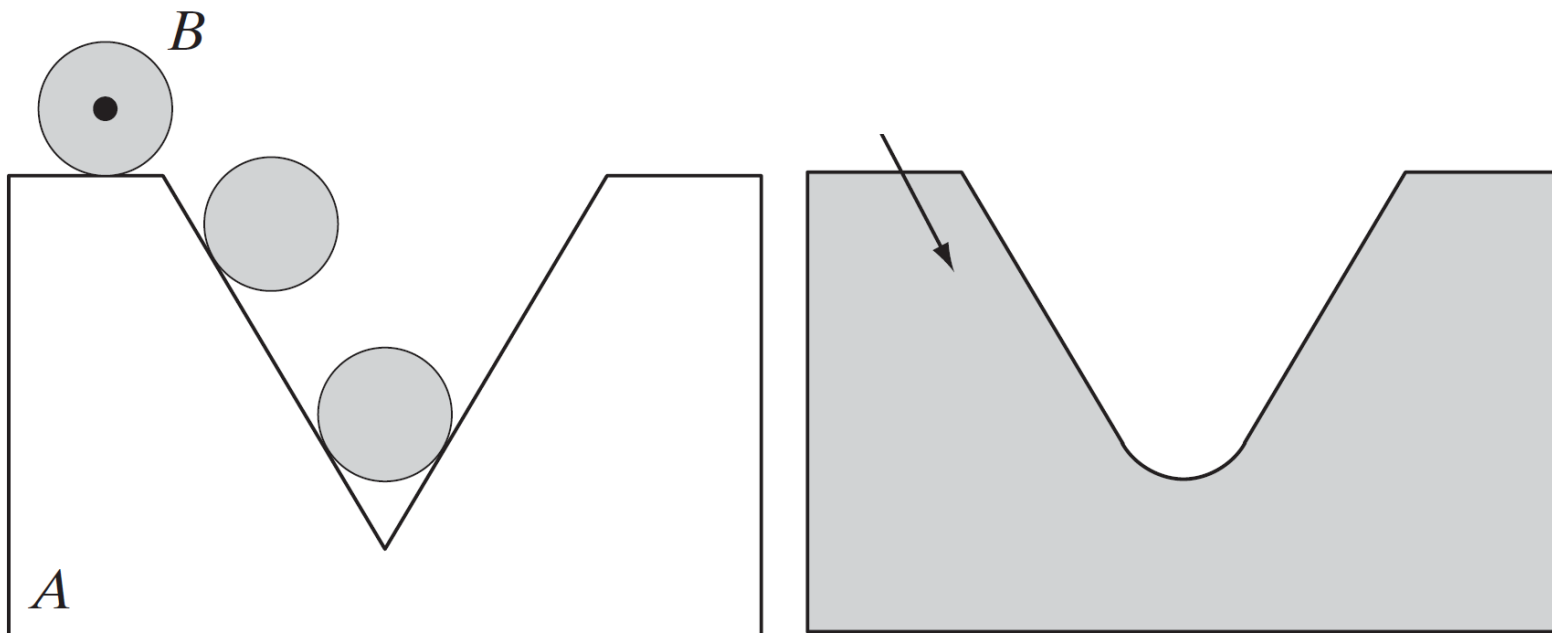


- 结构元 B 对集合 A 的闭操作

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

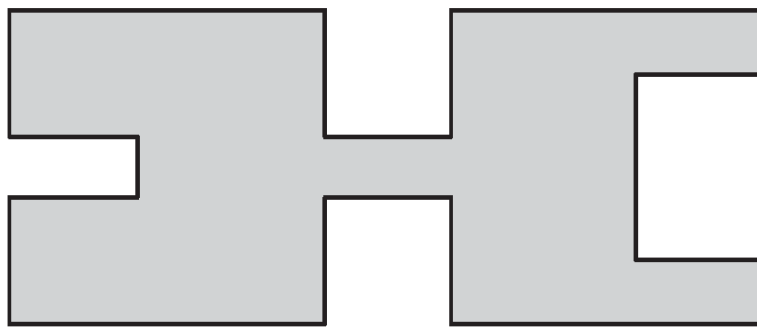
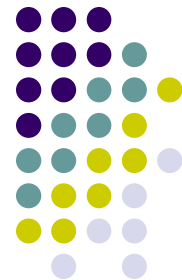
- 先用 B 膨胀 A ，然后再用 B 对结果进行腐蚀

$$A \bullet B = \{w | w \in (B)_z \Rightarrow (B)_z \cap A \neq \emptyset\}$$

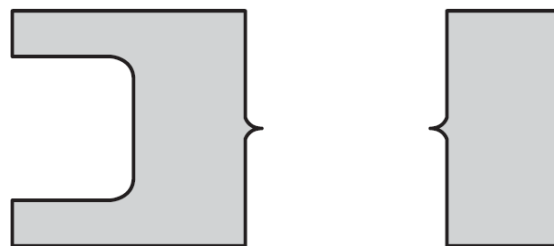
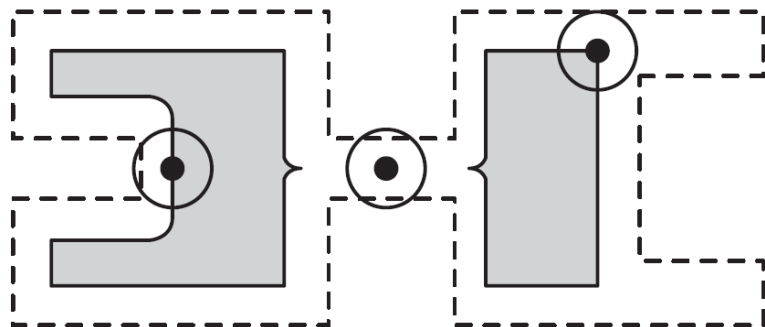


举例

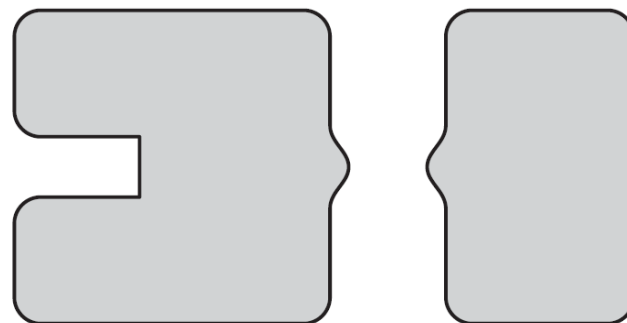
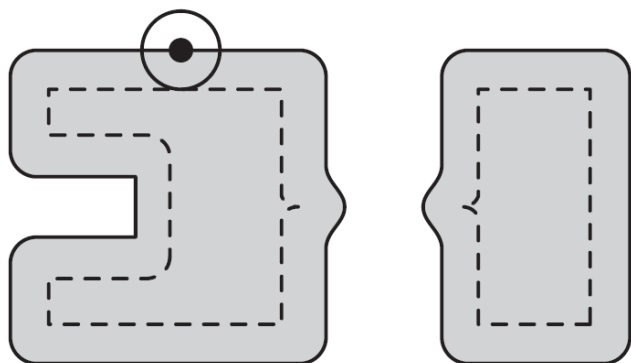
开操作



A



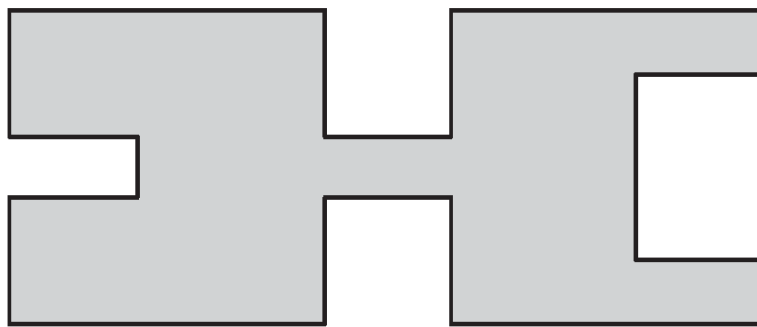
$A \ominus B$



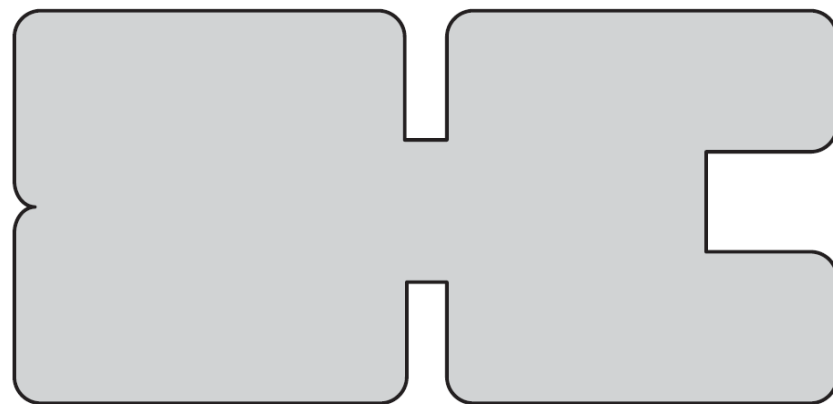
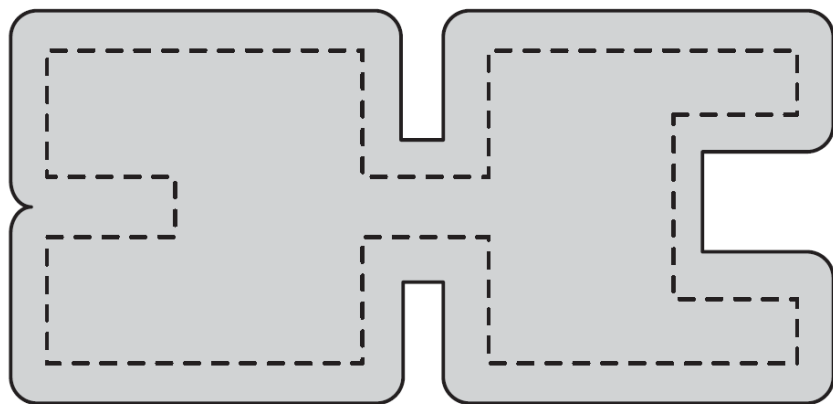
$A \circ B = (A \ominus B) \oplus B$

举例

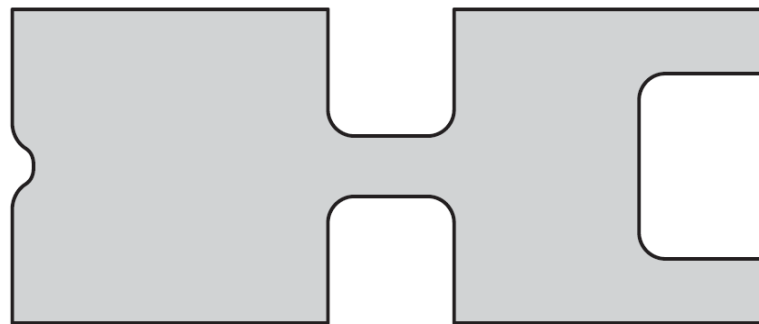
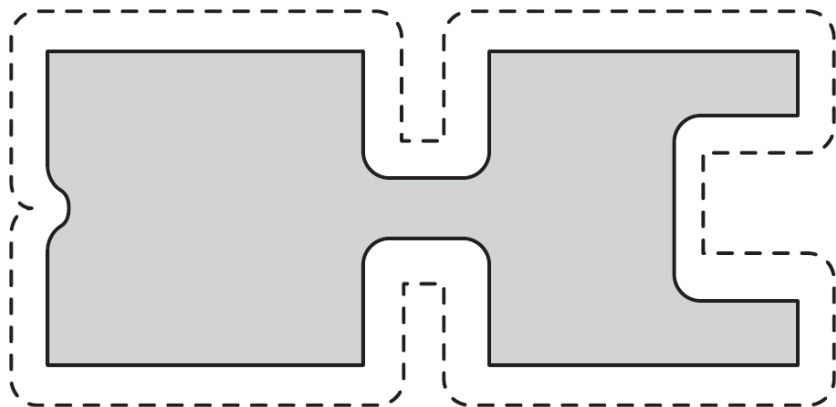
闭操作



A



$A \oplus B$



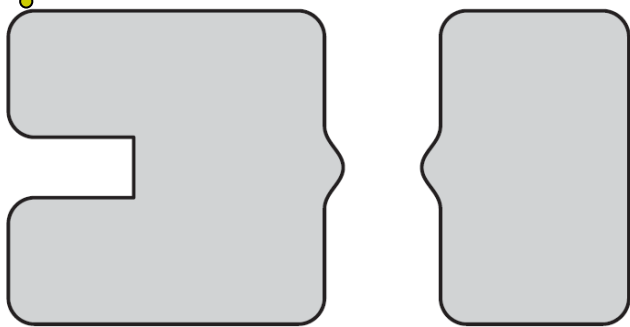
$A \cdot B = (A \oplus B) \ominus B$

举例

● 对比

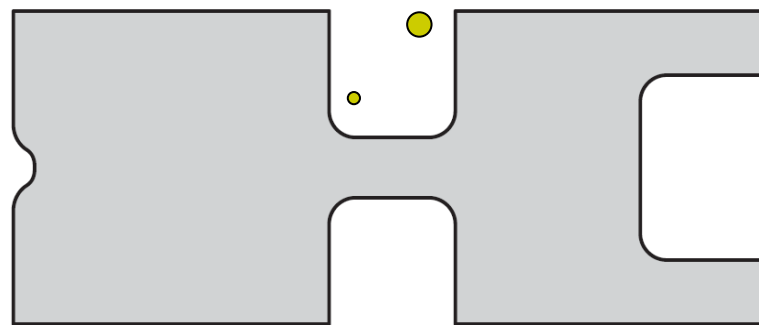
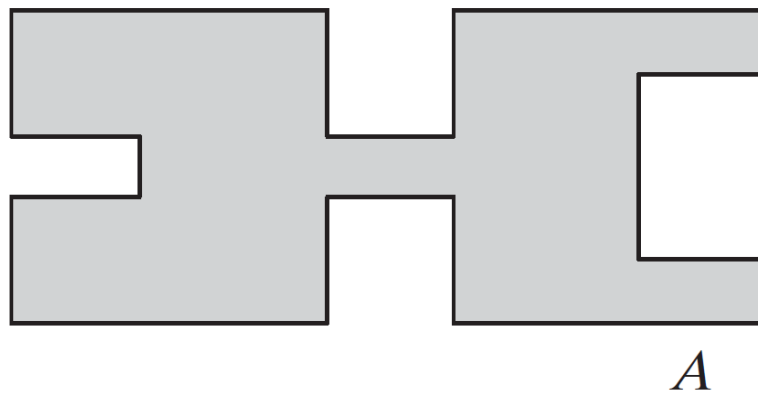
方向向外的角变圆

方向向内的角变圆



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

开操作



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

闭操作

性质



- 对偶性

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

- 开操作

1. $A \circ B$ 是 A 的子集
2. 如果 C 是 D 的子集, 那么 $C \circ B$ 是 $D \circ B$ 的子集
3. $(A \circ B) \circ B = A \circ B$

- 闭操作

1. A 是 $A \bullet B$ 的子集
2. 如果 C 是 D 的子集, 那么 $C \bullet B$ 是 $D \bullet B$ 的子集
3. $(A \bullet B) \bullet B = A \bullet B$

举例

● 去噪

结构元

1	1	1
1	1	1
1	1	1

B

1. 黑色背景中的白噪音被去除
2. 白色指纹中的黑噪声被加强

A



含噪声的指纹

$A \ominus B$



腐蚀

举例

去噪

1. 白色指纹中的黑噪声被削弱
2. 指纹纹路产生了断裂

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



开操作

1. 纹路中的大部分断裂被修复
2. 纹路变得更粗

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



开操作的膨胀

举例

● 去噪

1. 纹路变细
2. 噪声被消除
3. 存在部分断裂

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



开操作的闭操作



含噪声的指纹



提纲

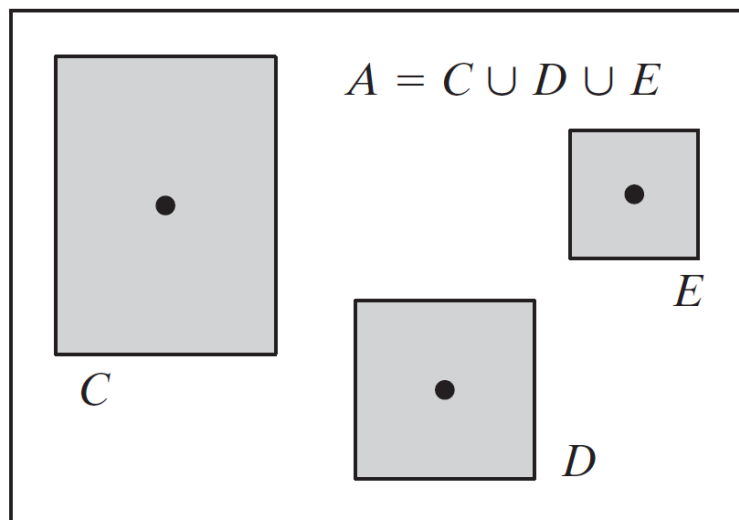


- 预备知识
- 腐蚀和膨胀
- 开操作和闭操作
- 击中或击不中变换
- 基本形态学算法
 - 边界提取、孔洞填充
 - 连通分量提取、凸包
 - 细化、粗化
 - 骨架、裁剪

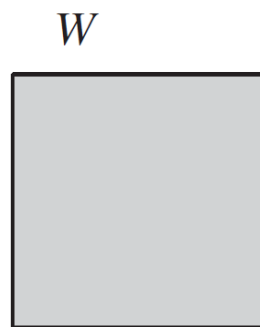
击中或击不中变换



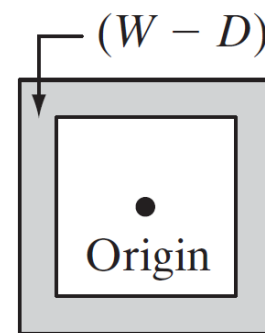
- 击中或击不中变换 (hit-or-miss transform)
 - 用于检测图像中的形状
- 检测形状 D



包含三个形状的组合 A



包含 D 的
小窗口 W



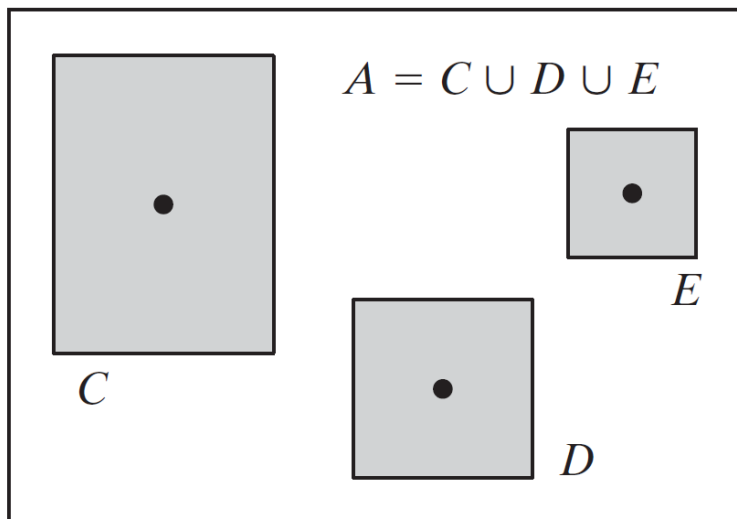
相对 W 而言，
 D 的局部背景

击中或击中不中变换

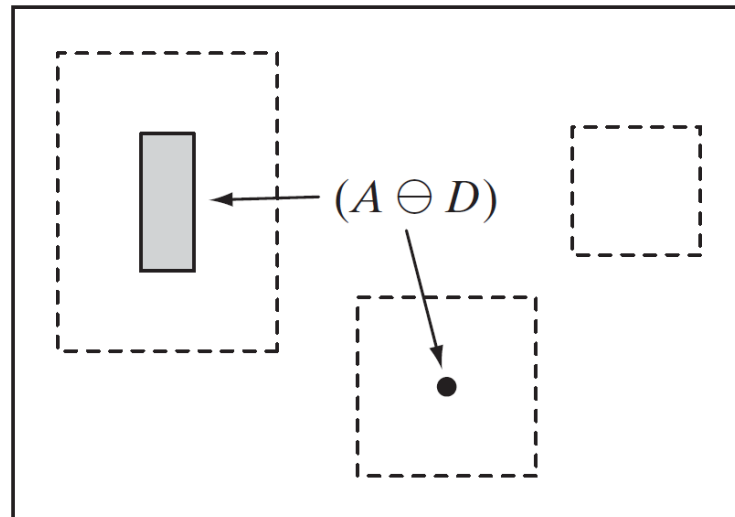


- 击中或击中不中变换 (hit-or-miss transform)
 - 用于检测图像中的形状
- 检测形状 D

表示 D 的匹配 (击中)



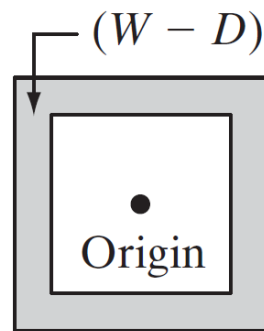
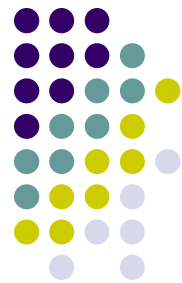
包含三个形状的组合 A



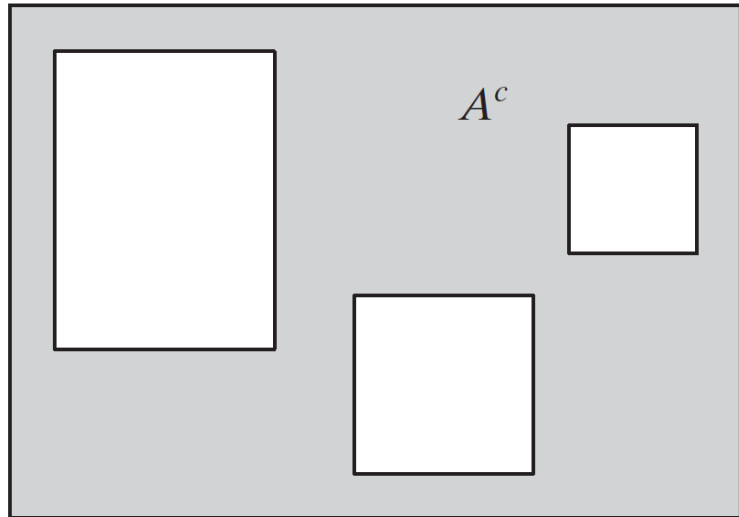
D 对 A 的腐蚀

击中或击中不中变换

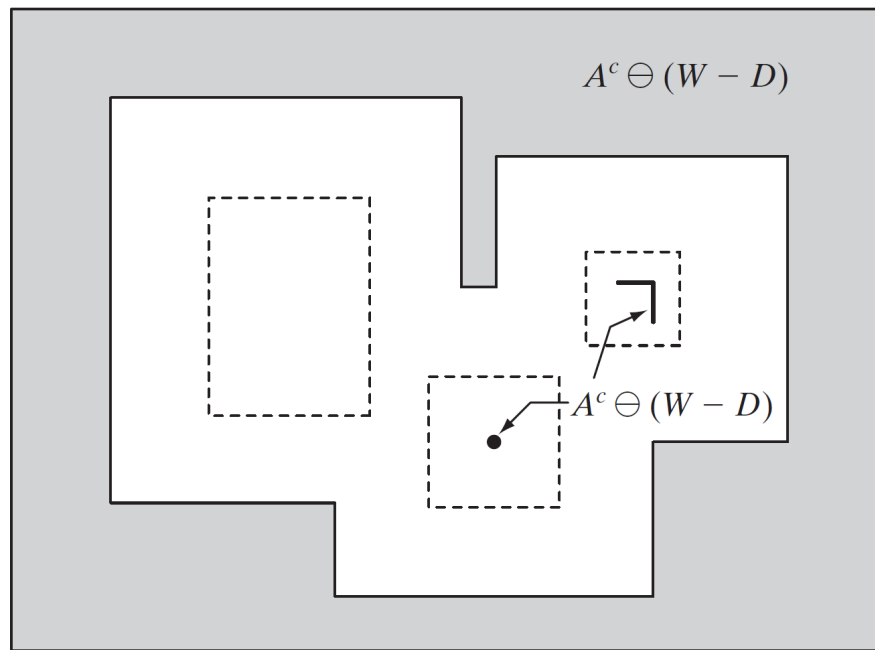
- 击中或击中不中变换
 - 用于检测图像中的形状
- 检测形状 D



相对 W 而言， D 的局部背景

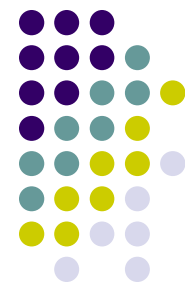


集合 A 的补集 A^c

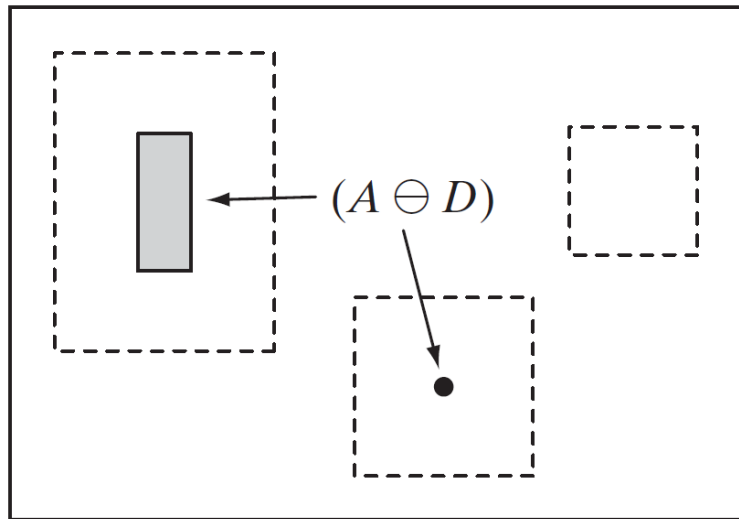


$W - D$ 对 A^c 的腐蚀

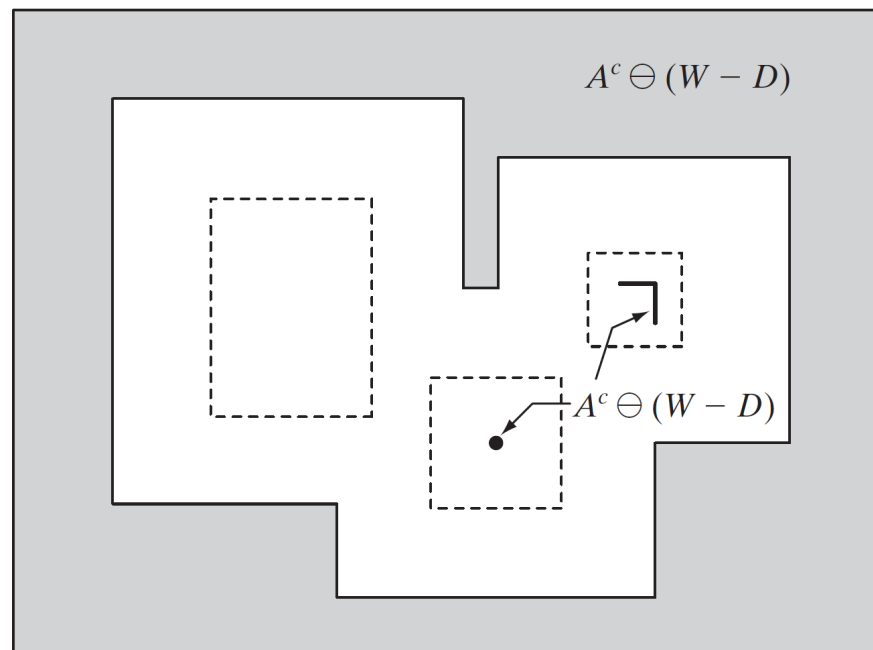
击中或击不中变换



- 击中或击不中变换 (hit-or-miss transform)
 - 用于检测图像中的形状
- 检测形状 D



D 对 A 的腐蚀

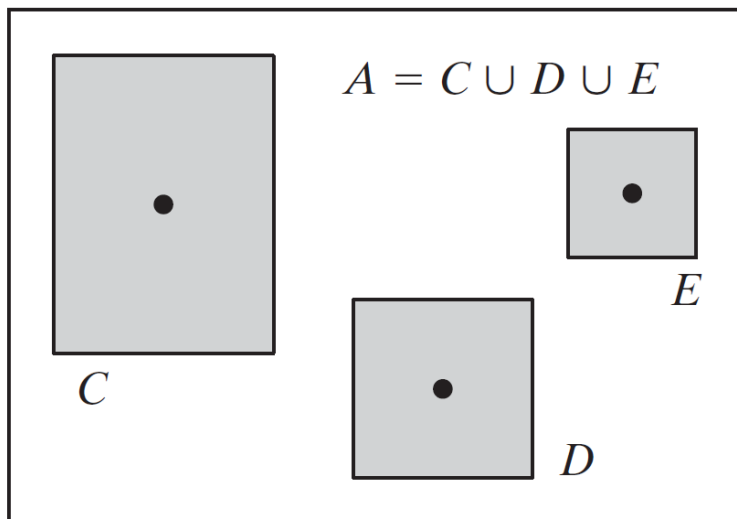


$W - D$ 对 A^c 的腐蚀

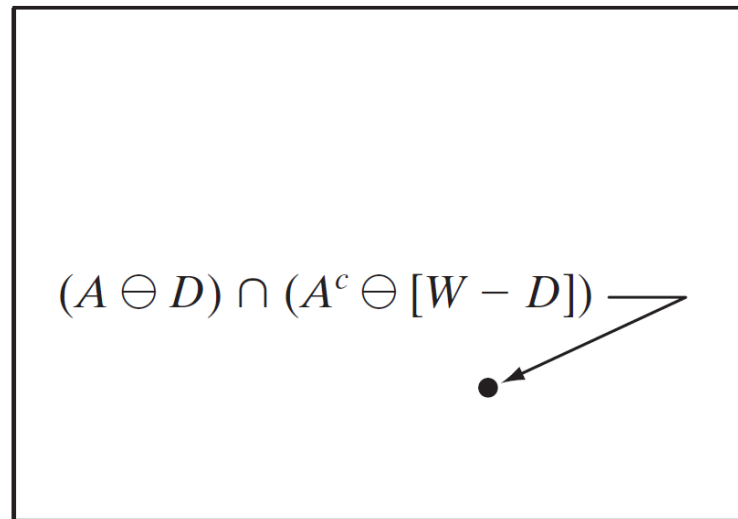
击中或击不中变换



- 击中或击不中变换 (hit-or-miss transform)
 - 用于检测图像中的形状
- 检测形状 D



包含三个形状的集合 A



交集确定 D 的位置

击中或击不中变换



- 击中或击不中变换 (hit-or-miss transform)
 - 用于检测图像中的形状
- 集合 B 在 A 中的匹配

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

- B 表示集合 D 及其背景

- 背景使物体独立出现
- 无背景时变成腐蚀

- 令 $B = (B_1, B_2)$

- $B_1 = D$ 表示物体, $B_2 = W - D$ 表示背景

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

- B_1 在 A 中匹配, B_2 在 A^c 中匹配

击中或击不中变换



- 击中或击不中变换 (hit-or-miss transform)
 - 用于检测图像中的形状
- 集合 B 在 A 中的匹配

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

- B 表示集合 D 及其背景
- 令 $B = (B_1, B_2)$

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

- 等价形式

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