

# 第九章 图像表示与描述

- 1、 形态学图像处理算法
- 2、 腐蚀
- 3、 膨胀
- 4、 开操作与闭操作

# 1、形态学图像处理算法

将数学形态学作为工具，从图像中提取用于表达和描绘区域形状的图像成分，如边界，骨架等。



形态学提取边界



形态学滤波的应用

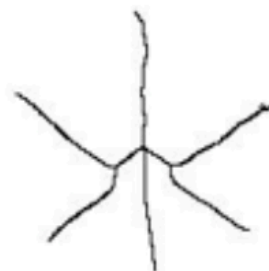
形态学图像处理的应用可以简化图像数据，保持它们基本的形状特性，并除去不相干的结构。

形态学图像处理的基本运算有 4 个：腐蚀、膨胀、开操作、闭操作。

汉字的细化 → 汉字的细化



提取骨架



# 基础知识

- 子集:

$$\forall a \in A, a \in B \Rightarrow A \subseteq B \quad (1)$$

- 并集:

$$C = \{c | c \in A \text{ or } c \in B\} = A \cup B \quad (2)$$

- 交集:

$$C = \{c | c \in A \text{ and } c \in B\} = A \cap B \quad (3)$$

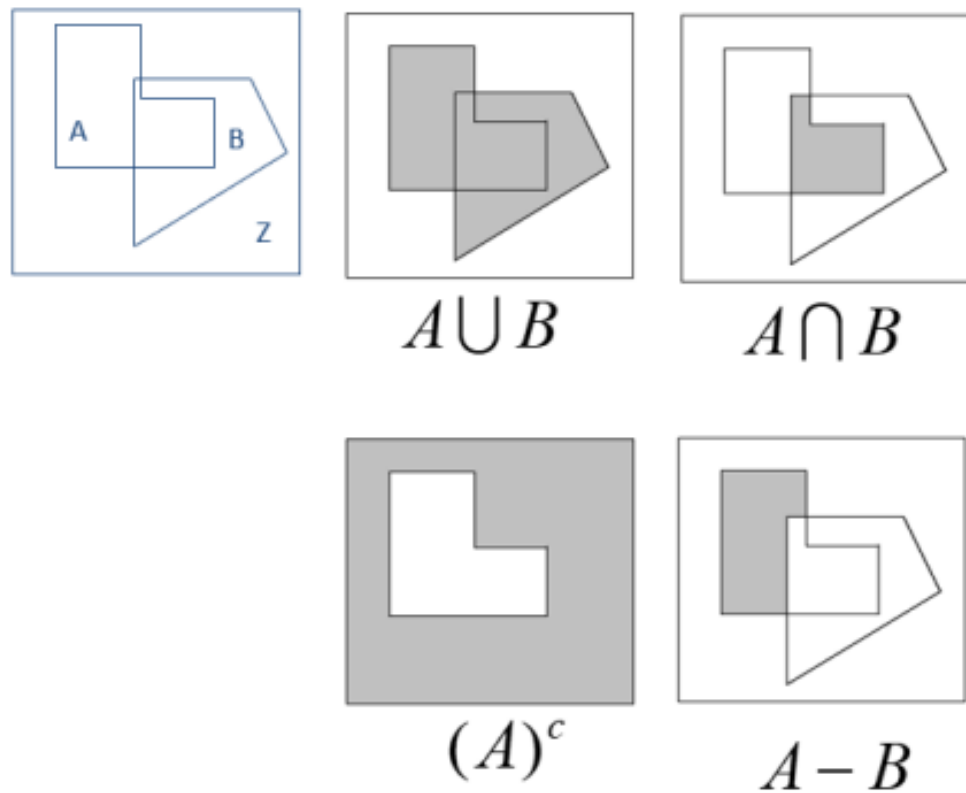
- 补集:

$$A^c = \{x | x \notin A\} \quad (4)$$

- 集合的差

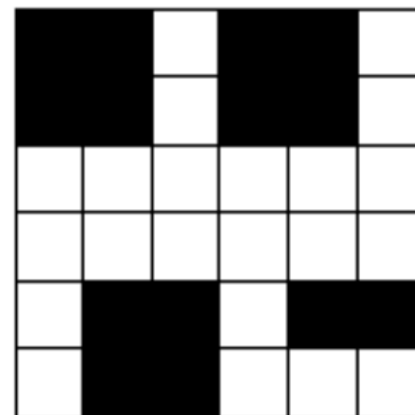
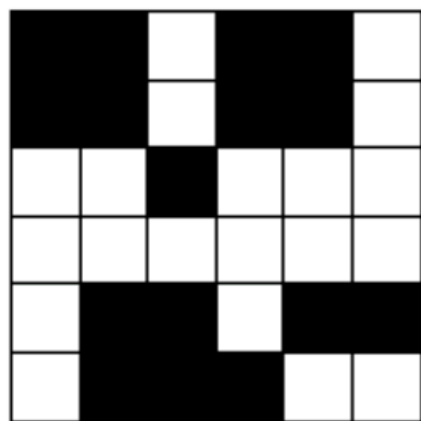
$$A - B = \{x | x \in A, x \notin B\} = A \cap B^c \quad (5)$$

# 基础知识



## 2、腐蚀

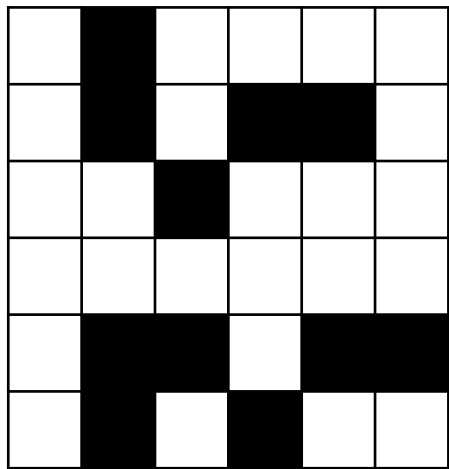
**腐蚀：消除连通域的边界点，使边界向内收缩的处理。**



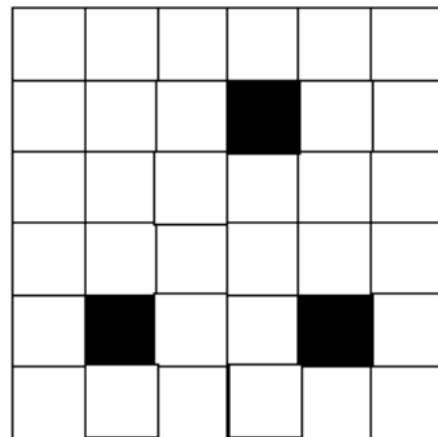


## 腐蚀的设计思想

设计一个**结构元素**，结构元素的原点定位在待处理的目标像素上，通过判断是否覆盖，来确定是否该点被腐蚀掉。



结构元素



$A$  和  $B$  是两个集合， $A$  被  $B$  腐蚀定义为：

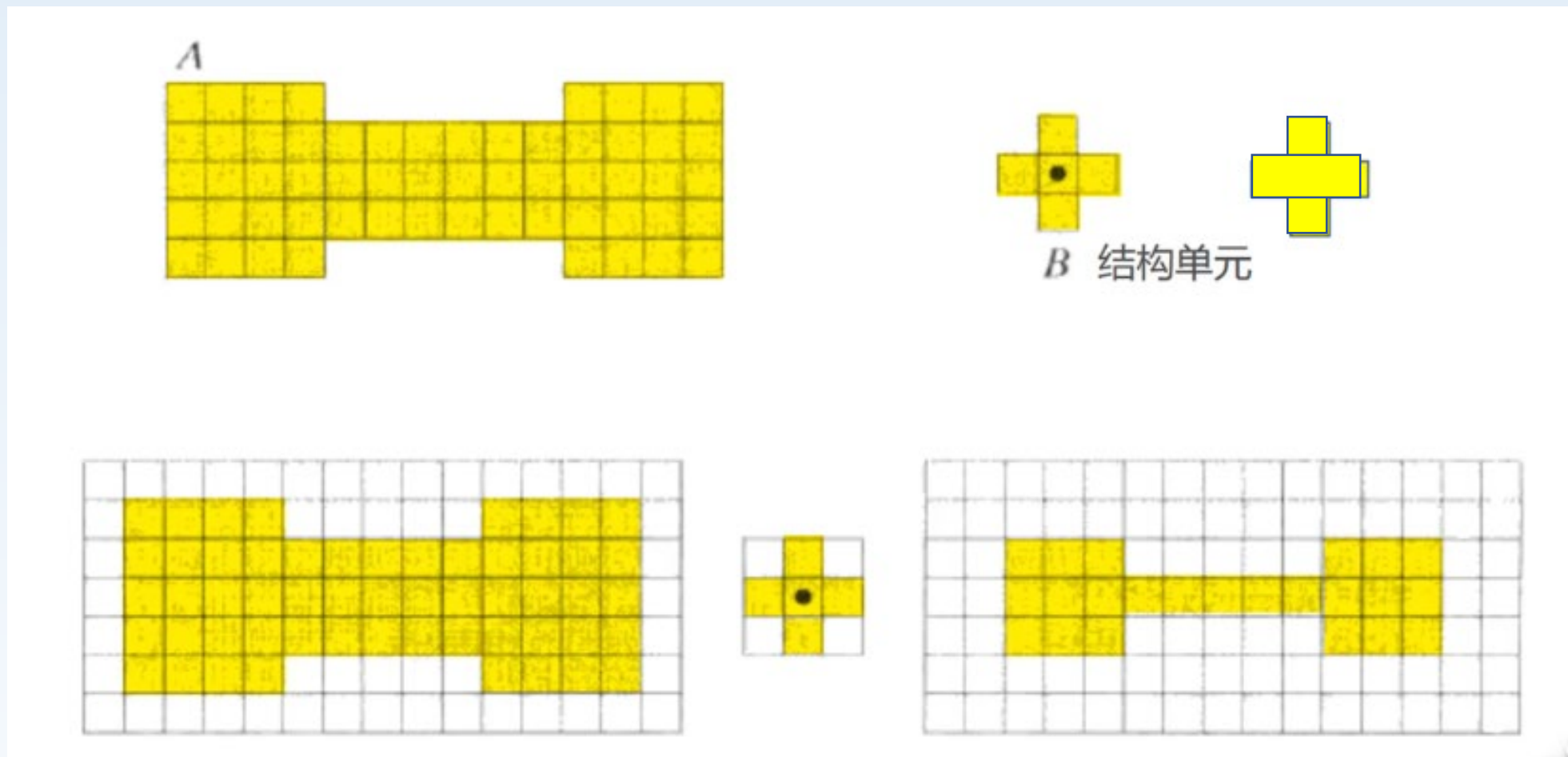
$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (8)$$

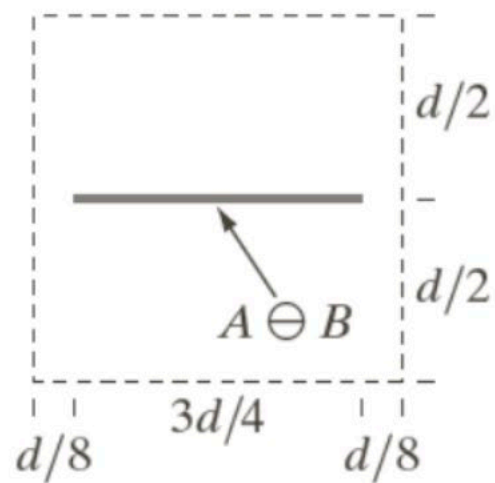
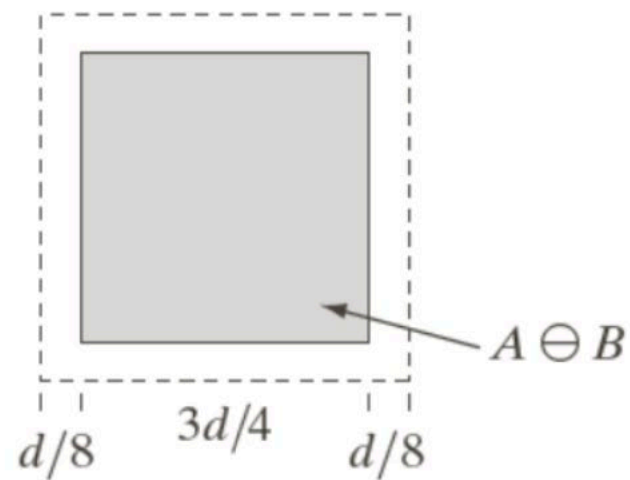
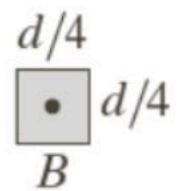
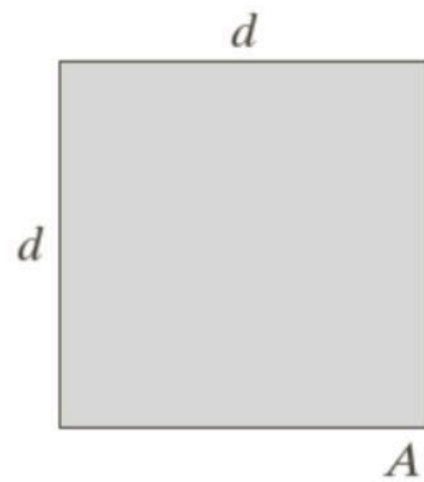
## 腐蚀：算法步骤

- ① 扫描原图，找到第一个像素值为 1 的目标点；
- ② 将预先设定好形状以及原点位置的结构元素的原点移到该点；
- ③ 判断该结构元素所覆盖的像素值是否全部为 1；
- ④ 如果是，则腐蚀后图像中的相同位置上的像素值为 1；
- ⑤ 如果不是，则腐蚀后图像中的相同位置上的像素值为 0；
- ⑥ 重复步骤 2 和步骤 3，直到所有原图中像素处理完成。

注意：第4步中，只有结构元中心点对应的原图位置1

## 2、腐蚀





## 2、腐蚀

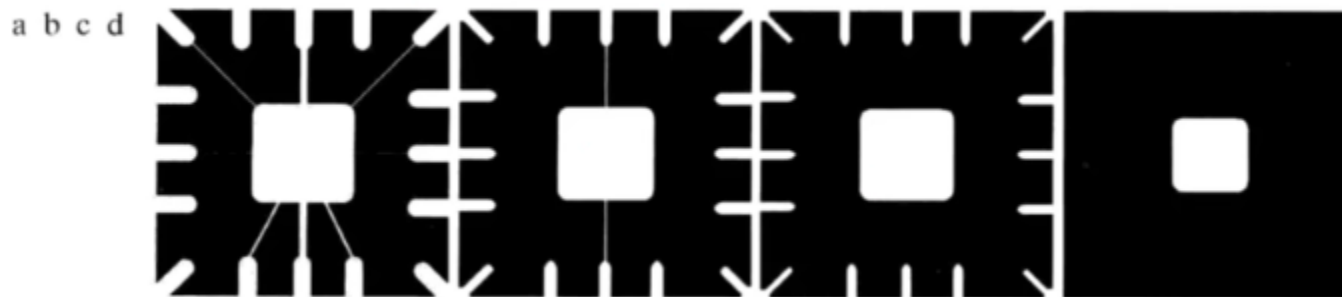


图 9.5 使用腐蚀去除图像中的部件：(a)一幅大小为  $486 \times 486$  的连线模板二值图像；(b)~(d)分别使用大小为  $11 \times 11$ ,  $15 \times 15$  和  $45 \times 45$  的结构元腐蚀的图像。SE 的元素都是 1

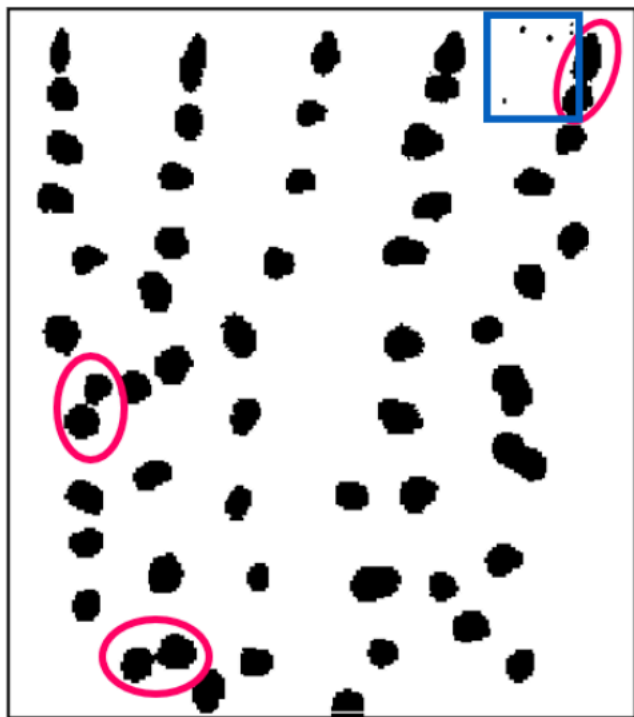
- 腐蚀缩小或者细化了二值图像中的物体。
- 腐蚀可以看成是形态学滤波操作，这种操作将小于结构元的图像细节从图像中滤除。



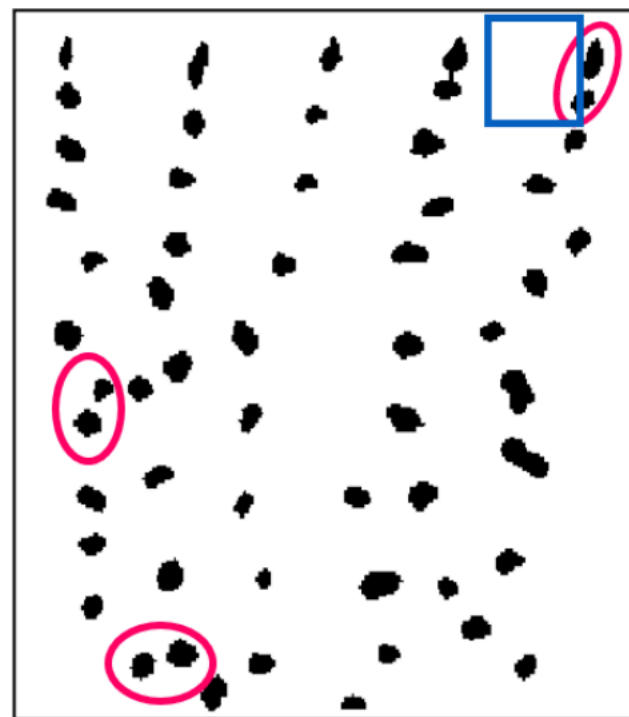
结构元

## 腐蚀的应用

- 腐蚀处理可以将粘连在一起的不同目标物**分离**，并可以将小的颗粒**去除**。



(a) 原图



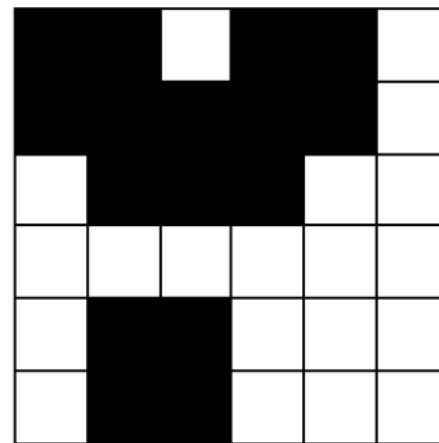
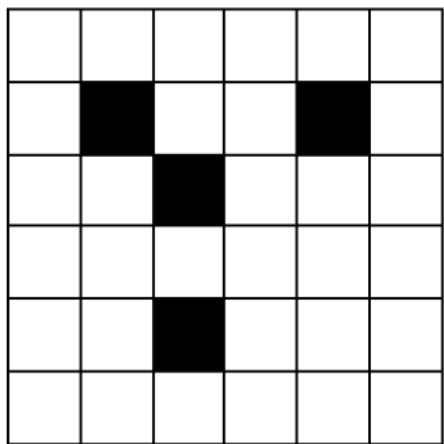
(b) 腐蚀两次



形态学滤波的应用

### 3、膨胀

- 膨胀是将与目标区域的背景点合并到该目标物中，使目标物边界向外部扩张的处理。





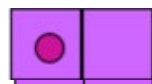
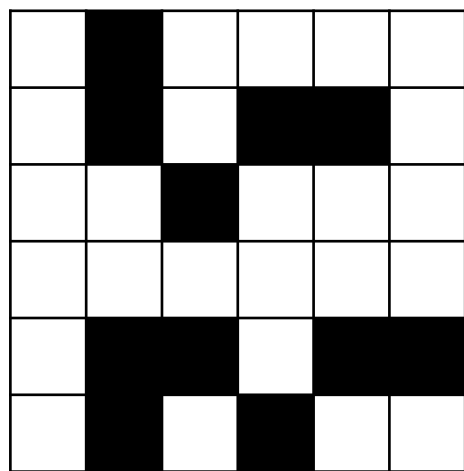
## 膨胀：算法步骤

- ① 扫描原图，找到第一个像素值为 0 的背景点；
- ② 将预先设定好形状以及原点位置的结构元素的原点移到该点；
- ③ 判断该结构元素所覆盖的像素值是否存在为 1 的目标点；
- ④ 如果是，则膨胀后图像中的相同位置上的像素值为 1；
- ⑤ 如果不是，则膨胀后图像中的相同位置上的像素值为 0；
- ⑥ 重复步骤 2 和步骤 3，直到所有原图中像素处理完成。

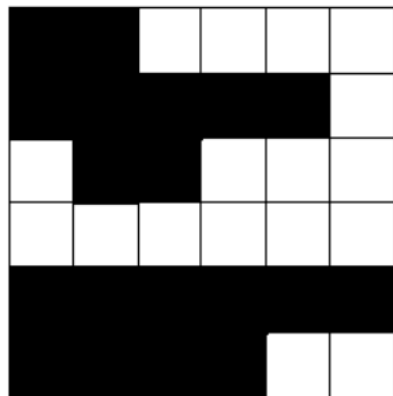
第4步中，只有结构元中心点对应的原图位置1

## 膨胀: 设计思想

- 设计一个结构元素，结构元素的原点定位在背景像素上，判断是否覆盖有目标点，来确定是否该点被膨胀为目标点。



结构元素

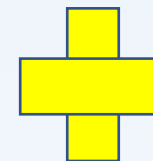
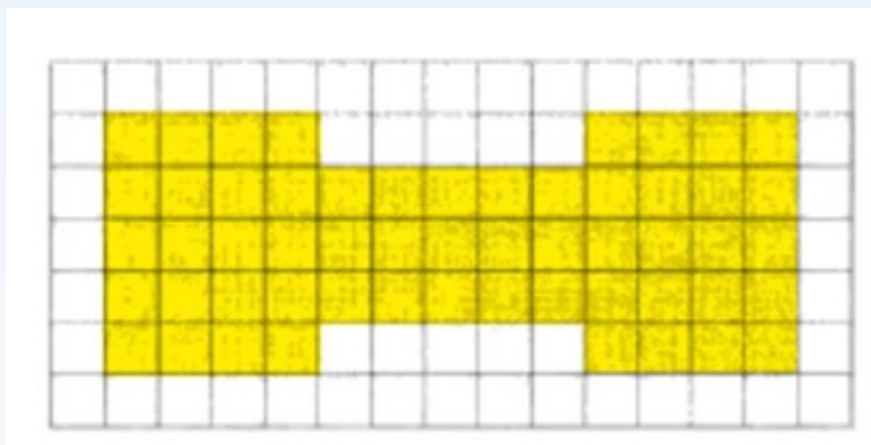
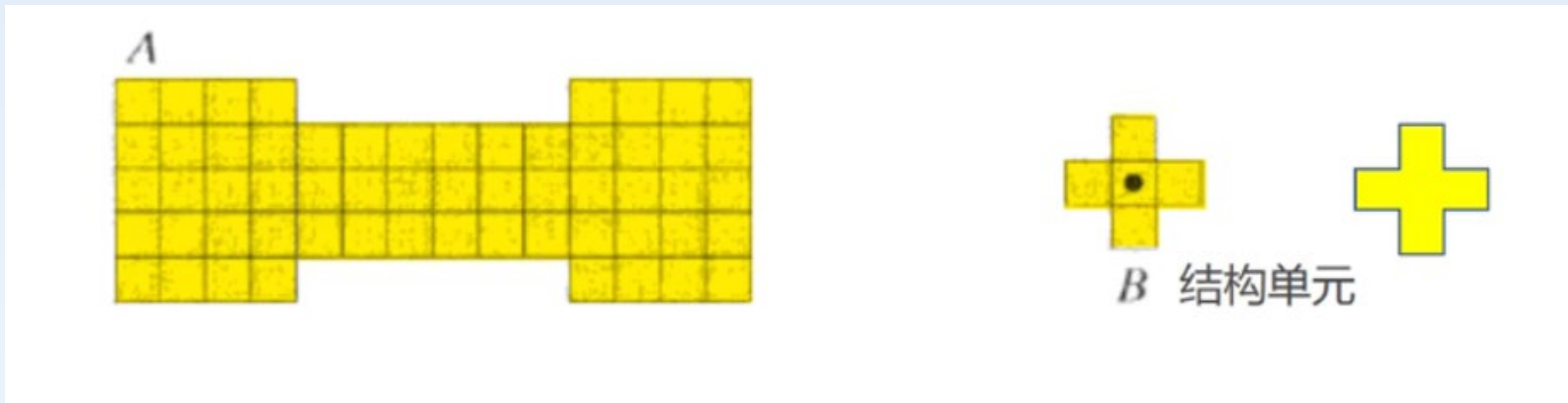


- $A$  和  $B$  是两个集合， $A$  被  $B$  膨胀定义为：

$$A \oplus B = \{z | (\hat{B})_z \cap A \subseteq A\} \quad (9)$$

- 上式表示， $B$  的反射进行平移与  $A$  的交集是  $A$  的子集。

### 3、膨胀



## 膨胀的应用

- 膨胀处理可以将断裂开的目标物进行合并，便于对其整体的提取。



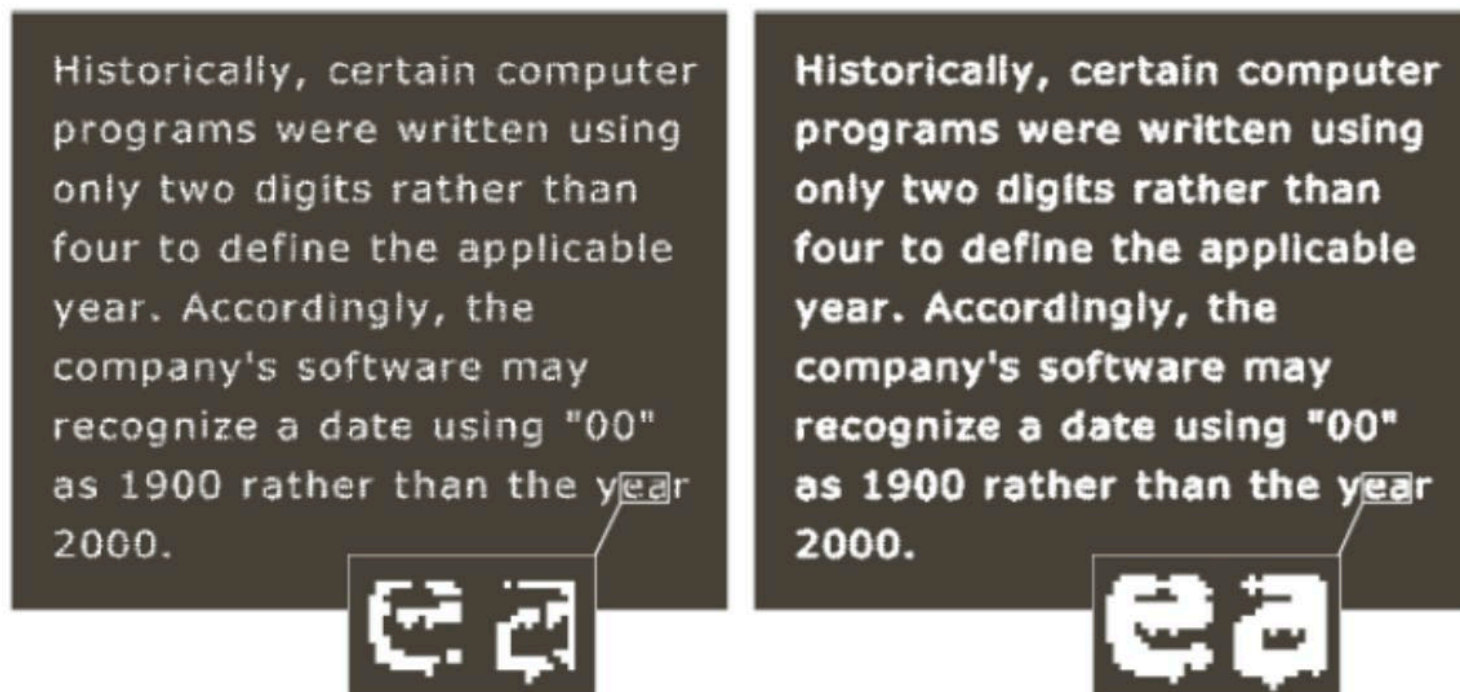
(a) 原图



(b) 膨胀一次

# 膨胀的应用

桥接文字裂缝优点：在一幅二值图像中直接得到结果，与低通滤波方法相比，**边缘仍然清晰**。



0	1	0
1	1	1
0	1	0

## 4、开运算与闭运算

### 提出背景

- 腐蚀和膨胀运算的缺点，改变了原目标物的大小。
- 为了解决这一问题，考虑到腐蚀与膨胀是一对逆运算，将膨胀与腐蚀运算同时进行。由此便构成了开运算与闭运算。

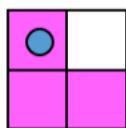
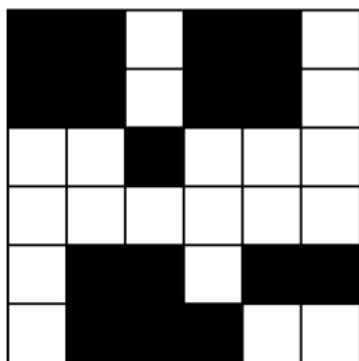
## 开运算：算法原理

- 开运算可以在分离粘连目标物的同时，基本保持原目标物的大小.
- 定义：

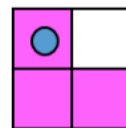
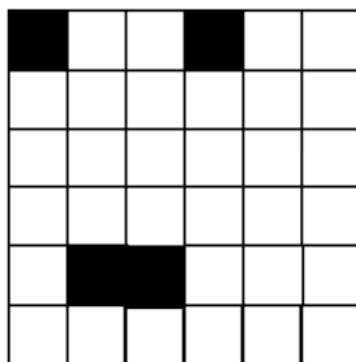
$$A \circ B = (A \ominus B) \oplus B \quad (10)$$

- 含义：先用 B 对 A 腐蚀，然后用 B 对结果膨胀.

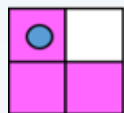
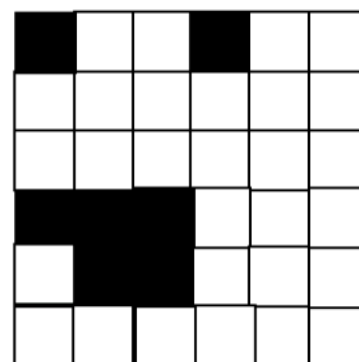
# 开运算示例



腐蚀

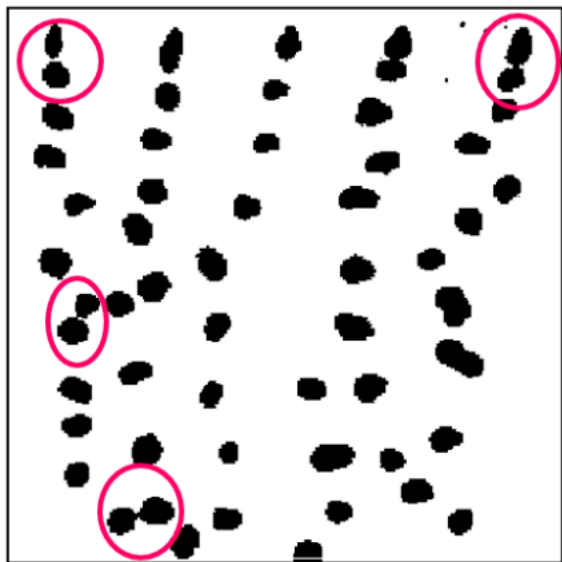


膨胀

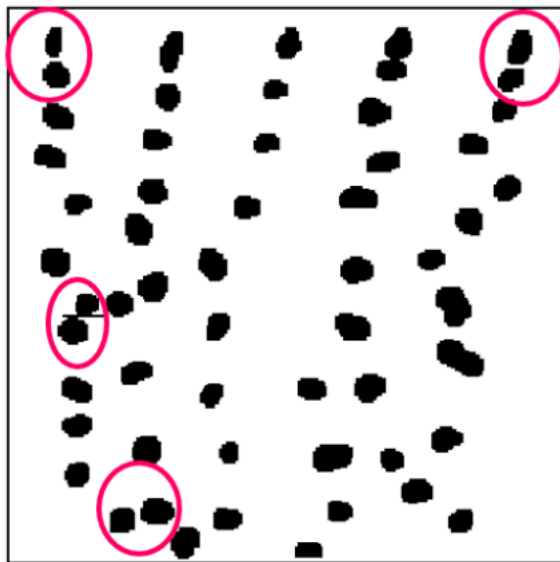




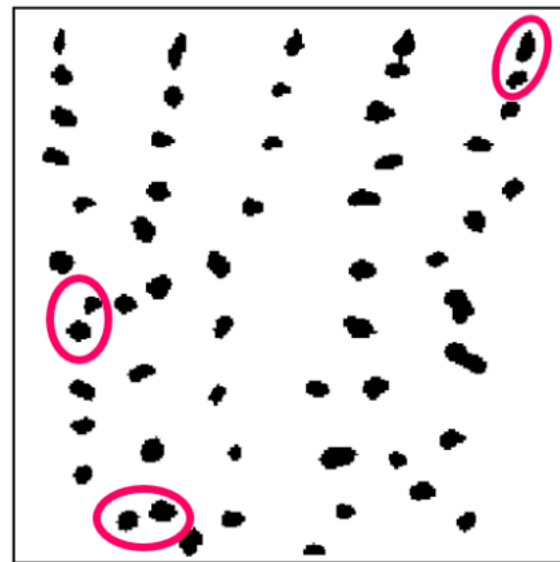
## 开运算应用示例



(a) 原图



(b) 开运算结果



(c) 腐蚀运算结果

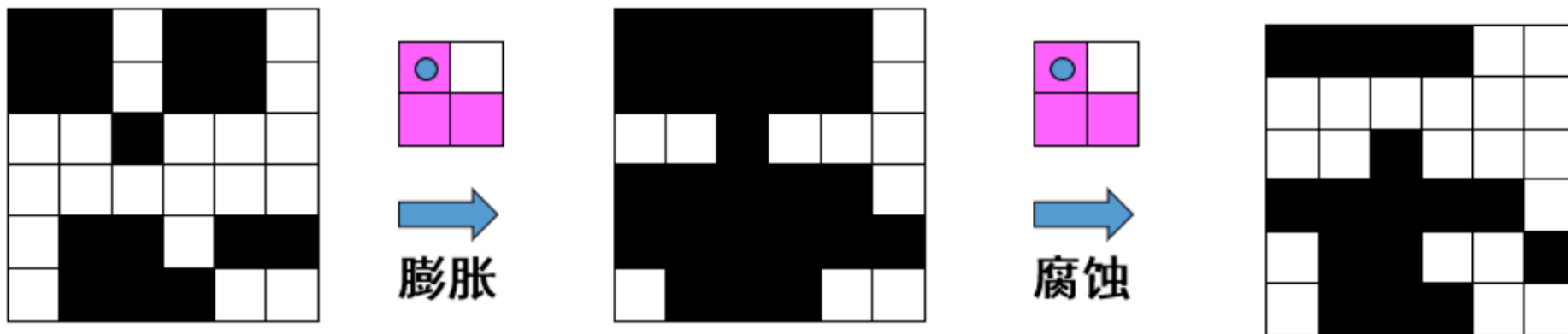
## 闭运算：算法原理

- 闭运算可以在合并断裂目标物的同时，基本保持原目标物的大小.
- 定义：

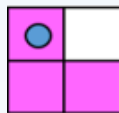
$$A \cdot B = (A \oplus B) \ominus B \quad (11)$$

- 含义：先用 B 对 A 膨胀，然后用 B 对结果腐蚀。

## 闭运算示例

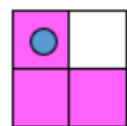
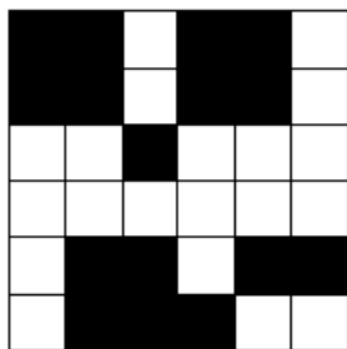


- 本例未能将分裂成两个连通域的目标合并，怎么办？

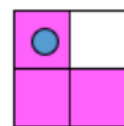
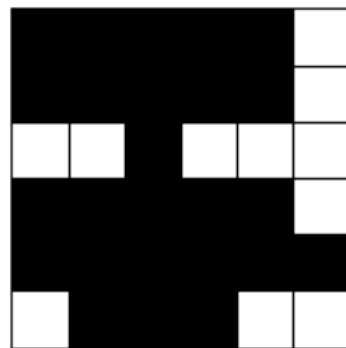


- 如果当按照常规的开运算不能分离粘连，或者是闭运算不能合并断裂：
- 对于开运算可以先进行  $N$  次腐蚀，再进行  $N$  次膨胀；
- 对于闭运算可以先进行  $N$  次膨胀，再进行  $N$  次腐蚀。

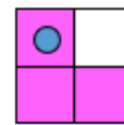
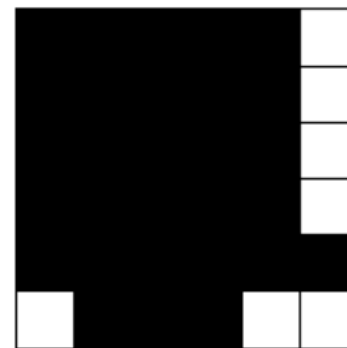
# 变形闭运算的示例



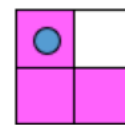
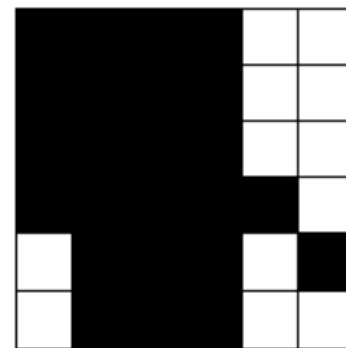
一次膨胀



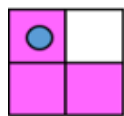
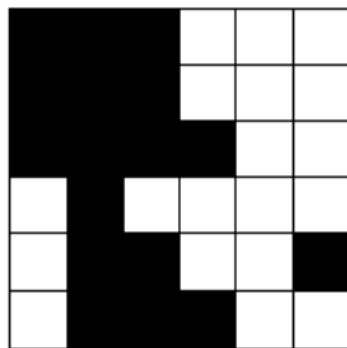
二次膨胀



一次腐蚀



二次腐蚀



## 闭运算应用示例



(a) 原图

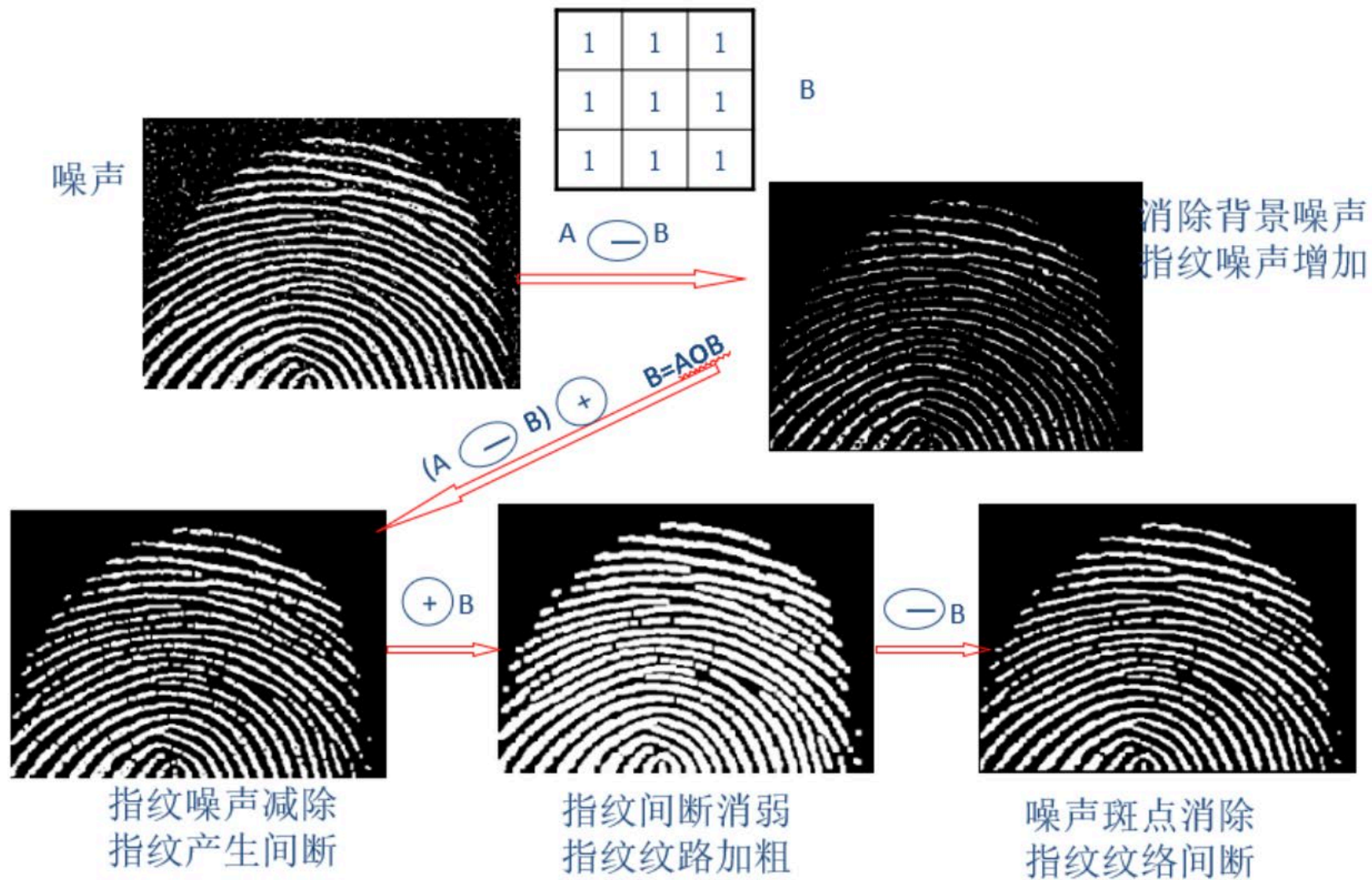


(b) 闭运算结果



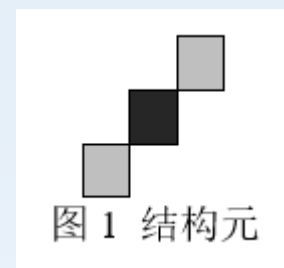
(c) 膨胀运算结果

# 开和闭运算应用示例



练习：对下图用图1的结构元进行腐蚀与膨胀。

		↖	↖	↖	↖	↖	↖	↖	↖
		↖	↖	↖	↖	↖	↖	↖	↖
	↖	↖	↖	↖	↖	↖	↖	↖	↖
	↖	↖	↖	↖	↖	↖	↖	↖	↖
	↖	↖	↖		↖	↖	↖	↖	↖
↖	↖	↖	↖	↖	↖	↖	↖	↖	↖



		√	√	√					
		√	√	√				√	
			√						

腐蚀结果二值图像

			√	√	√	√			
		√	√	√	√	√	√	√	√
	√	√	√	√	√	√	√	√	√
√	√	√	√	√	√	√	√	√	√
√	√	√	√		√	√	√	√	
√	√	√	√			√	√		

膨胀结果二值图像



本 章 结 束！