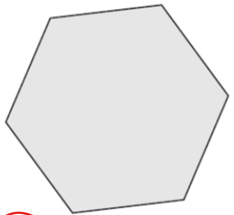
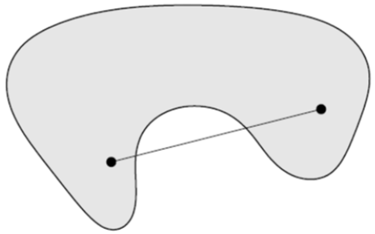
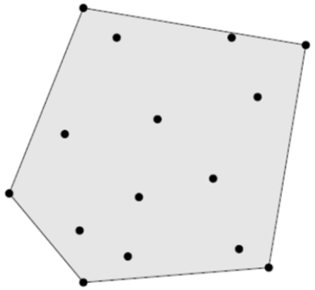
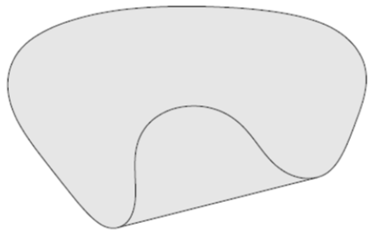
凸包

- 凸集合（convex set）
  - 集合内任意两点的连线属于该集合

- 集合S的凸包（convex hull）H
  - 包含S的最小凸集合

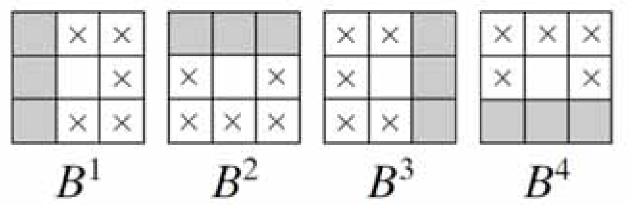



- 凸缺（convex deficiency）：$H - S$

构建凸包算法

相差90°

- 四个结构元
  - 黑色表示1前景
  - 白色表示0背景，×表示任意值



⊛不用考虑外围背景

1. 按照下面的公式更新

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

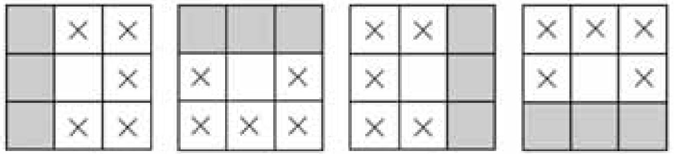
   - 其中$X_0^i = A$

2. 重复上述公式，直到$X_k^i = X_{k-1}^i$
3. 集合A的凸包

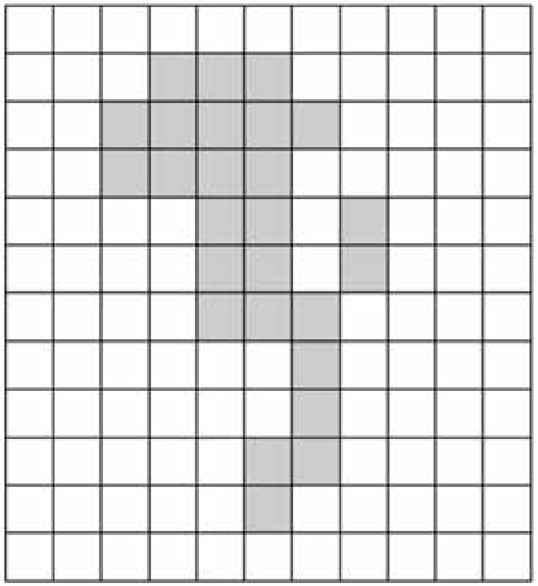
$$C(A) = \bigcup_{i=1}^{4} D^i$$

   - 其中$D^i = X_k^i$

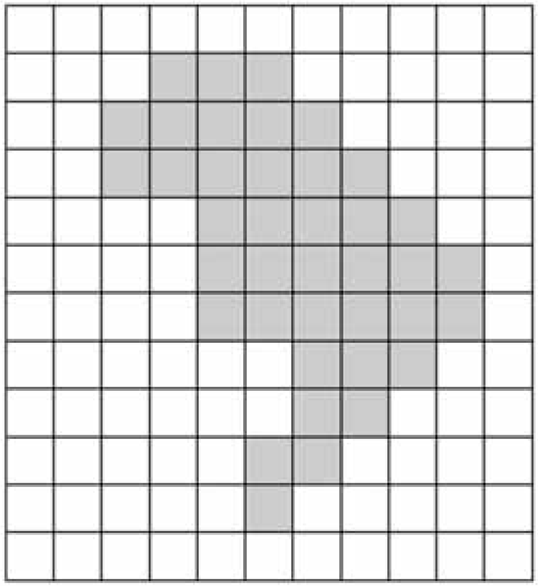
举例



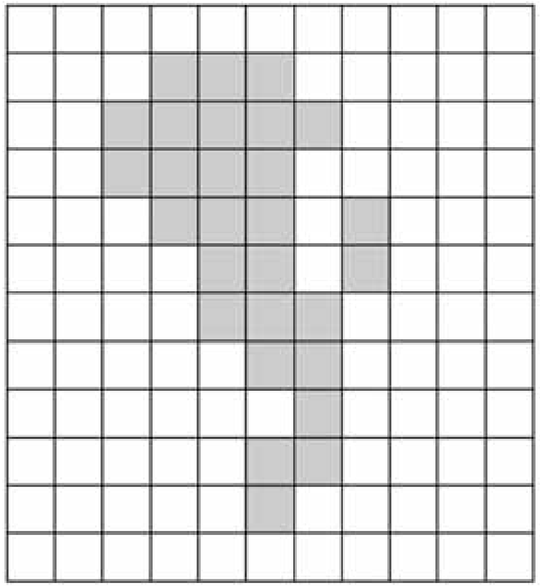
B^1 B^2 B^3 B^4



$X_0^1 = A$

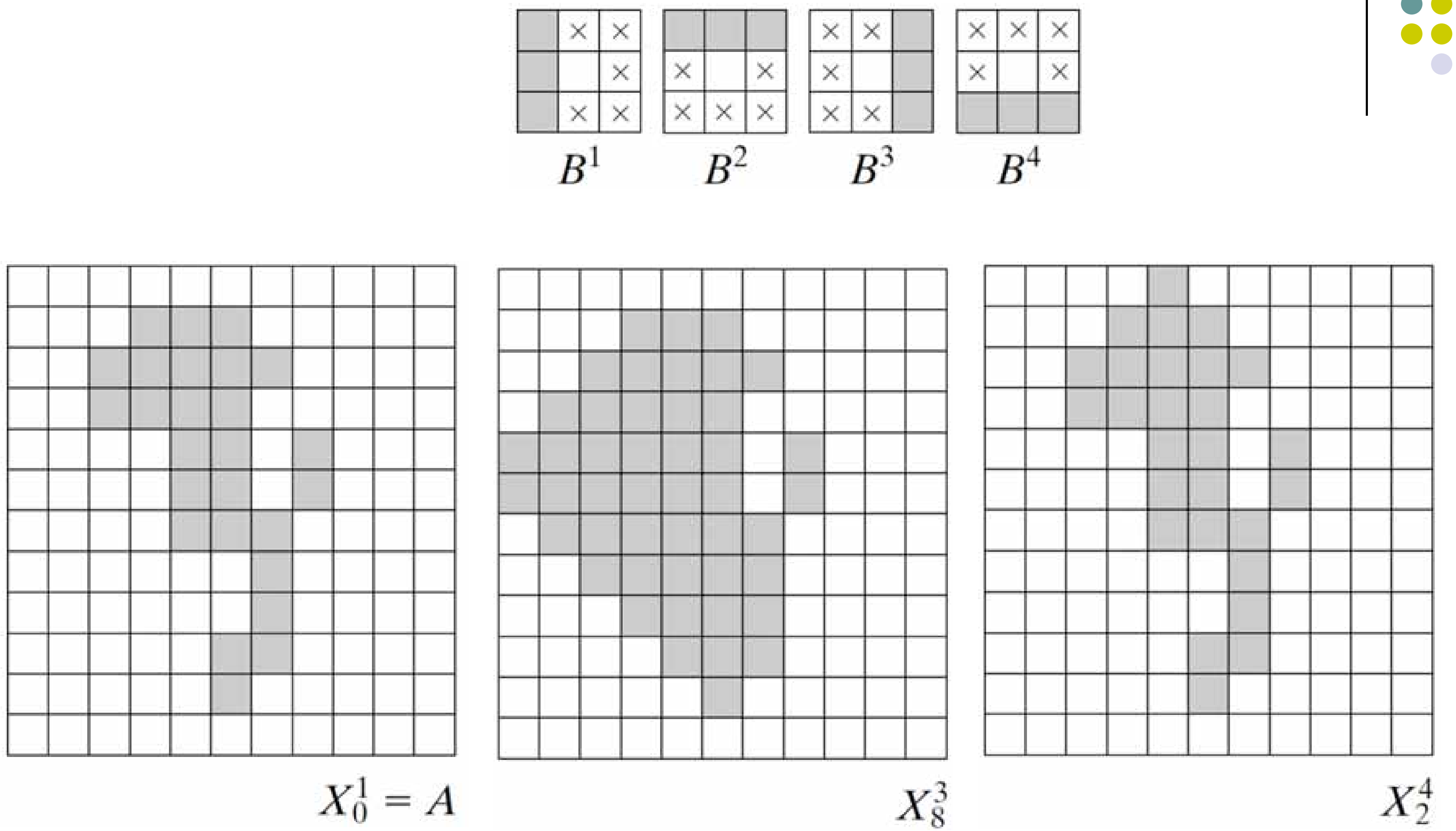


X_4^1

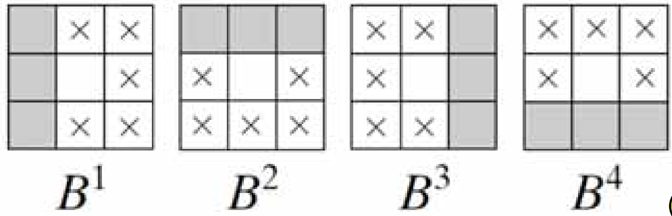


X_2^2

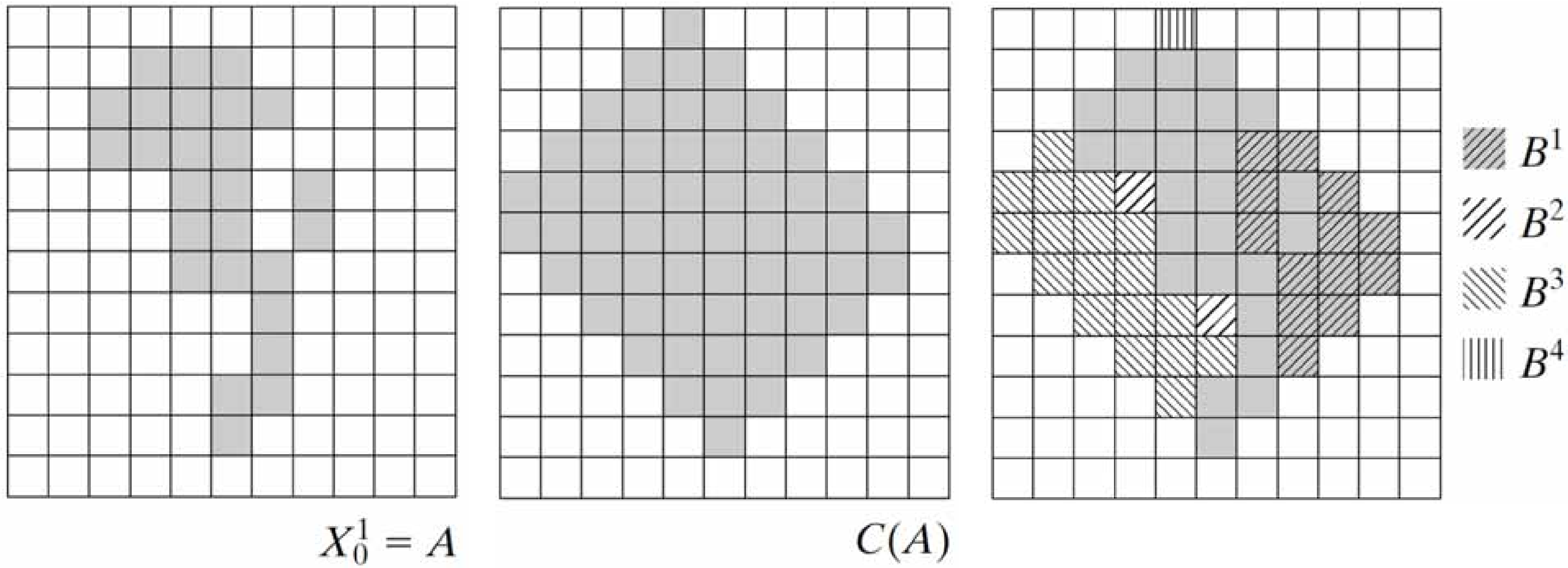
举例



举例

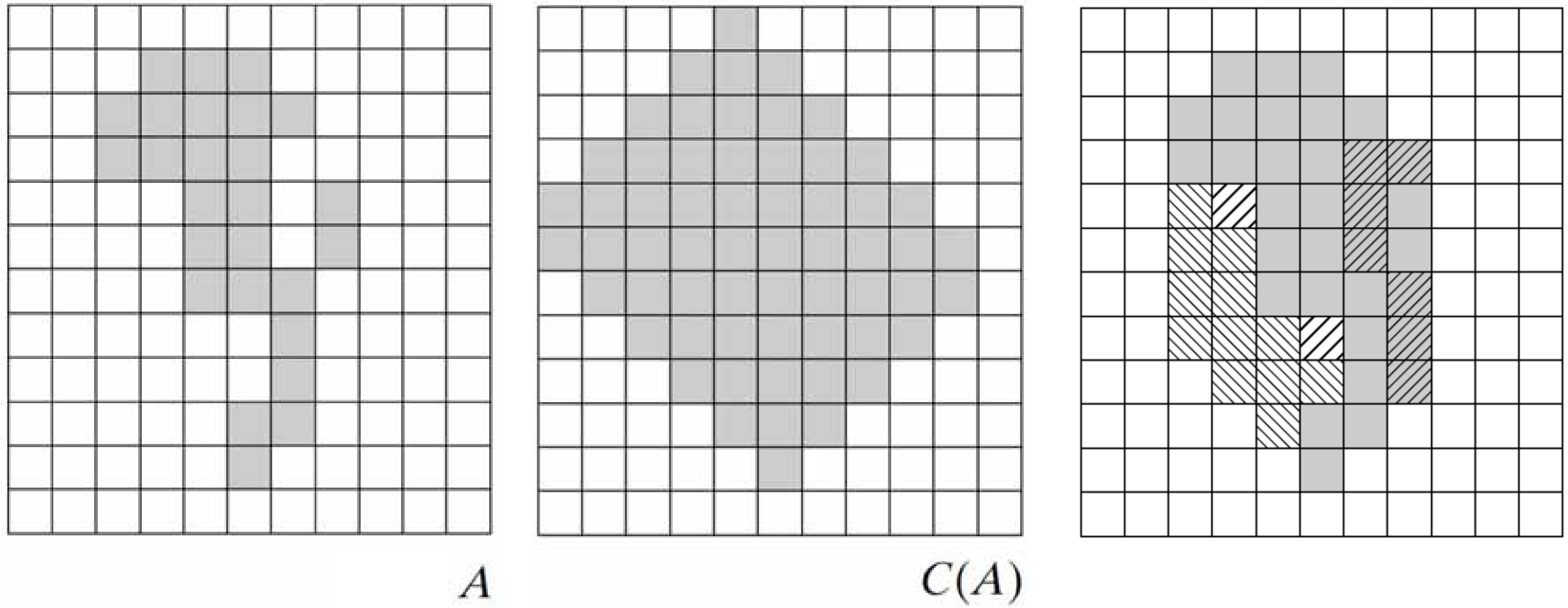


不是最小的凸集合



举例

- 不能超过原图像的垂直和水平范围



- 还可以添加更复杂的约束

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

细化

- 结构元B对集合A的细化（thinning）

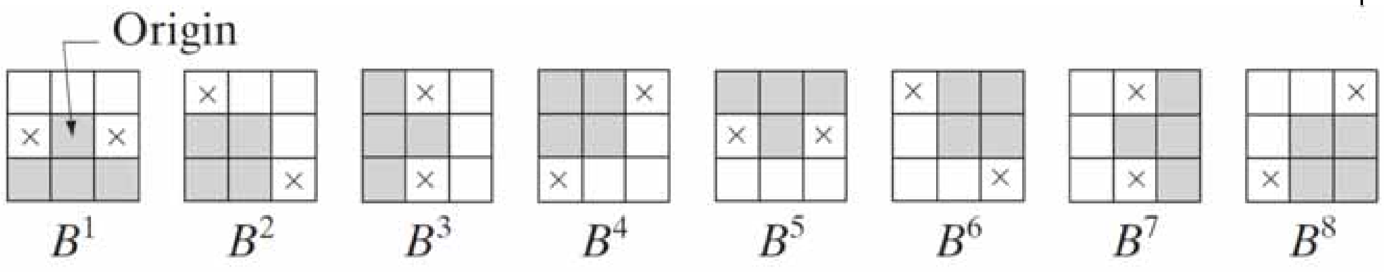
$$A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$$

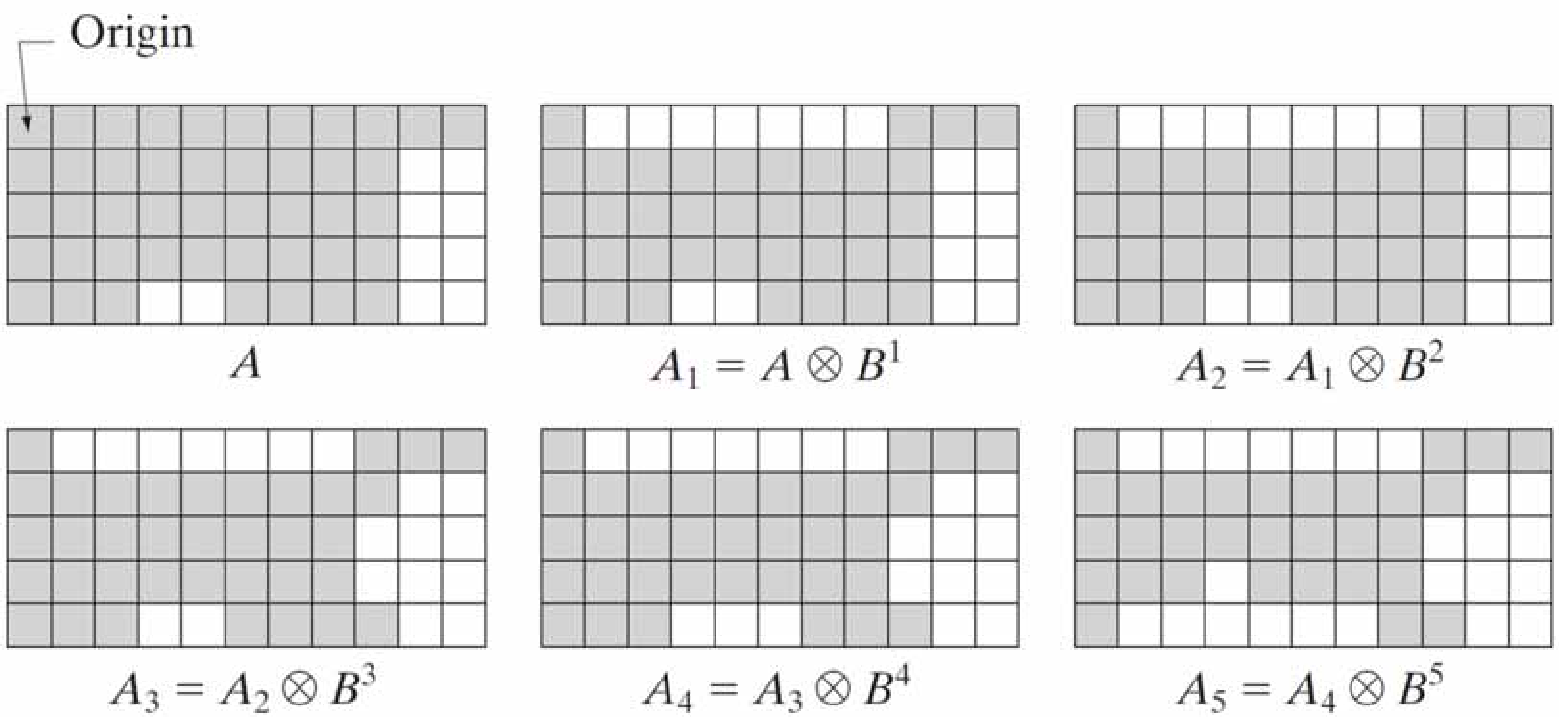
  - $\circledast$不用考虑外围背景
  - 与边界提取很类似
- 结构元序列$\{B\} = \{B^1, B^2, \ldots, B^n\}$对集合$A$的细化

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

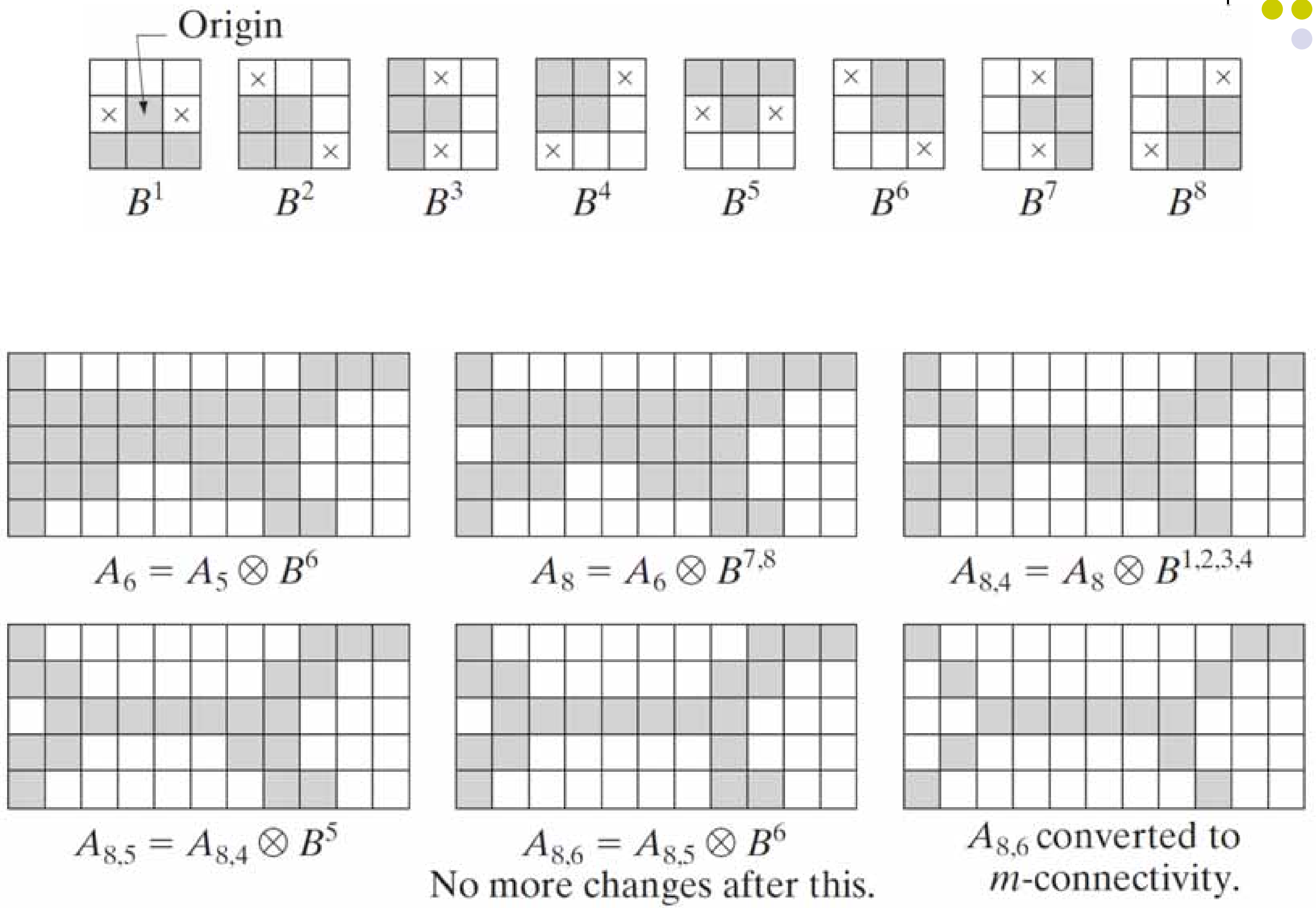
  - B^i是B^{i-1}的旋转版本
  - 重复上述过程，直至结果不发生变化

举例


Origin
B^1
B^2
B^3
B^4
B^5
B^6
B^7
B^8


Origin
A
$A_1 = A \otimes B^1$
$A_2 = A_1 \otimes B^2$
$A_3 = A_2 \otimes B^3$
$A_4 = A_3 \otimes B^4$
$A_5 = A_4 \otimes B^5$

举例



Origin
B^1 B^2 B^3 B^4 B^5 B^6 B^7 B^8
$A_6 = A_5 \otimes B^6$
$A_8 = A_6 \otimes B^{7,8}$
$A_{8,4} = A_8 \otimes B^{1,2,3,4}$
$A_{8,5} = A_{8,4} \otimes B^5$
$A_{8,6} = A_{8,5} \otimes B^6$
No more changes after this.
$A_{8,6}$ converted to
m-connectivity.

粗化

- 结构元B对集合A的粗化（thickening）

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

  - $\circledast$不用考虑外围背景
  - 结构元和细化的相反
- 结构元序列$\{B\} = \{B^1, B^2, \ldots, B^n\}$对集合$A$的粗化

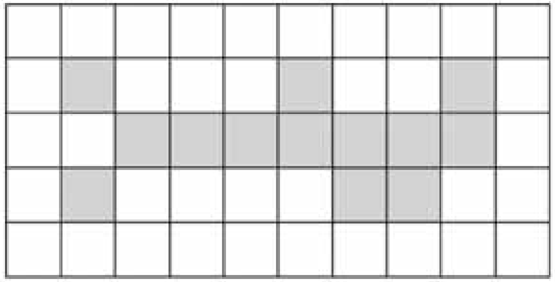
$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2) \ldots) \odot B^n)$$

  - B^i是B^{i-1}的旋转版本
  - 重复上述过程，直至结果不发生变化

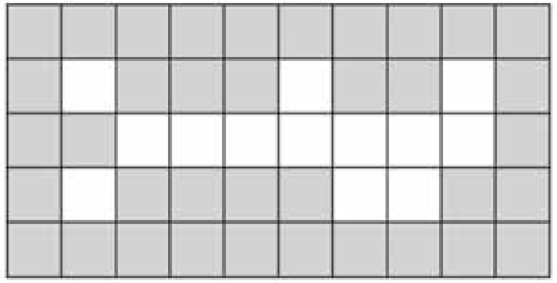
粗化算法

1. 计算集合A的补集C
2. 细化C
3. 计算上述结果的补集

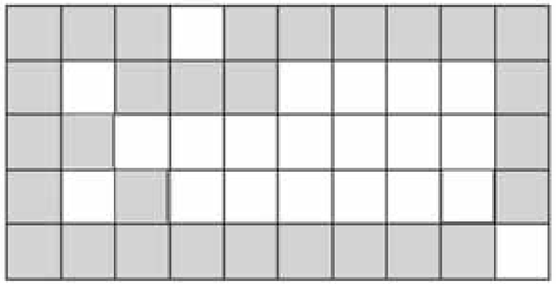
A



$C = A^c$

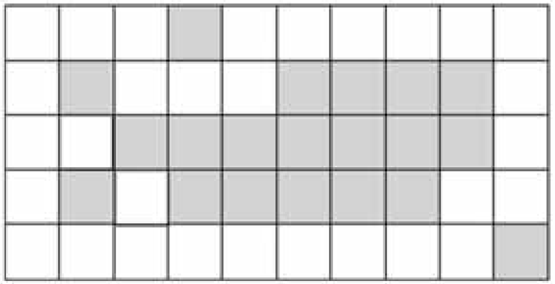


细化



形成了一个边界

求补

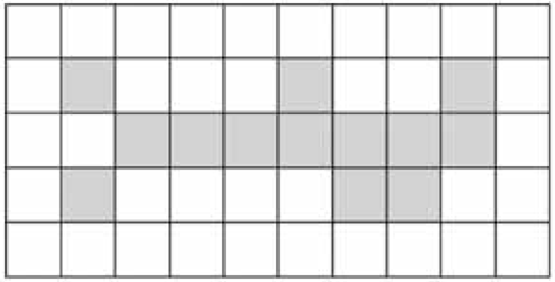


存在孤立点

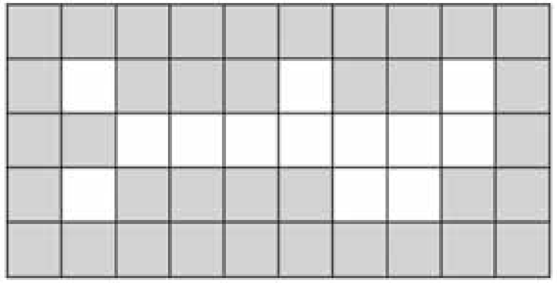
粗化算法

1. 计算集合A的补集C
2. 细化C
3. 计算上述结果的补集

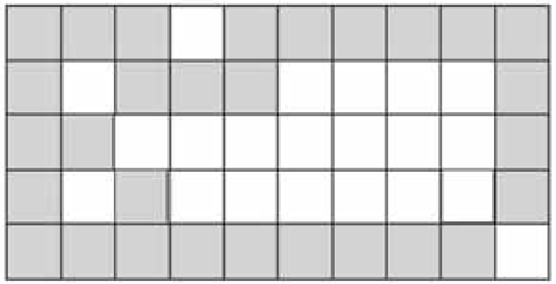
A



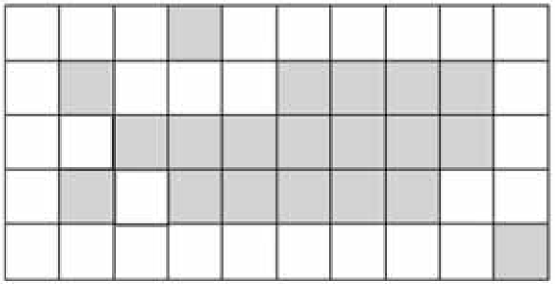
$C = A^c$



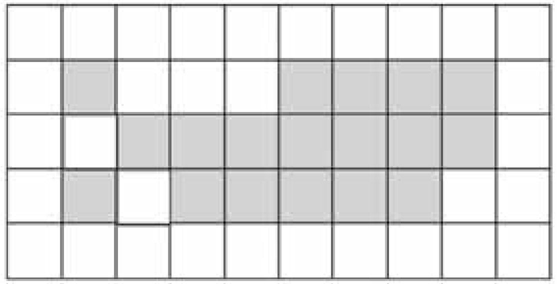
细化



求补



去掉孤立点

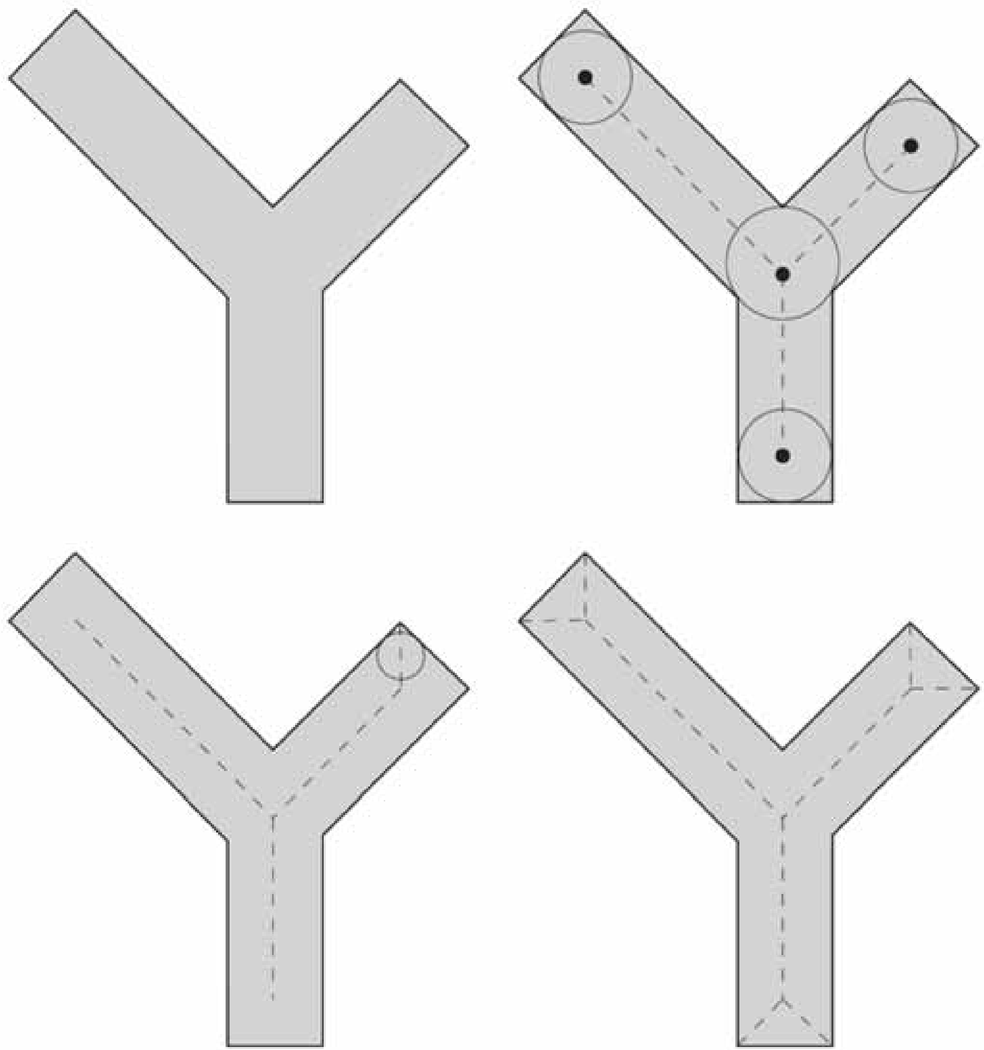


提纲

- 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

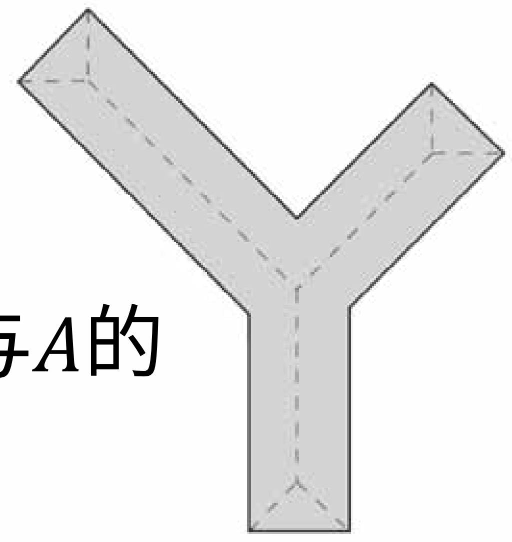
骨架

- 举例



骨架

- 集合A的骨架（skeleton）记为$S(A)$
  1. 如果$z \in S(A)$，并且$(D)_z$是A内以z为中心的最大圆盘，则不存在包含$(D)_z$且位于A内的更大圆盘。
     - $(D)_z$被称为最大圆盘
  2. $(D)_z$在两个或多个不同的位置与A的边界接触。



骨架

- 数学公式

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

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

  - $S_k(A)$是骨架子集
  - B是结构元
  - $A \ominus kB$表示对A进行k次连续腐蚀

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

  - K是A被腐蚀成空集的最后一次迭代

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

骨架

- 数学公式

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

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

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

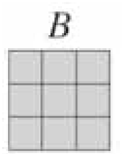
$$K = \max\{k \mid (A \ominus kB) \neq \varnothing\}$$

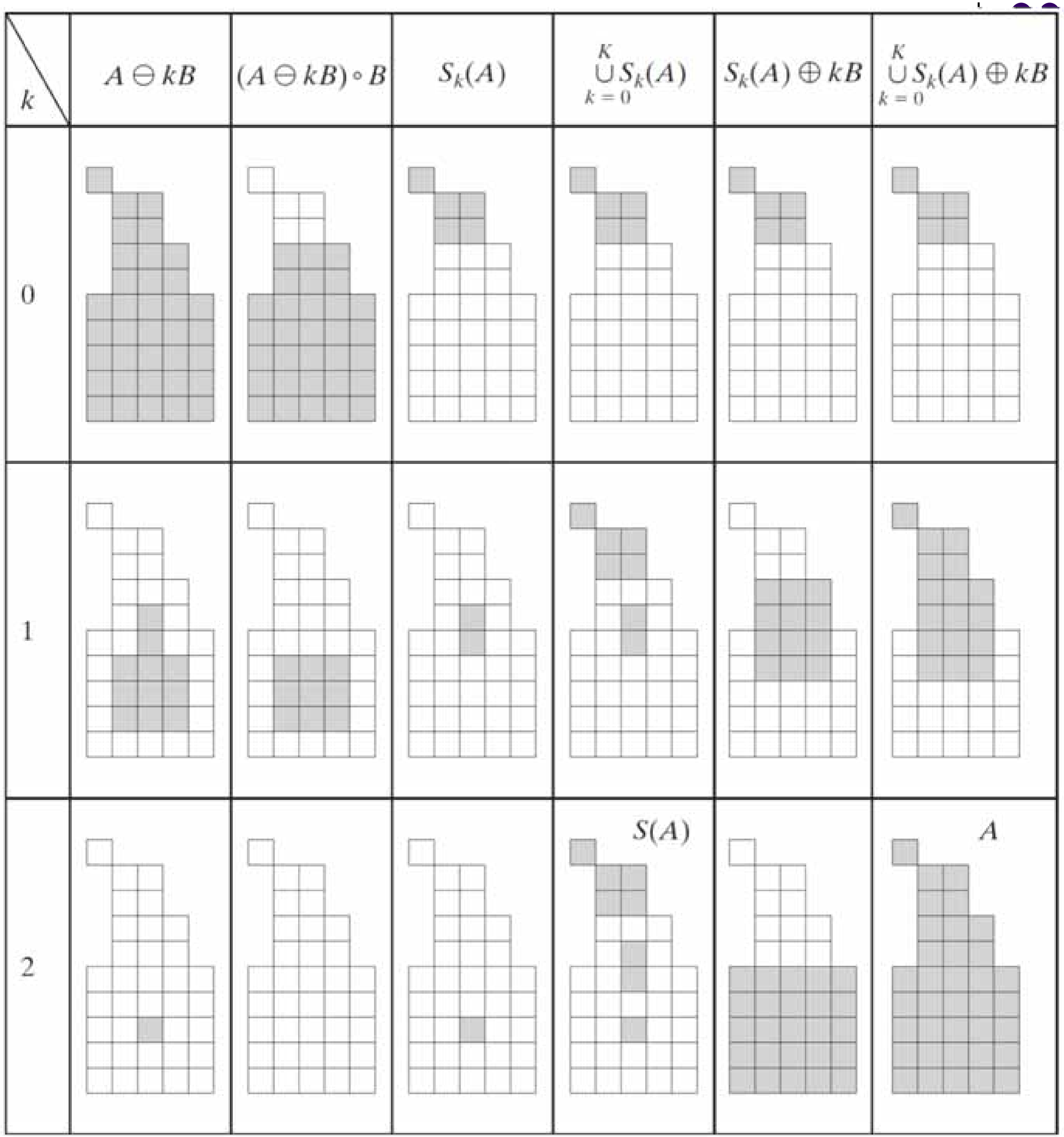
- 重构集合A

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

$$(S_k(A) \oplus kB) = ((\ldots((S_k(A) \oplus B) \oplus B) \oplus \ldots) \oplus B)$$

举例


B


k
$A \ominus kB$
$(A \ominus kB) \circ B$
$S_k(A)$
$\bigcup_{k=0}^{K} S_k(A)$
$S_k(A) \oplus kB$
$\bigcup_{k=0}^{K} S_k(A) \oplus kB$
0
1
2
$S(A)$
A

骨架粗，
且不连续

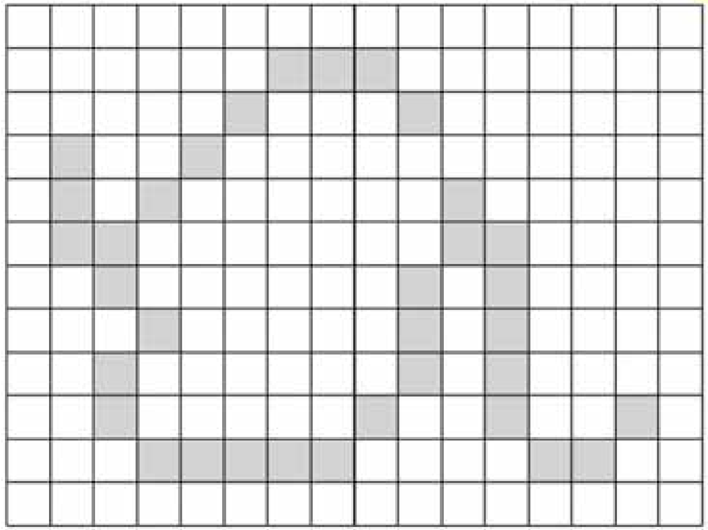
裁剪

- 裁剪（pruning）的作用
  - 对细化、骨架的补充
  - 上述操作易产生寄生分量，需要后处理去除

- 自动手写体识别
  - 通过需要分析字母的骨架形状
  - 但骨架往往带有许多“毛刺”（寄生分量）
  - “毛刺”是由笔画的不均匀造成
  - 假设寄生分量的长度较短

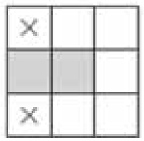
裁剪

- 字符a的骨架
  - 最左边存在毛刺
  - 通过删除端点去除
  - 删除长度≤3的分支

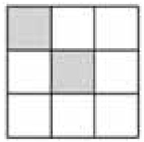


1. 使用检测端点的结构元对集合A细化

$$X_1 = A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2) \ldots) \otimes B^n)$$



B^1, B^2, B^3, B^4 (rotated 90°)



B^5, B^6, B^7, B^8 (rotated 90°)

裁剪

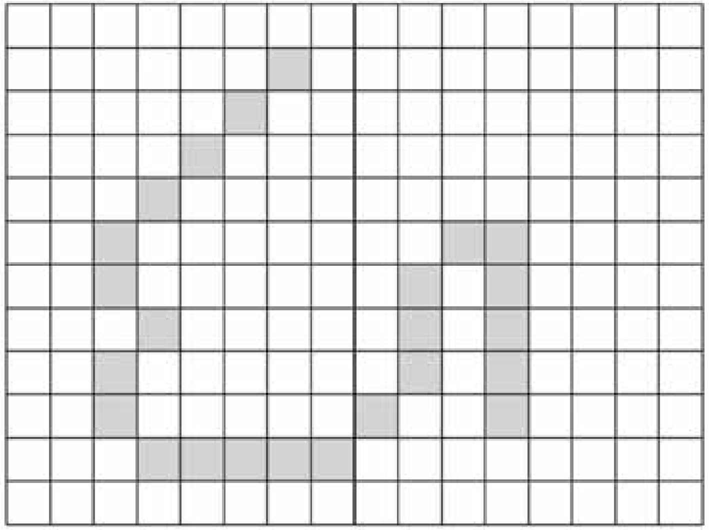
1. 使用检测端点的结构元对集合A细化

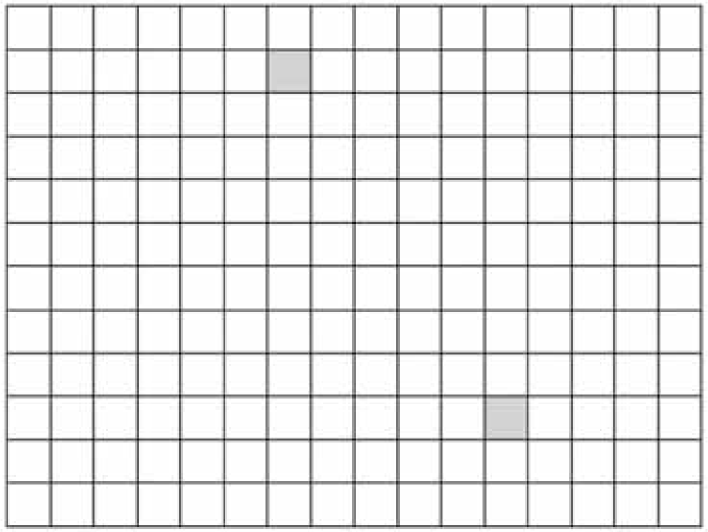
$$X_1 = A \otimes \{B\}$$

- 细化3次
- 复原形状

2. 计算X_1的端点

$$X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k)$$



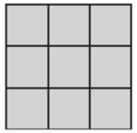


裁剪

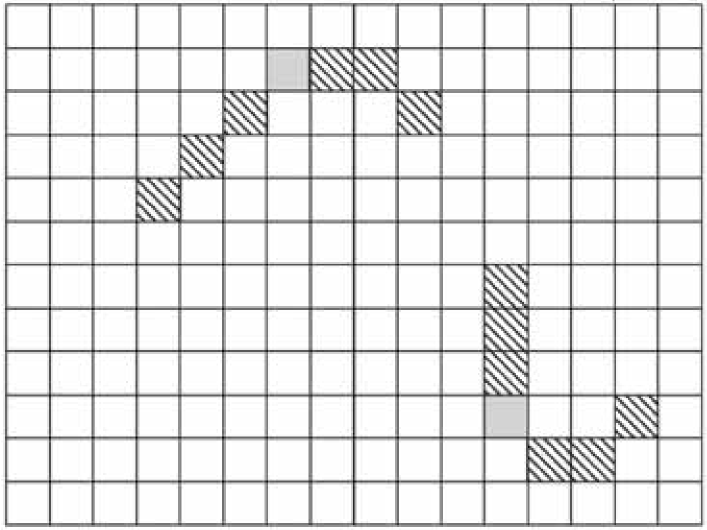
3. 对端点进行膨胀

$$X_3 = (X_2 \oplus H) \cap A$$

- 条件膨胀3次



H



4. 合并结果

$$X_4 = X_1 \cup X_3$$