



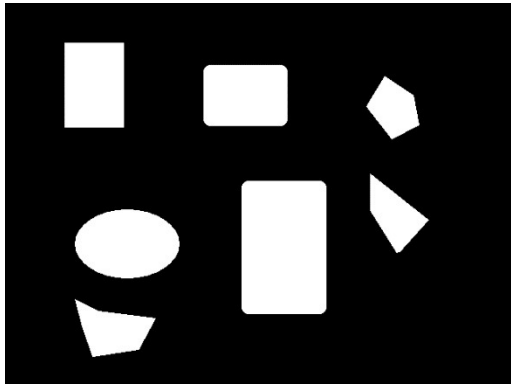
计算机视觉 Computer Vision

-- Structures 1

钟 凡

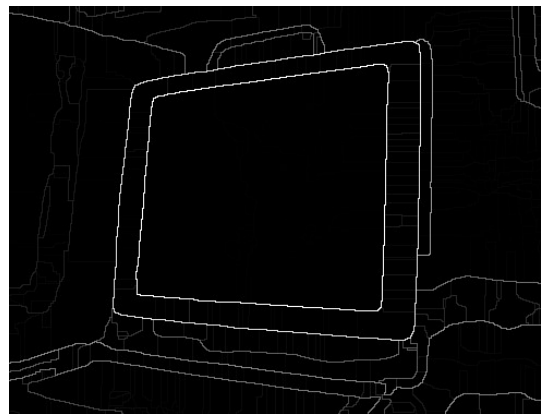
zhongfan@sdu.edu.cn

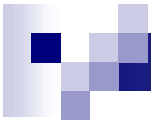
图像结构



图像结构

- 如何提取和利用图像的结构信息？

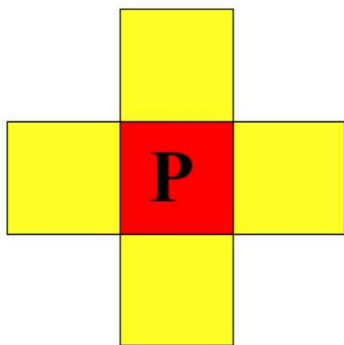




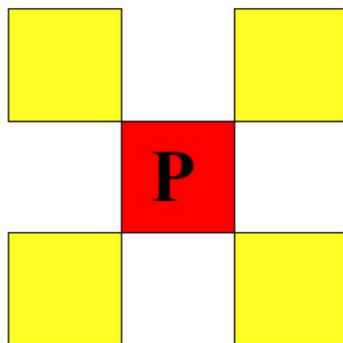
像素连接关系



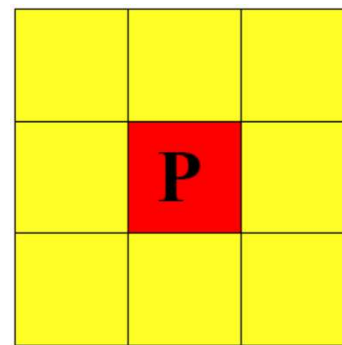
像素邻域



四邻域



D邻域



八邻域

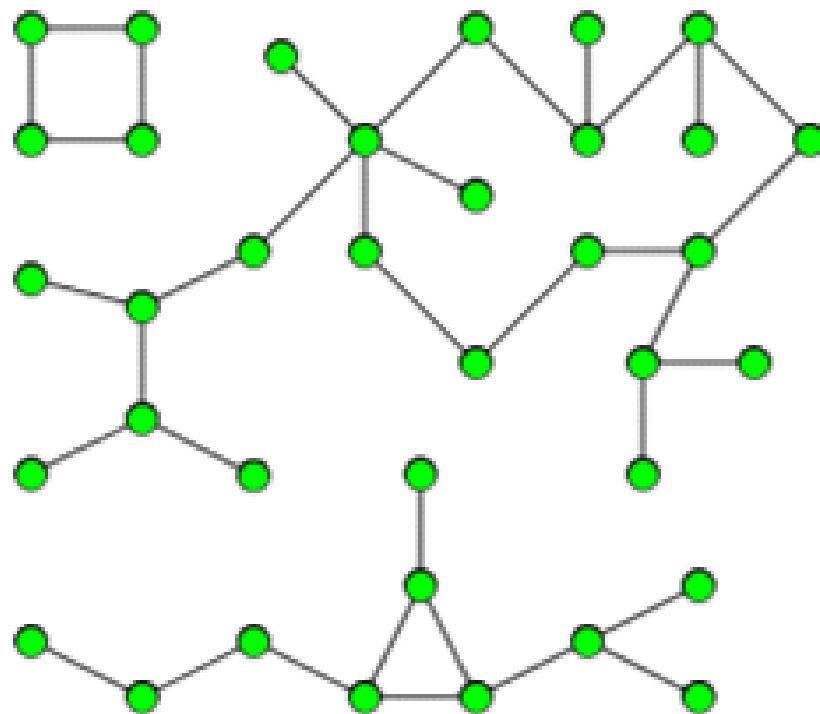


图像 & 图

- 图像可以被看作是一张图，每个像素为一个结点，像素之间根据邻接关系和颜色值等以边相连：
 - 连通域
 - 最短路径/距离
 - 图切割
 - ...

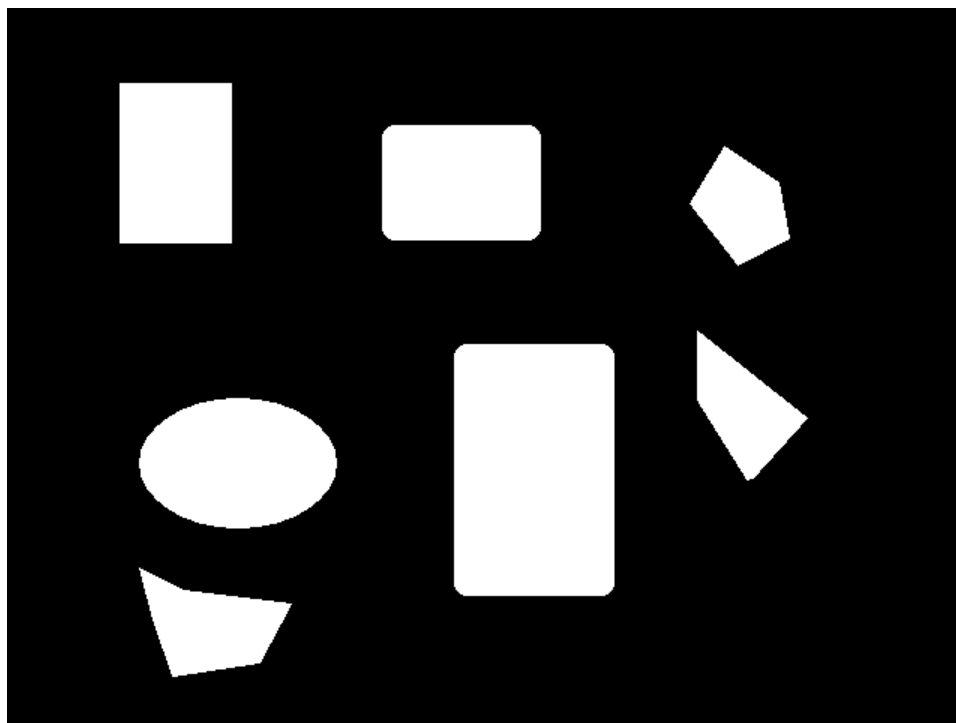
图的连通域

- 同一连通域中的任意两结点都能以路径相连




图像连通域

- D4/D8/Dm连通
- 根据相邻像素的连通性，搜索连通区域



? ?

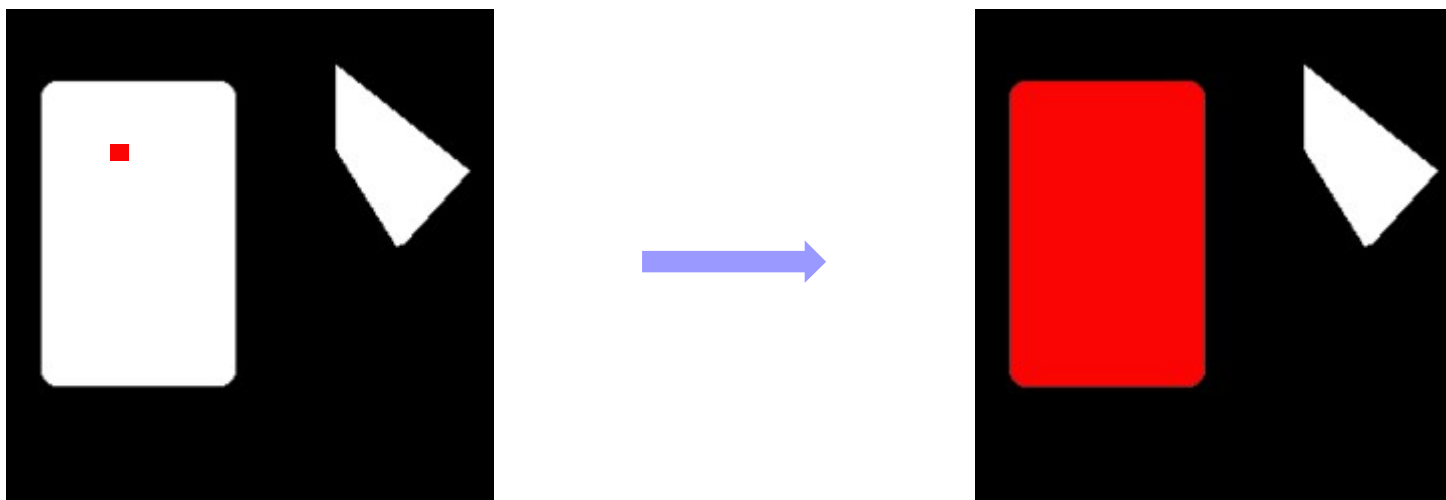


方法? ?

■

区域增长/种子填充

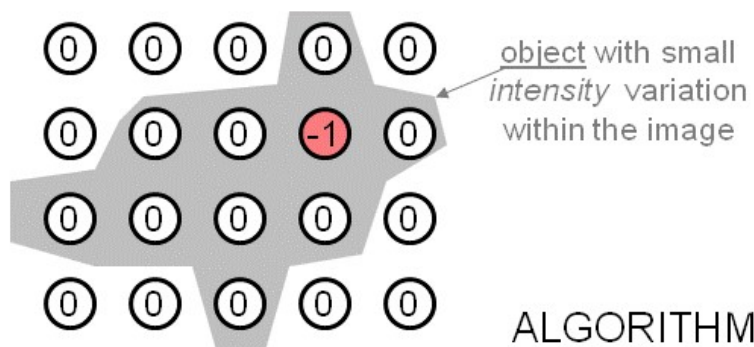
- 基于给定的种子像素，搜索最大连通区域



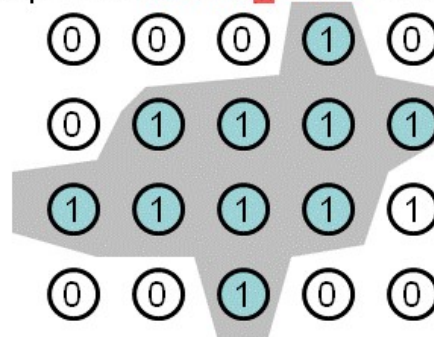
区域增长/种子填充

growRegion: red nodes are the “active_front” (queue or stack)

add seed into **active_front**



stop when **active_front** is empty



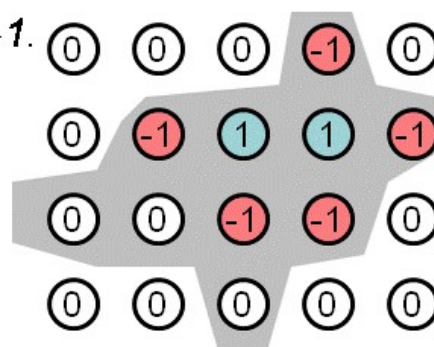
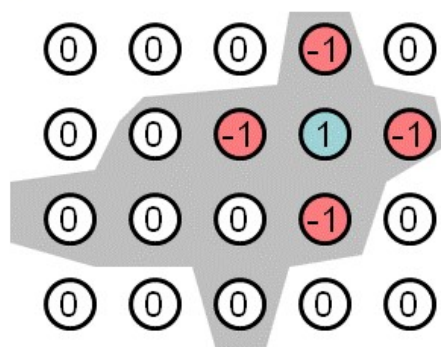
ALGORITHM:

Remove pixel p from **active_front** and mark it as $region[p] = 1$.

Add all neighbors q such that:

$region[q] == 0, |I_p - I_q| < T$

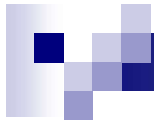
and set $region[q] = -1$.



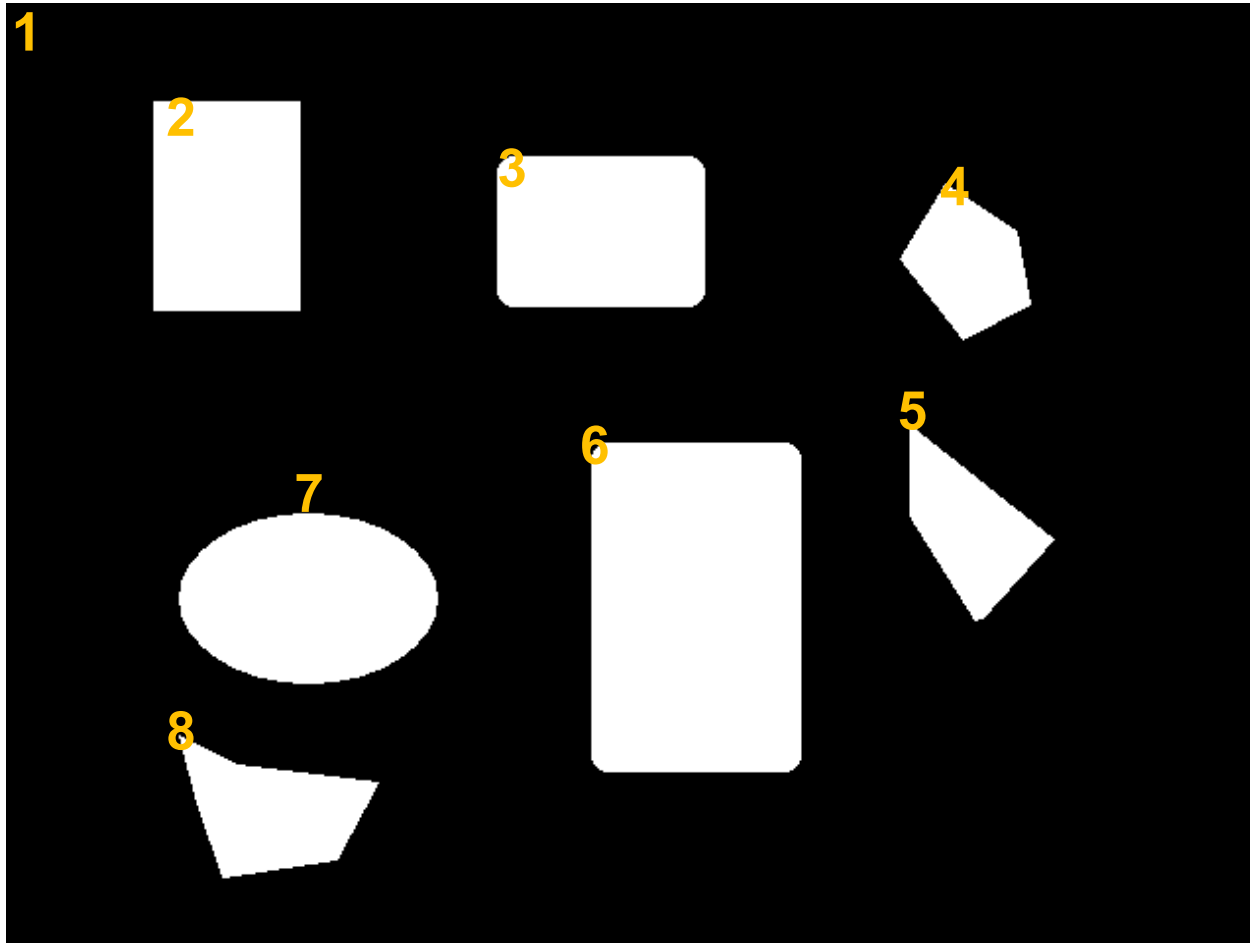


实现

- Push seeds into the stack
- While (stack is not empty)
 1. $s = \text{pop stack}$
 2. label s
 3. push unlabelled neighbors of s into the stack



图像连通域



快速连通域算法

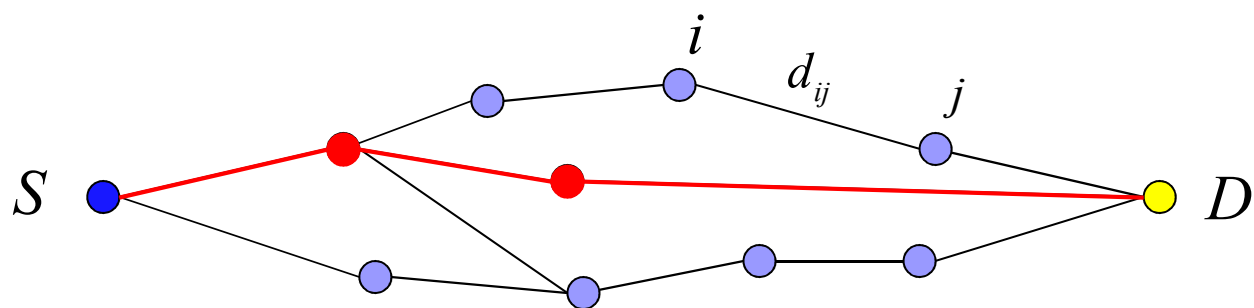
■ 一次扫描 + 合并等价类

					q		
				p	x		

- $x \neq p \ \&\& \ x \neq q$
- $x = p \ \text{或} \ x = q$
- $x = p = q, \ L(p) = L(q)$
- $x = p = q, \ L(p) \neq L(q)$

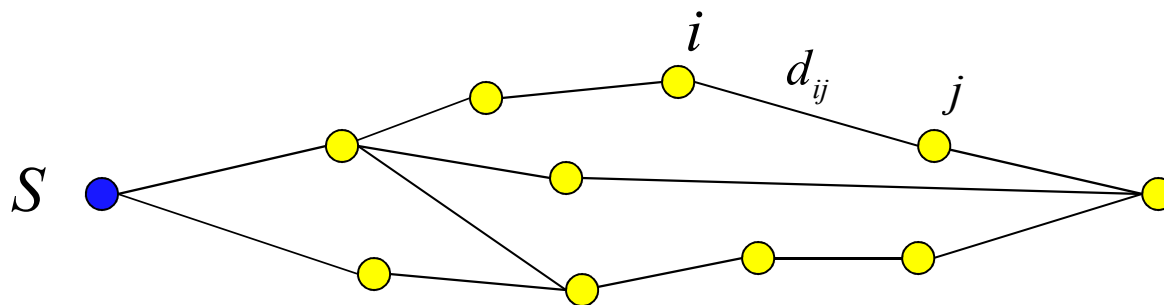
最短路径

■ 两点之间



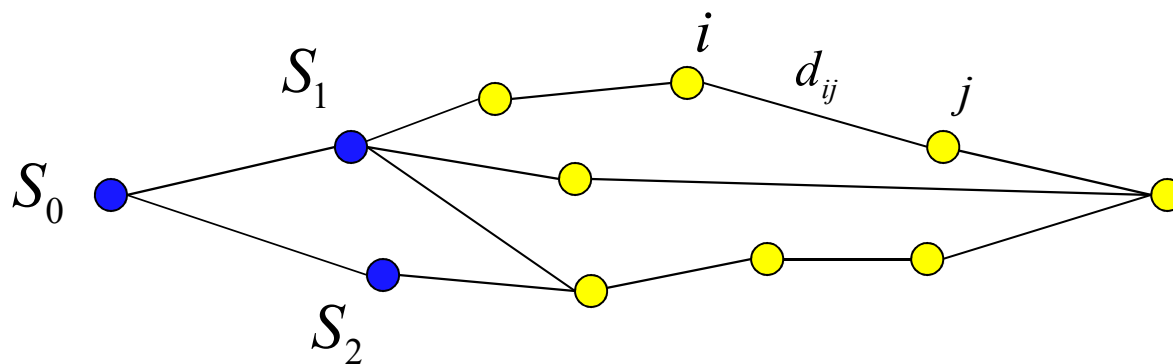
最短路径

- 单源最短路径：所有结点到某个结点的最短路径 (Dijkstra)



最短路径

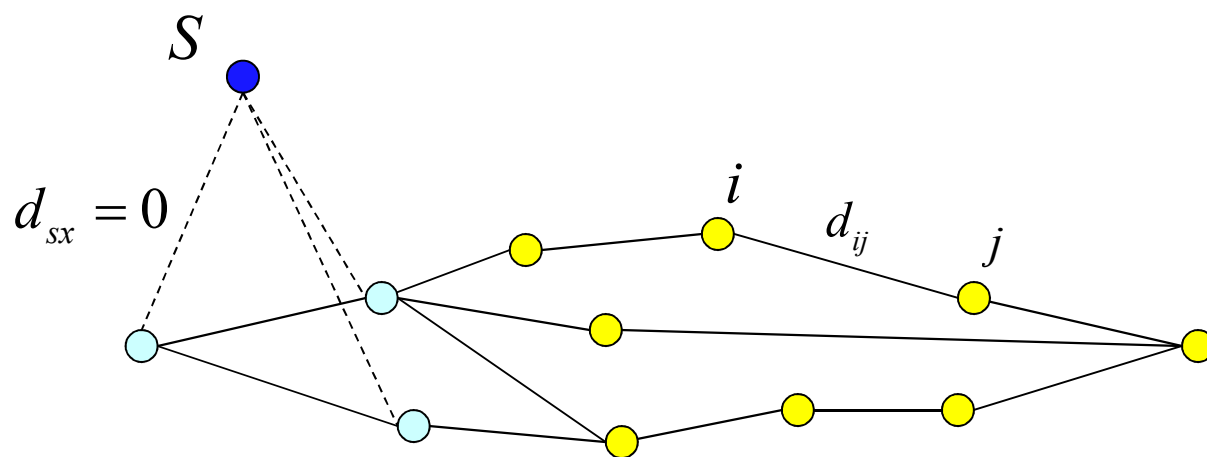
■ 多源??



求每个结点到 S_0 S_1 S_2 的最短路径中最短的一条

最短路径

- 多源 \Rightarrow 单源



像素之间的最短路径

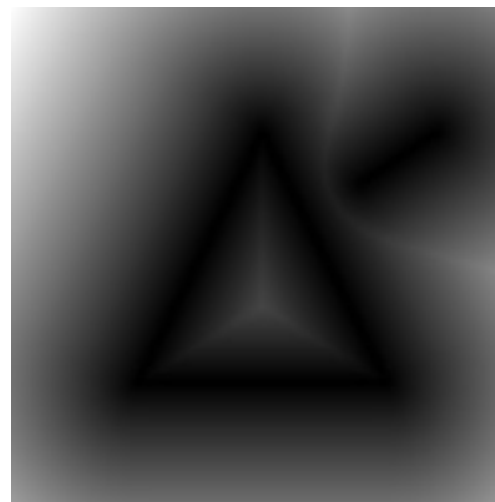
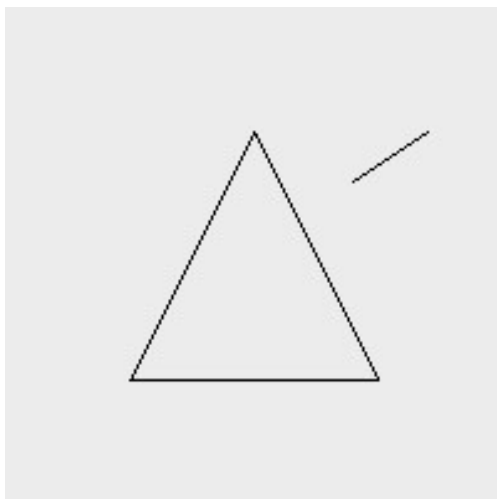
- 定义相邻像素之间的距离为其颜色差的函数，颜色差越大，距离越小：



$$d_{ij} \propto e^{-\beta \|I_i - I_j\|^2}$$

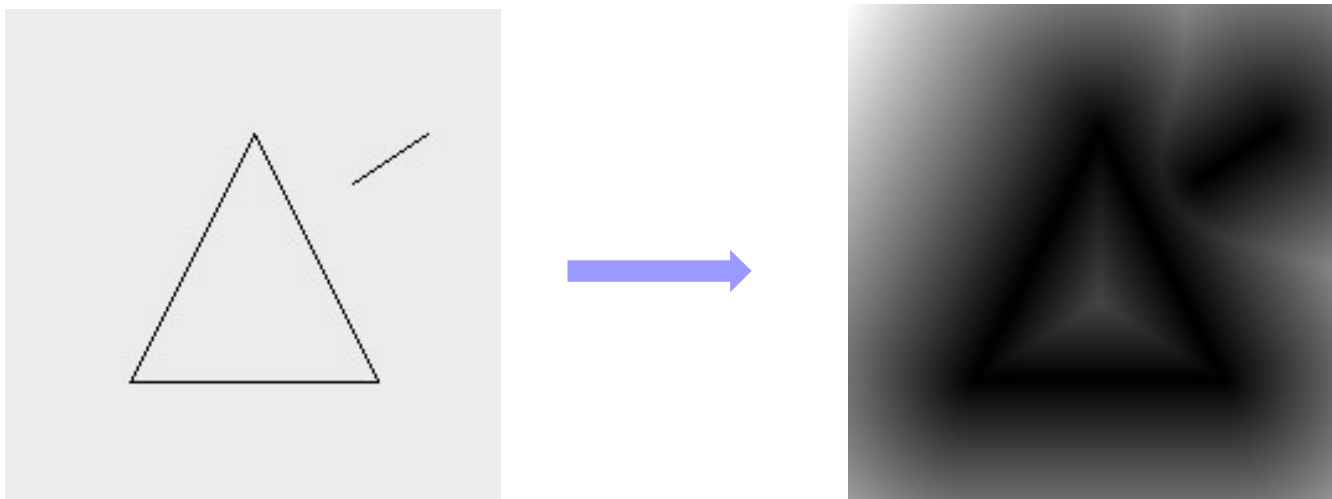
图像距离场

- 所有像素到种子像素的最短距离：



距离变换

- 利用像素规则布局的特点，对图像求距离场的快速方法



$O(n)$ 距离变换 (一维)

1. Initialize: For all j

- $D[j] \leftarrow 0$ or ∞ // 0 if j is seed, ∞ otherwise

2. Forward: For j from 1 up to $n-1$

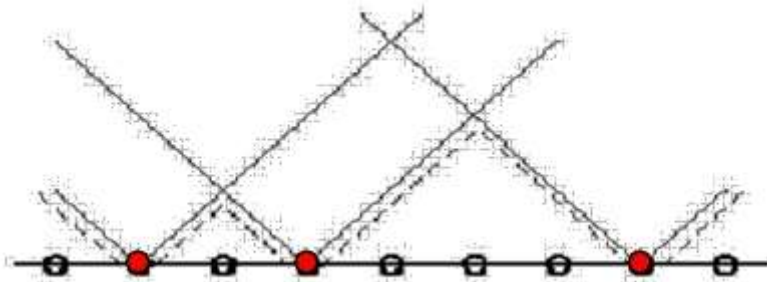
- $D[j] \leftarrow \min(D[j], D[j-1]+1)$

+1	0
----	---

3. Backward: For j from $n-2$ down to 0

- $D[j] \leftarrow \min(D[j], D[j+1]+1)$

0	+1
---	----



∞	0	∞	0	∞	∞	∞	0	∞
----------	---	----------	---	----------	----------	----------	---	----------

∞	0	1	0	1	2	3	0	1
----------	---	---	---	---	---	---	---	---

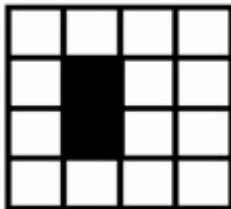
1	0	1	0	1	2	1	0	1
---	---	---	---	---	---	---	---	---

$O(n)$ 距离变换 (二维)

- Initialization
- Forward and backward pass
 - Fwd pass finds closest above and to the left
 - Bwd pass finds closest below and to the right

0	1
1	-

-	1
1	0



∞	∞	∞	∞
∞	0	∞	∞
∞	0	∞	∞
∞	∞	∞	∞

∞	∞	∞	∞
∞	0	1	∞
∞	0	∞	∞
∞	∞	∞	∞

∞	∞	∞	∞
∞	0	1	2
∞	0	1	2
∞	1	2	3

2	1	2	3
1	0	1	2
1	0	1	2
2	1	2	3

$O(n)$ 距离变换 (二维)

(0,0)

0	1	2
1	1.4	2.1969
2	2.1969	2.8

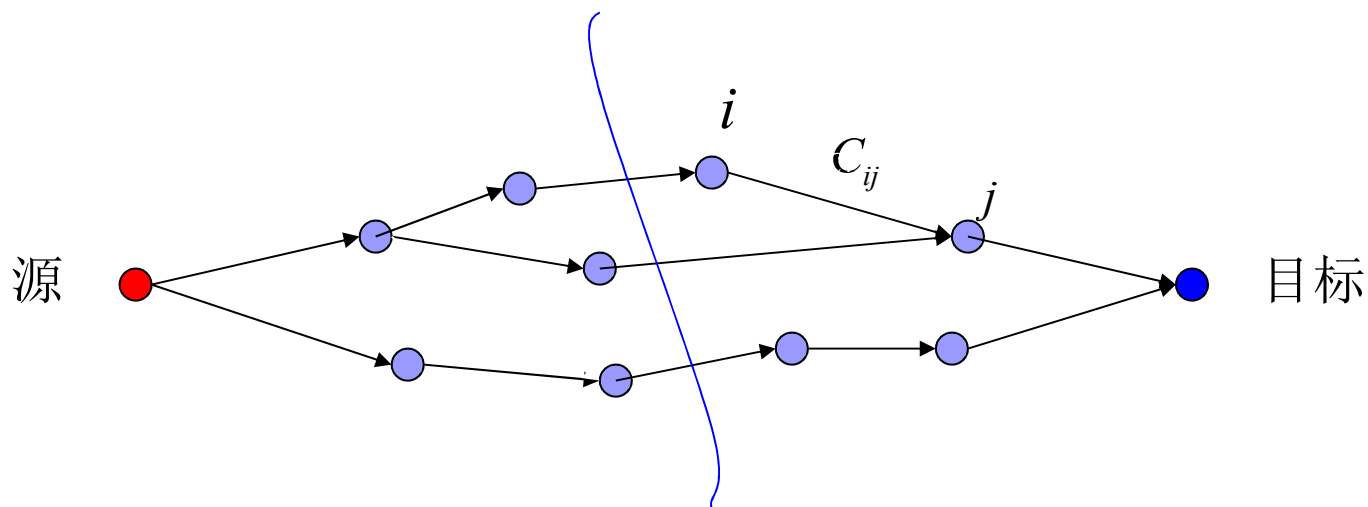
(2,2)

3*3 欧氏距离模板

O(n) 距离变换 (二维)

3,5969	2,8	2,1969	1,4	1	1,4 1,4142	2,1969 2,2361	2,8 2,8284	3,5969 3,6051	4,3938 4,4721
3,1969	2,1969	1,4	1	0	1	1,4 1,4142	2,1969 2,2361	3,1969 3,1623	4,1969 4,1231
3	2	1	0	0	0	1	2	3	4
3	2	1	0	0	0	1	2	3	4
3,1969	2,1969	1,4	1	0	1	1,4	2,1969	3,1969	4,1969
3,5969	2,8	2	1	0	1	2	2,8	3,5969	4,3938
4,1969	3,1969	2,1969	1,4	1	1,4	2,1969	3,1969	4,1969	4,9969

图切割/最小割

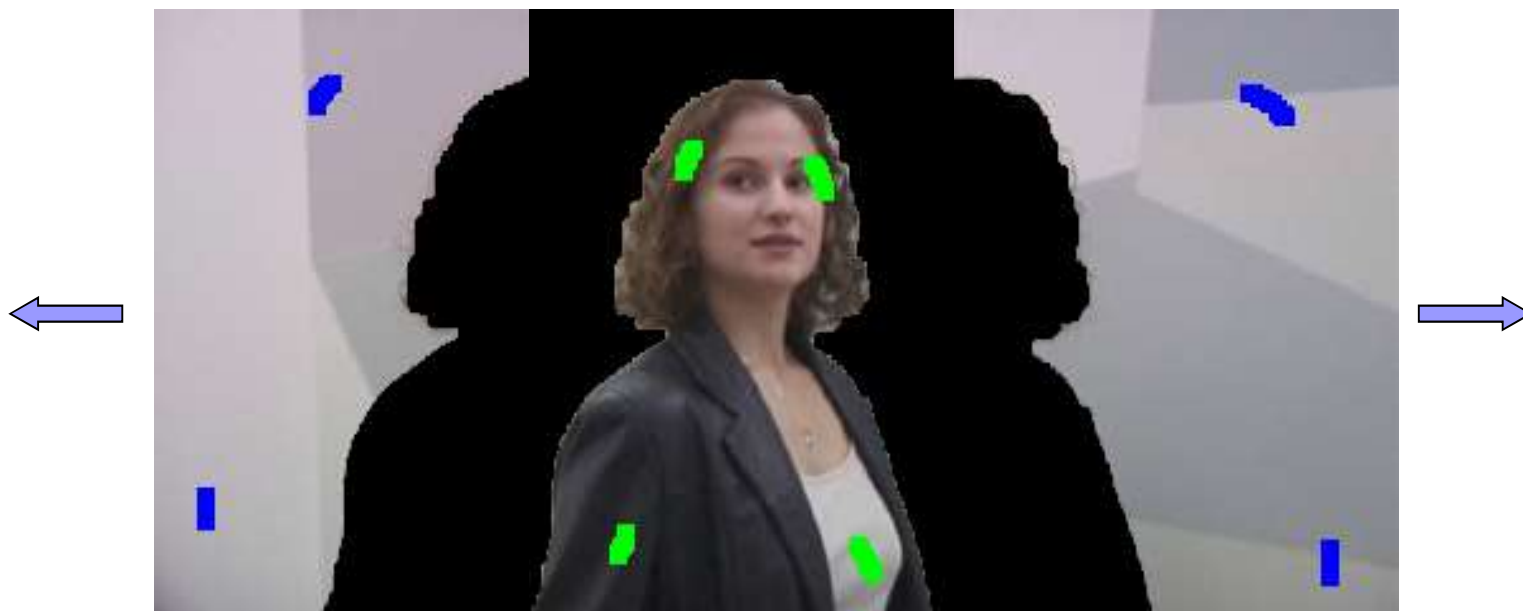


边的容量：每一条边允许通过的最大流量（ C_{ij} ）；

图的切割：能将源和目标之间所有路径切断的图的剖分；

最小割：所有图的切割中所切断的边的容量之和最小的一个，对应于网络流的瓶颈，即最大流；

图像切割（分割）





形态学与二值图结构



形态学图像处理

■ 膨胀和腐蚀

- 膨胀：使图像扩大

A和B是两个集合，A被B膨胀定义为：

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

- 上式表示：B的反射进行平移与A的交集不为空
- B的反射：相对于自身原点的映象
- B的平移：对B的反射进行位移

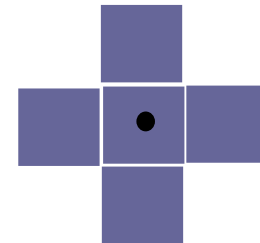
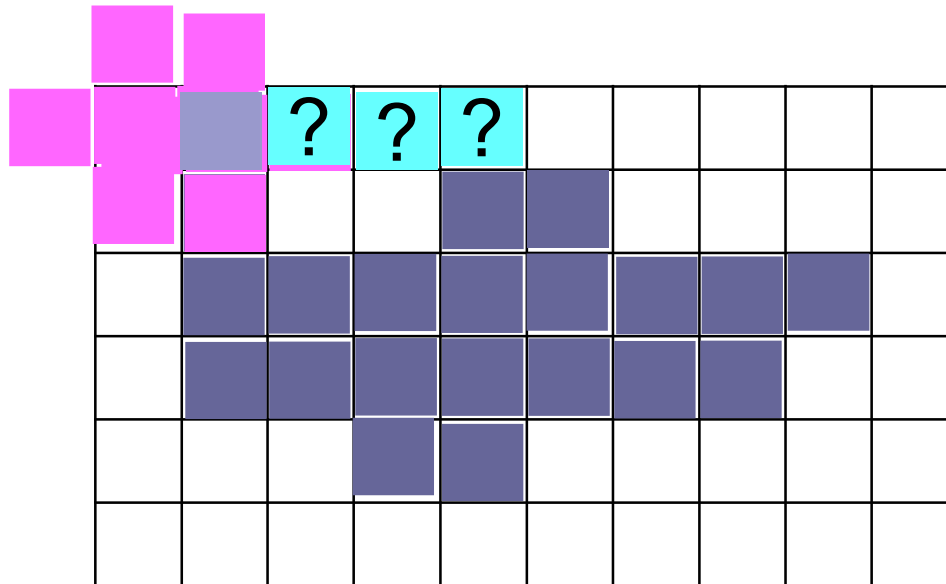
形态学图像处理

■ 膨胀和腐蚀

□ 膨胀的另一个定义

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \subseteq A\}$$

□ 上式表示：B的反射进行平移与A的交集是A的子集



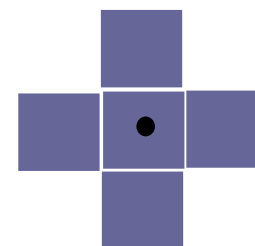
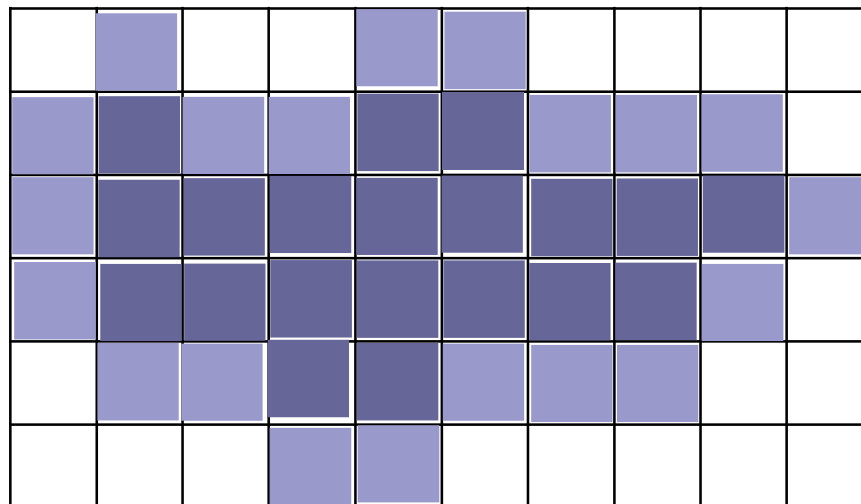
形态学图像处理

■ 膨胀和腐蚀

□ 膨胀的另一个定义

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \subseteq A\}$$

□ 上式表示：B的反射进行平移与A的交集是A的子集



膨胀举例

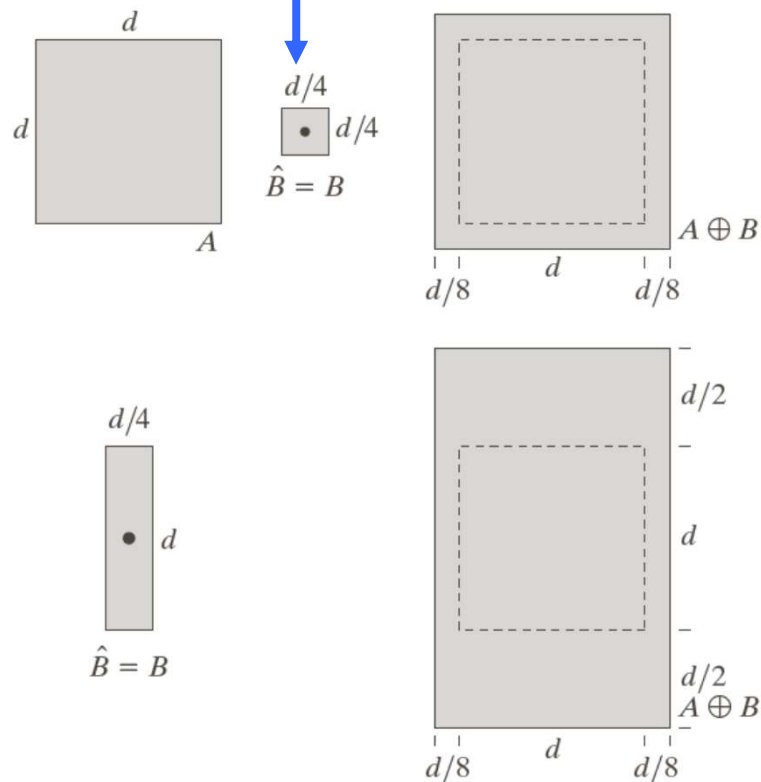
- 相当于拿集合B对集合A作一遍滤波式加法

a	b	c
d		e

FIGURE 9.6

(a) Set A .
 (b) Square structuring element (the dot denotes the origin).
 (c) Dilation of A by B , shown shaded.
 (d) Elongated structuring element. (e) Dilation of A using this element. The dotted border in (c) and (e) is the boundary of set A , shown only for reference

集合B关于原点对称

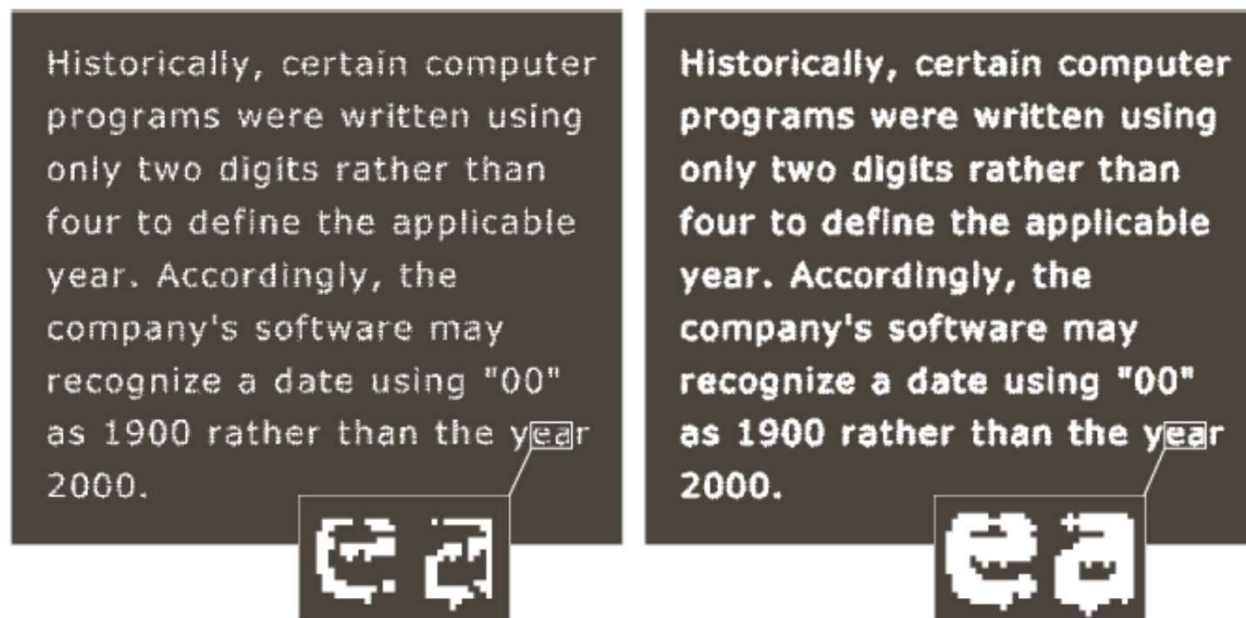


保证B的反射的平移与A的交集不为空

膨胀应用举例

■ 桥接文字裂缝

优点：在一幅二值图像中直接得到结果，可与低通滤波方法对比



a b c

FIGURE 9.7

(a) Sample text of poor resolution with broken characters (see magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

0	1	0
1	1	1
0	1	0



形态学图像处理

■ 膨胀和腐蚀

- 腐蚀：使图像缩小

A和B是两个集合，A被B腐蚀定义为：

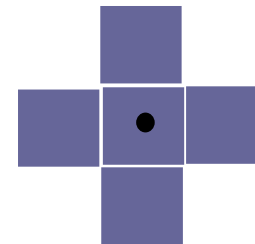
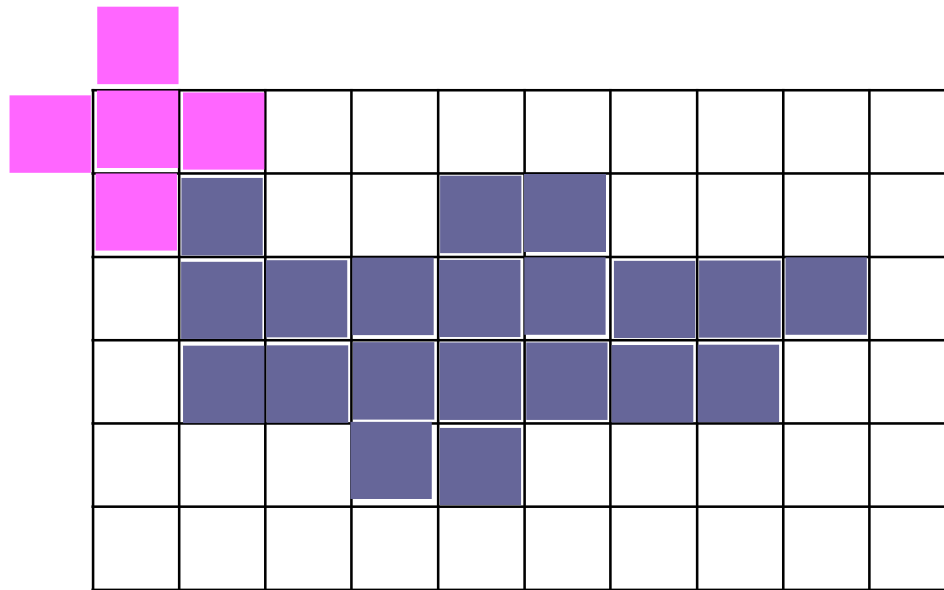
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

- 上式表示：B进行平移后包含于A

形态学图像处理

■ 腐蚀

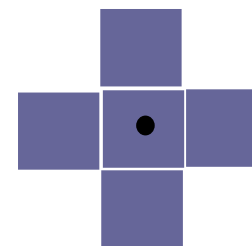
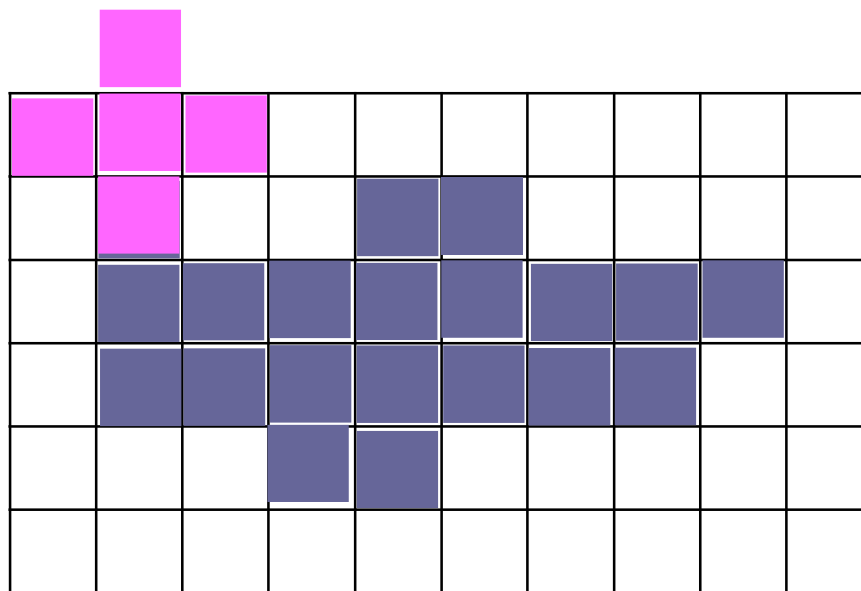
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$



形态学图像处理

■ 腐蚀

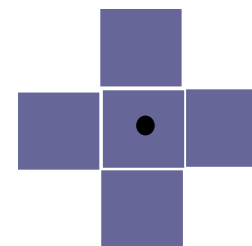
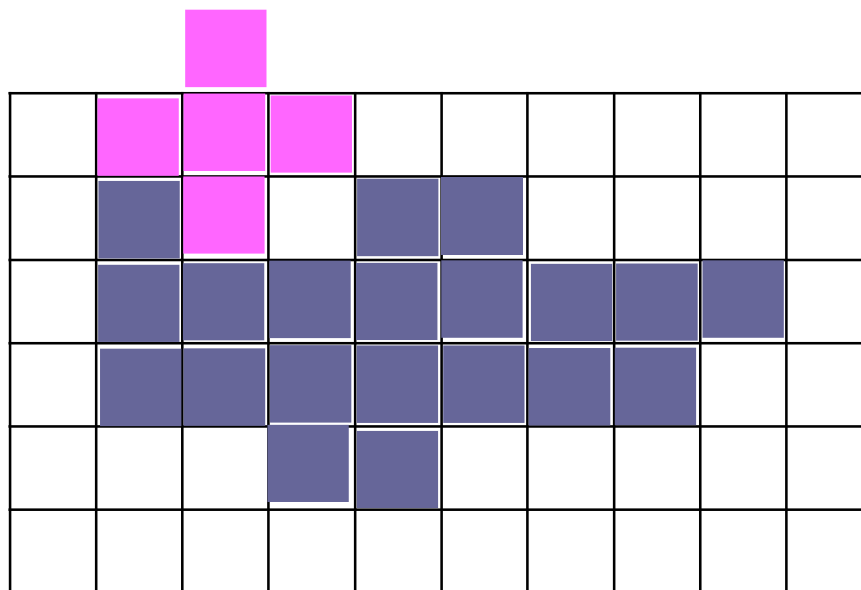
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$



形态学图像处理

■ 腐蚀

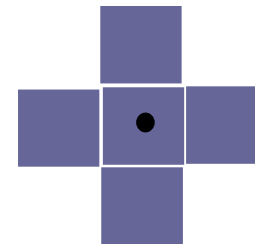
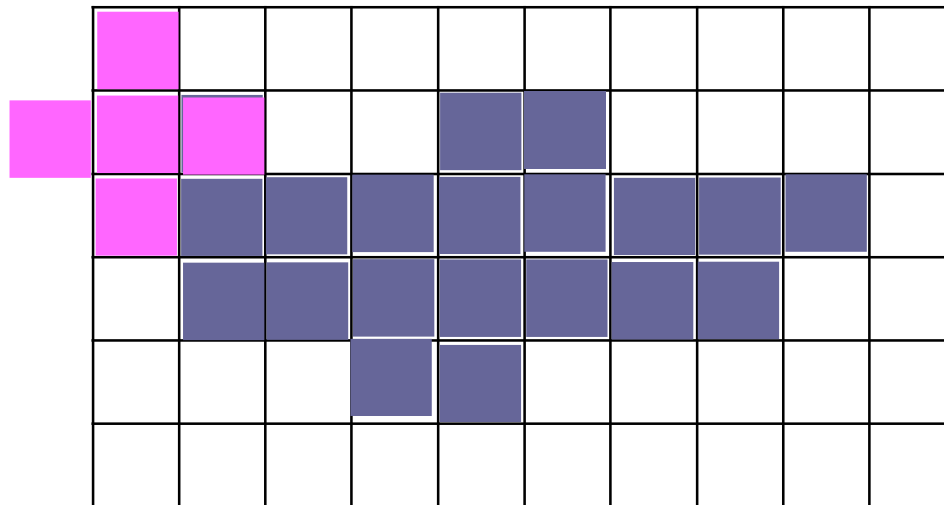
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$



形态学图像处理

■ 腐蚀

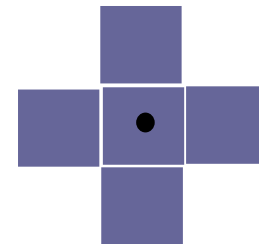
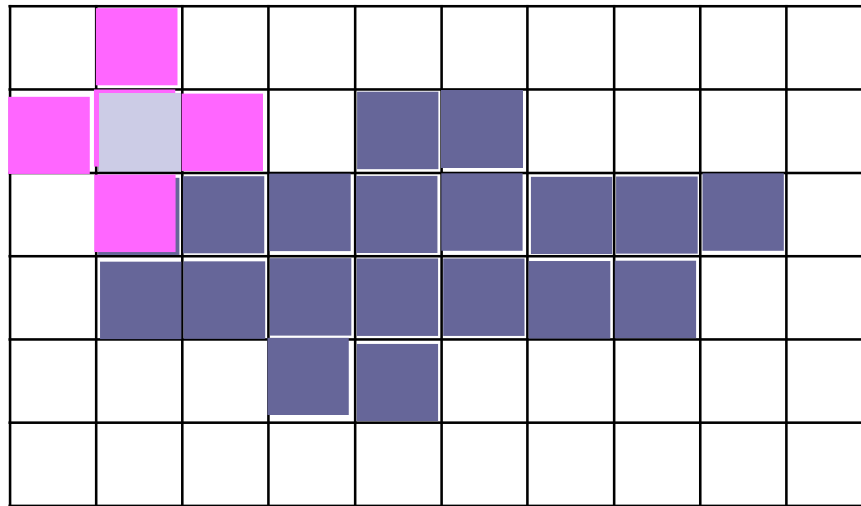
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$



形态学图像处理

■ 腐蚀

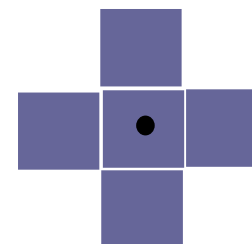
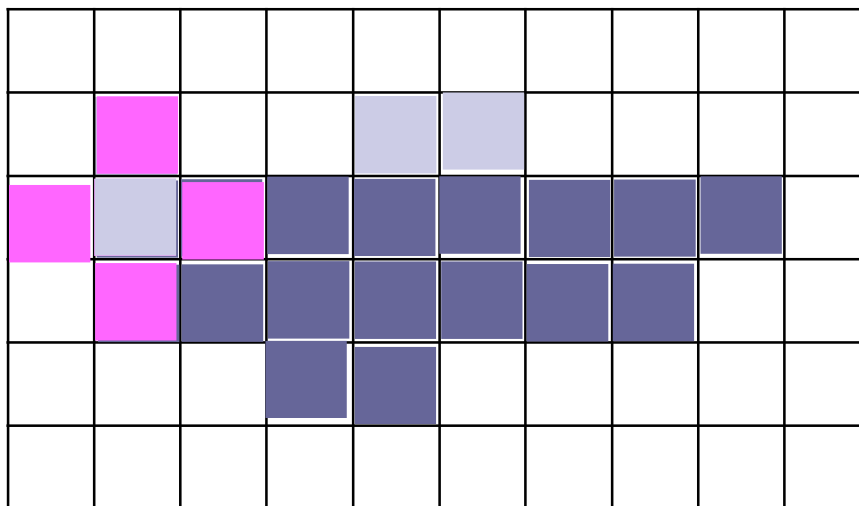
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$



形态学图像处理

■ 腐蚀

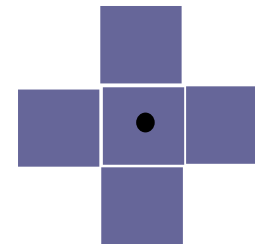
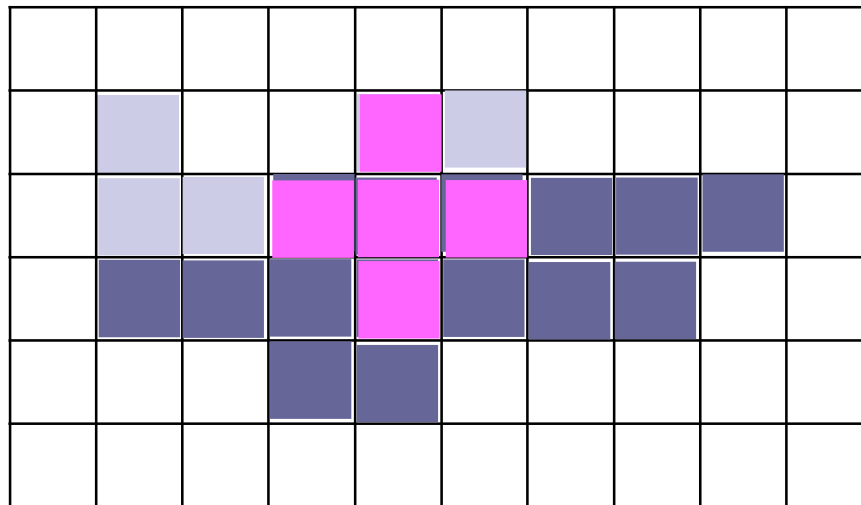
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$



形态学图像处理

■ 腐蚀

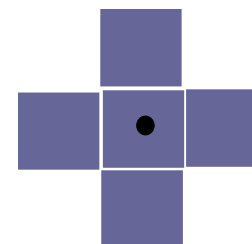
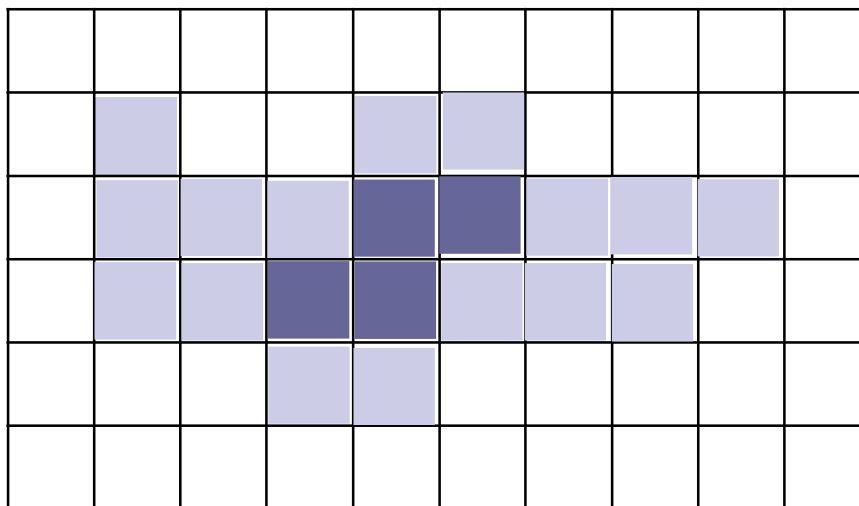
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$



形态学图像处理

■ 腐蚀

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



腐蚀举例

- 相当于拿集合B对集合A作一遍滤波式减法

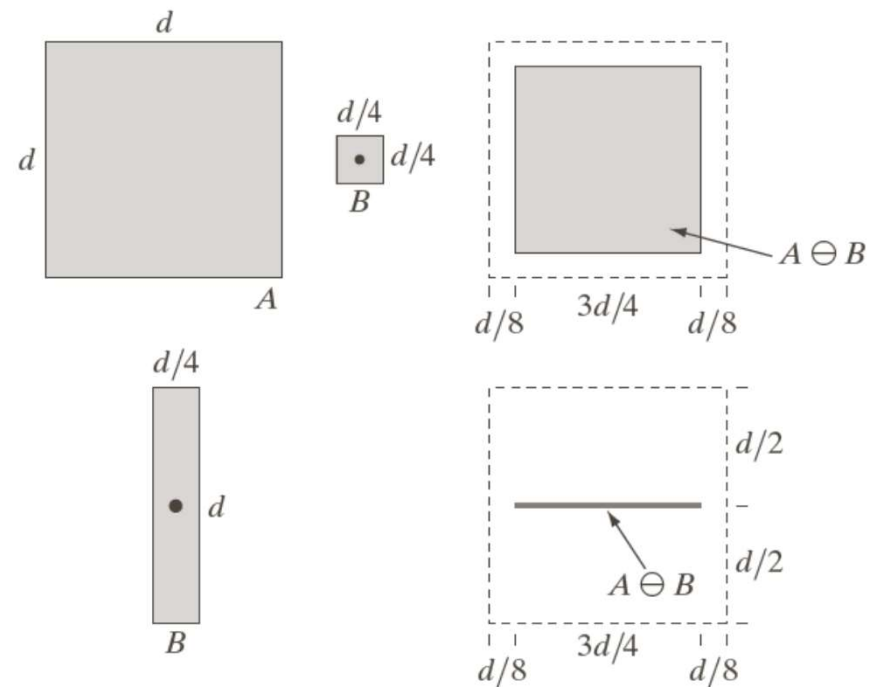


FIGURE 9.4 (a) Set A . (b) Square structuring element, B . (c) Erosion of A by B , shown shaded. (d) Elongated structuring element. (e) Erosion of A by B using this element. The dotted border in (c) and (e) is the boundary of set A , shown only for reference.

腐蚀举例

- 相当于拿集合B对集合A作一遍滤波式减法

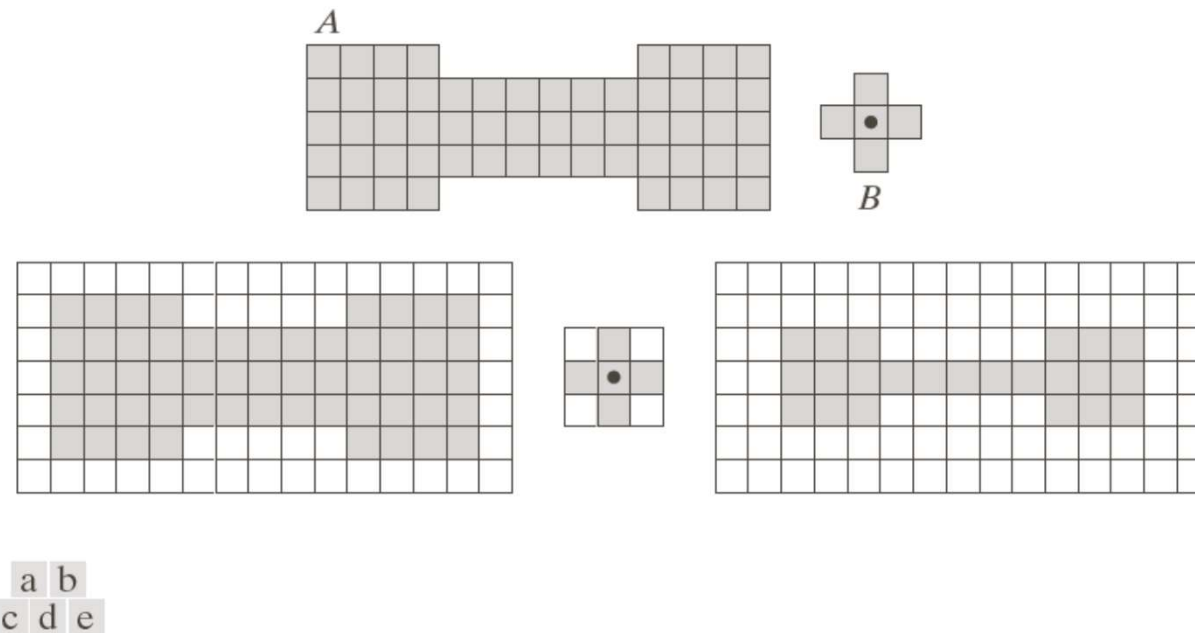
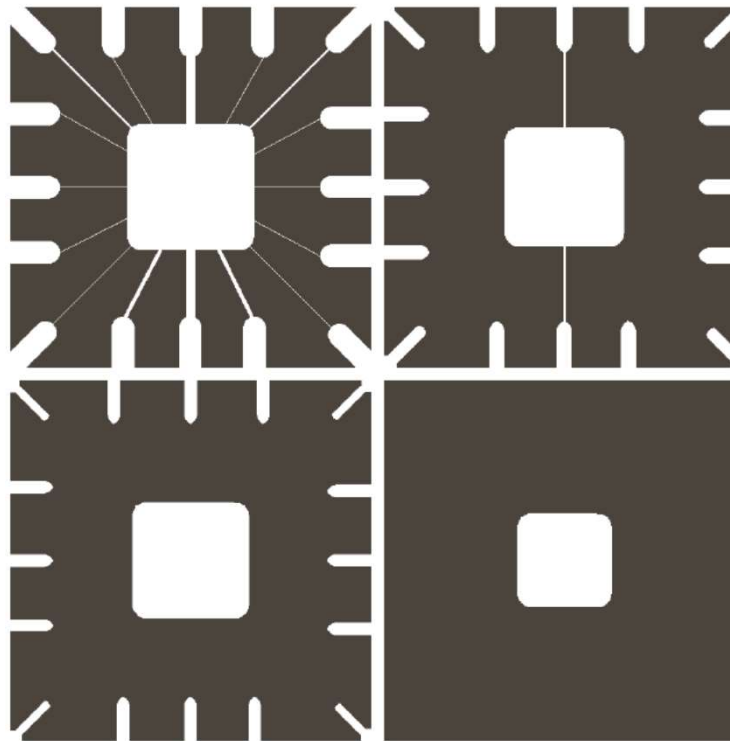


FIGURE 9.3 (a) A set (each shaded square is a member of the set). (b) A structuring element. (c) The set padded with background elements to form a rectangular array and provide a background border. (d) Structuring element as a rectangular array. (e) Set processed by the structuring element.

腐蚀举例

- 用腐蚀的方法去掉不同粗细的区域



a	b
c	d

FIGURE 9.5 Using erosion to remove image components. (a) A 486×486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11×11 , 15×15 , and 45×45 , respectively. The elements of the SEs were all 1s.

腐蚀应用举例

■ 使用腐蚀消除图像的细节部分，产生滤波器的作用

包含边长为1,3,5,7,9
和15像素正方形的二
值图像

使用 13×13 像素大小
的结构元素腐蚀原图
像的结果

使用 13×13 像素大小的结
构元素膨胀图b，恢复原来
 15×15 尺寸的正方形

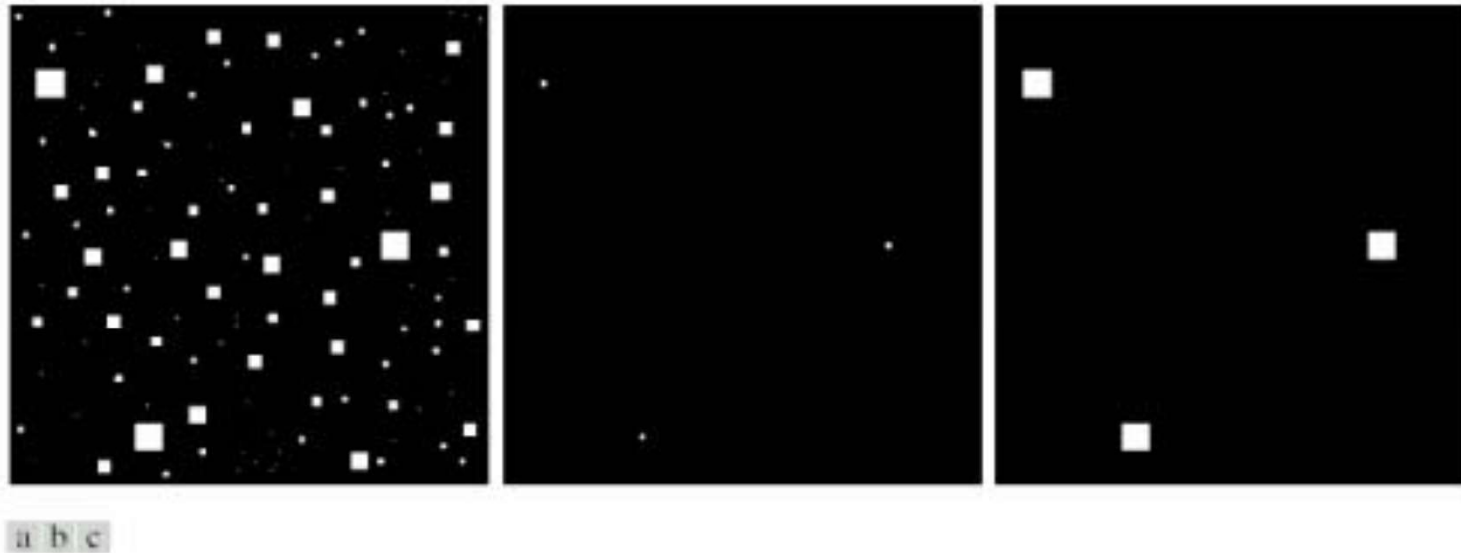


FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

开操作和闭操作

■ 开操作

- 在不改变形状的前提下，使图像的轮廓变得光滑
- 断开狭窄的间断
- 消除细的突出物

■ 使用结构元素B对集合A进行开操作，定义为：

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

- 含义：先用B对A腐蚀，然后用B对结果膨胀
- 另一个定义形

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

开操作的几何解释

- $A \circ B$ 的边界通过B中的点完成
- B在A的边界内转动时，B中的点所能到达的A的边界的最远点

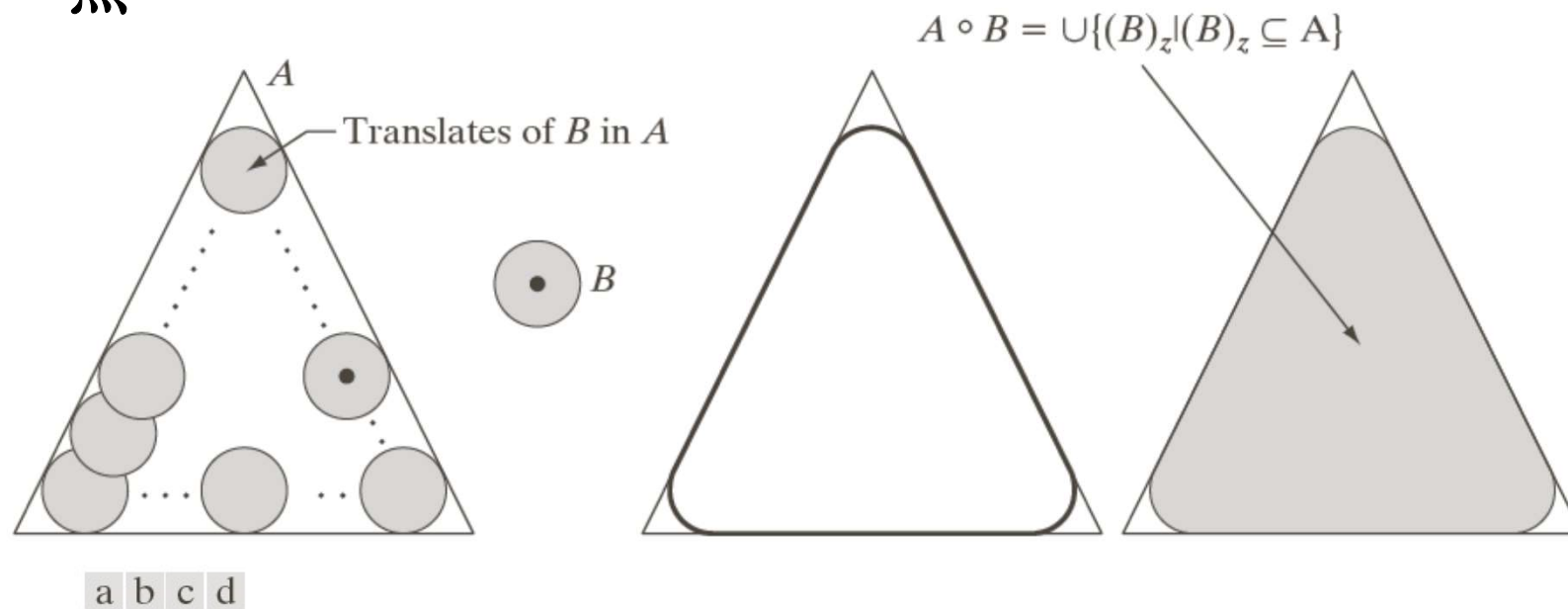


FIGURE 9.8 (a) Structuring element B “rolling” along the inner boundary of A (the dot indicates the origin of B). (b) Structuring element. (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded). We did not shade A in (a) for clarity.



开操作和闭操作

■ 闭操作

- 在不明显改变面积前提下，使图像的轮廓变得光滑
- 消除小的孔洞
- 消除狭窄的间断
- 细长的鸿沟
- 填补轮廓线中的裂痕

■ 使用结构元素B对集合A进行闭操作，定义为：

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

- 含义：先用B对A膨胀，然后用B对结果腐蚀

闭操作的几何解释

- $A \bullet B$ 的边界通过B中的点完成
- B在A的边界外部转动时

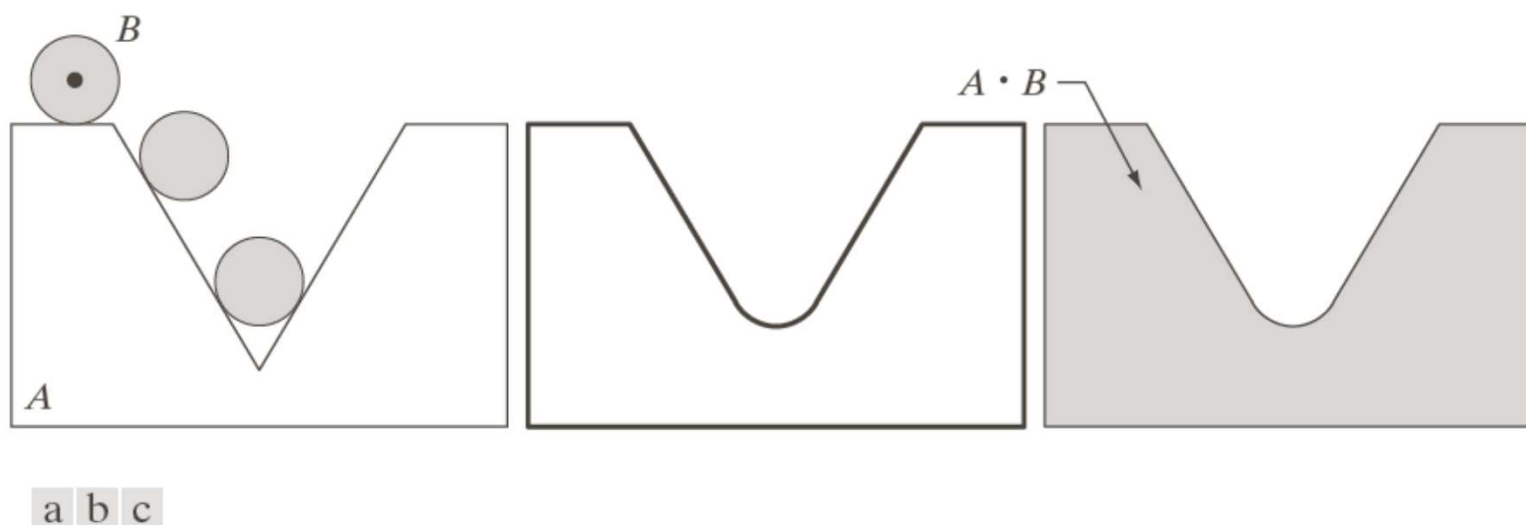


FIGURE 9.9 (a) Structuring element B “rolling” on the outer boundary of set A . (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade A in (a) for clarity.

开/闭操作的性质

■ 开操作的3条性质

- $A \circ B$ 是A的子集合
- 如果C是D的子集, 则 $C \circ B$ 是 $D \circ B$ 的子集

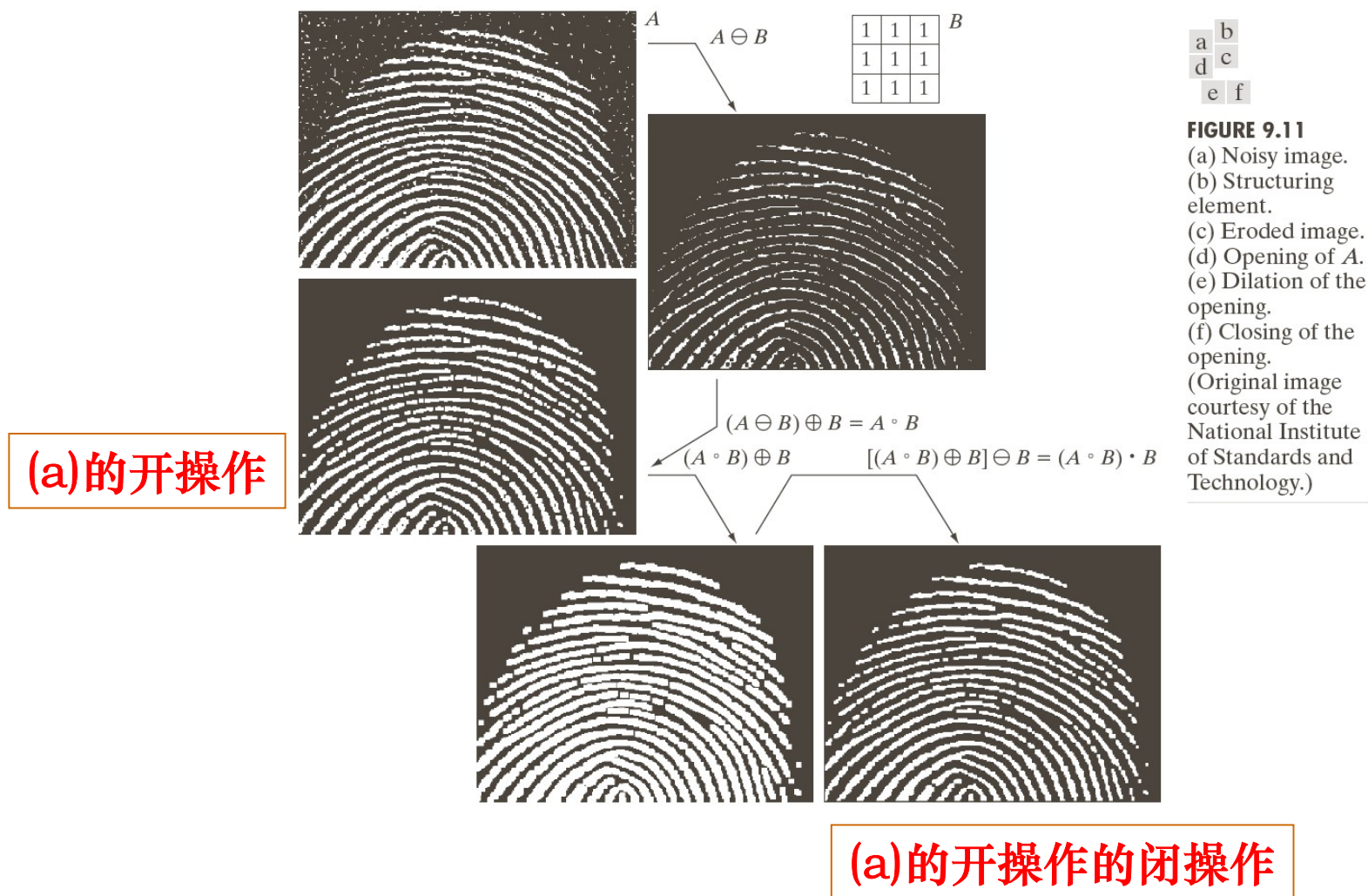
$$(A \circ B) \circ B = A \circ B$$

■ 闭操作的3条性质

- A是 $A \bullet B$ 的子集合
- 如果C是D的子集, 则 $C \bullet B$ 是 $D \bullet B$ 的子集

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

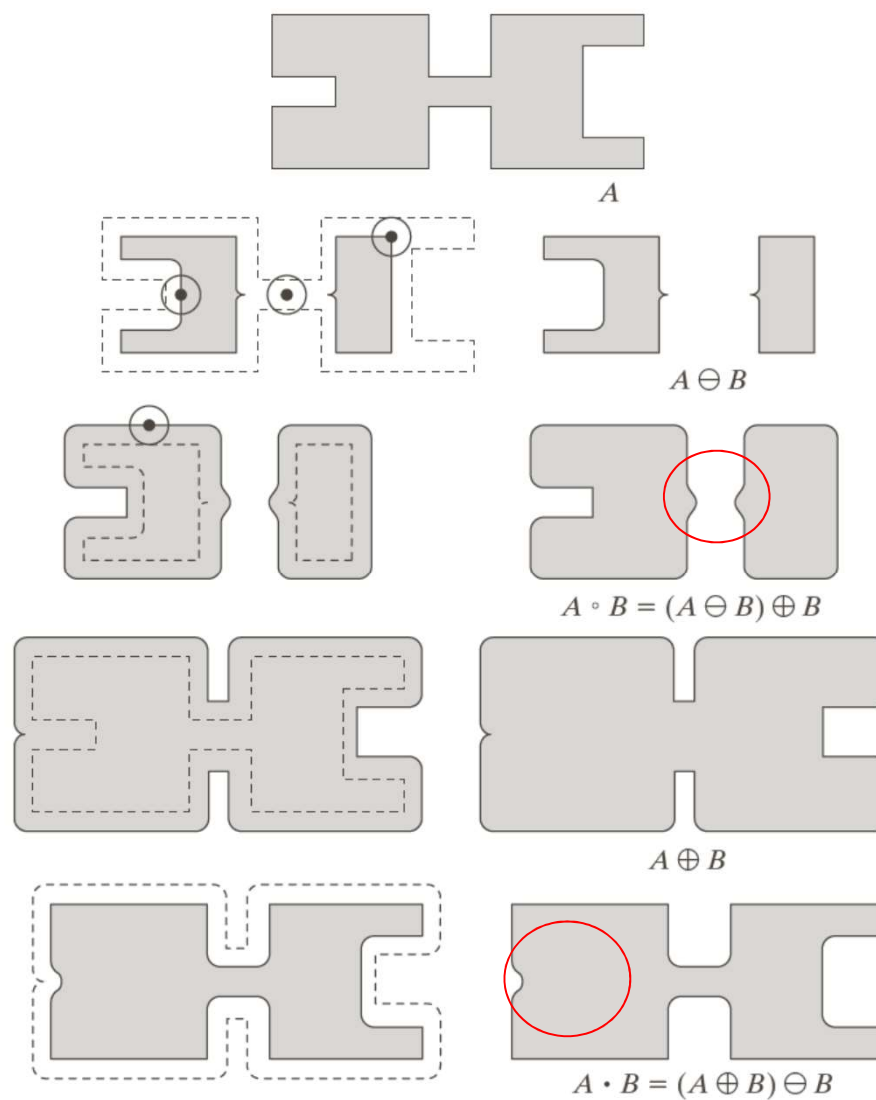
开操作和闭操作的应用举例



开操作和闭操作的应用举例

a
b c
d e
f g
h i

FIGURE 9.10
Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The SE was not shaded here for clarity. The dark dot is the center of the structuring element.



开操作消除
狭长区域

闭操作消除
狭长裂缝

形态学的主要应用

■ 边界提取

□ 边界提取定义为：

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

□ 上式表示：先用B对A腐蚀，然后用A减去腐蚀得到，B是结构元素

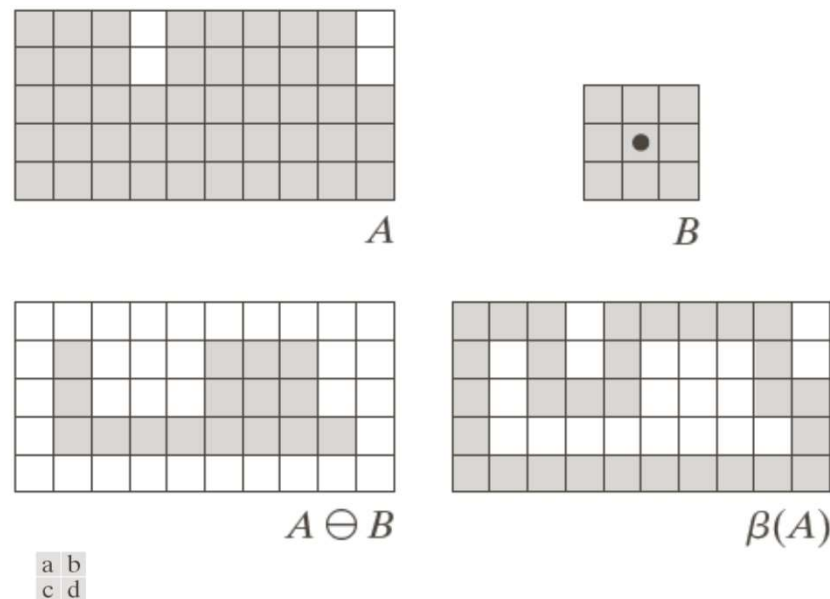
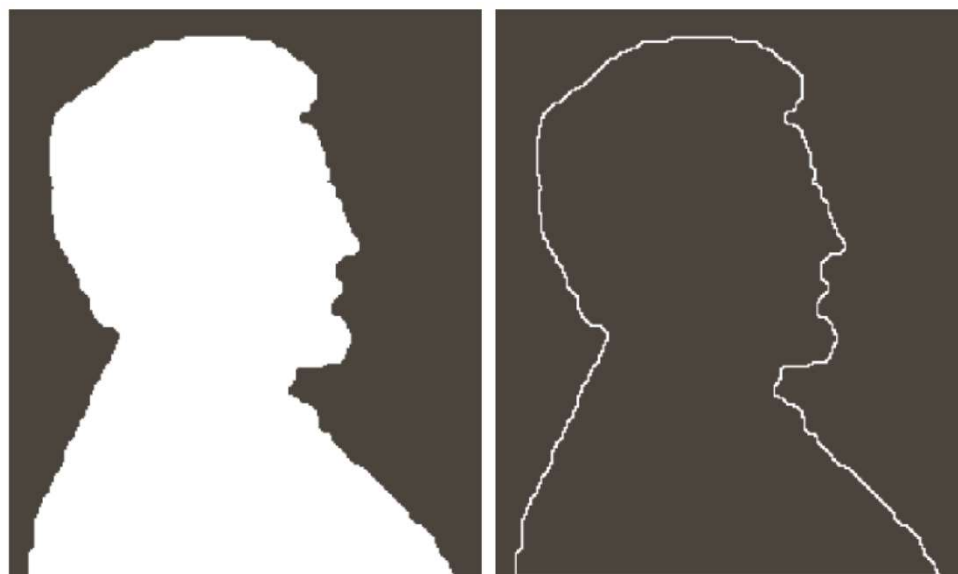
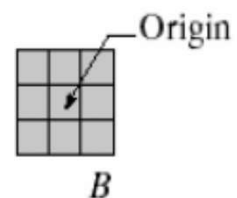


FIGURE 9.13 (a) Set A . (b) Structuring element B . (c) A eroded by B . (d) Boundary, given by the set difference between A and its erosion.

形态学的主要应用

■ 边界提取举例



a b

FIGURE 9.14

(a) A simple binary image, with 1s represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

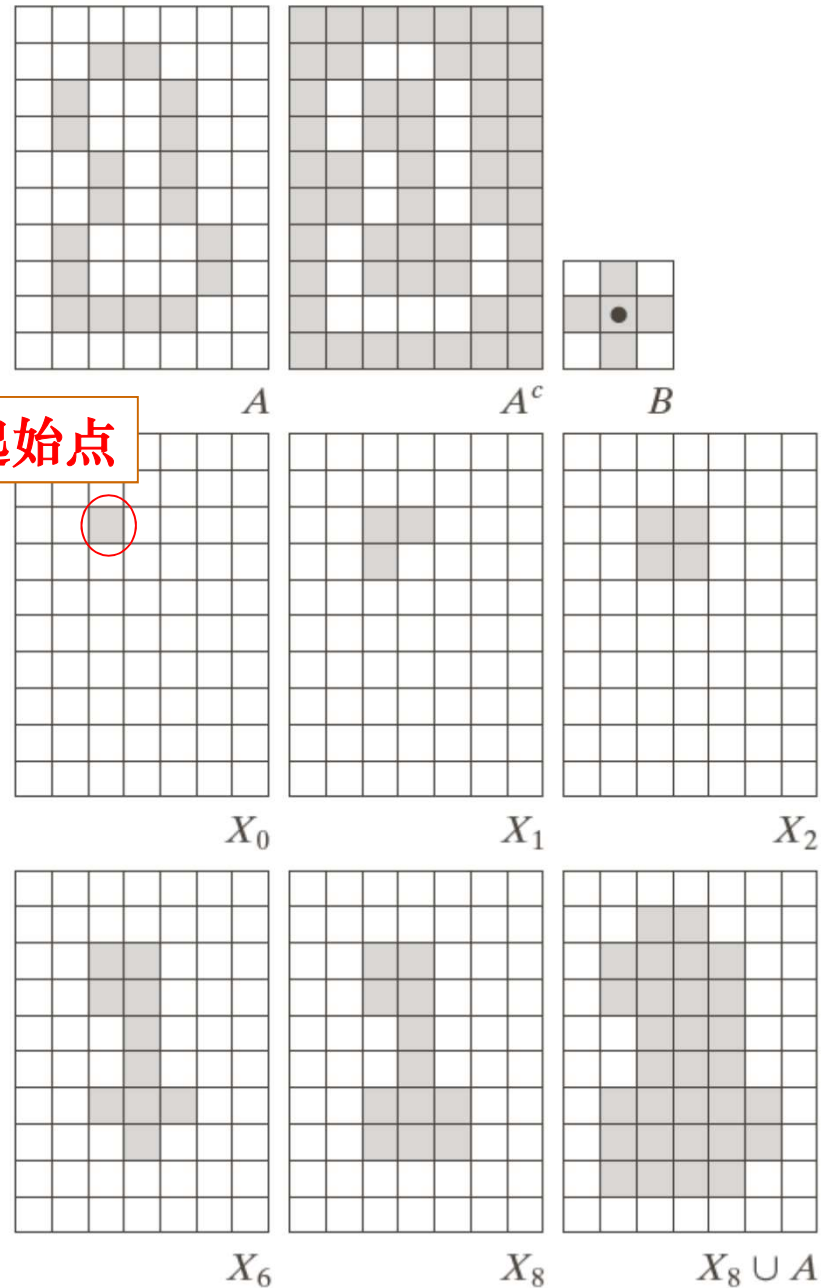
形态学的主要应用

■ 区域填充应用举例

a	b	c
d	e	f
g	h	i

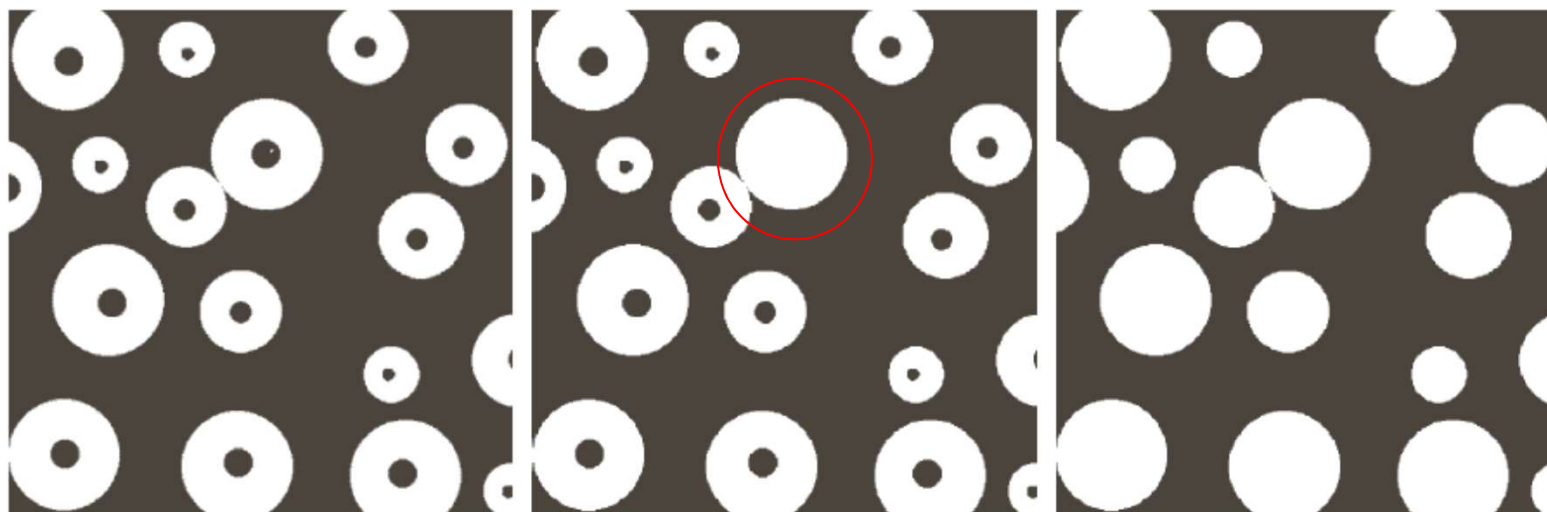
交互定义起始点

FIGURE 9.15 Hole filling. (a) Set A (shown shaded). (b) Complement of A . (c) Structuring element B . (d) Initial point inside the boundary. (e)–(h) Various steps of Eq. (9.5-2). (i) Final result [union of (a) and (h)].



形态学的主要应用

- 通过区域填充消除白色圆圈内的黑点



a b c

FIGURE 9.16 (a) Binary image (the white dot inside one of the regions is the starting point for the hole-filling algorithm). (b) Result of filling that region. (c) Result of filling all holes.

形态学的主要应用

■ 连通分量的提取应用举例

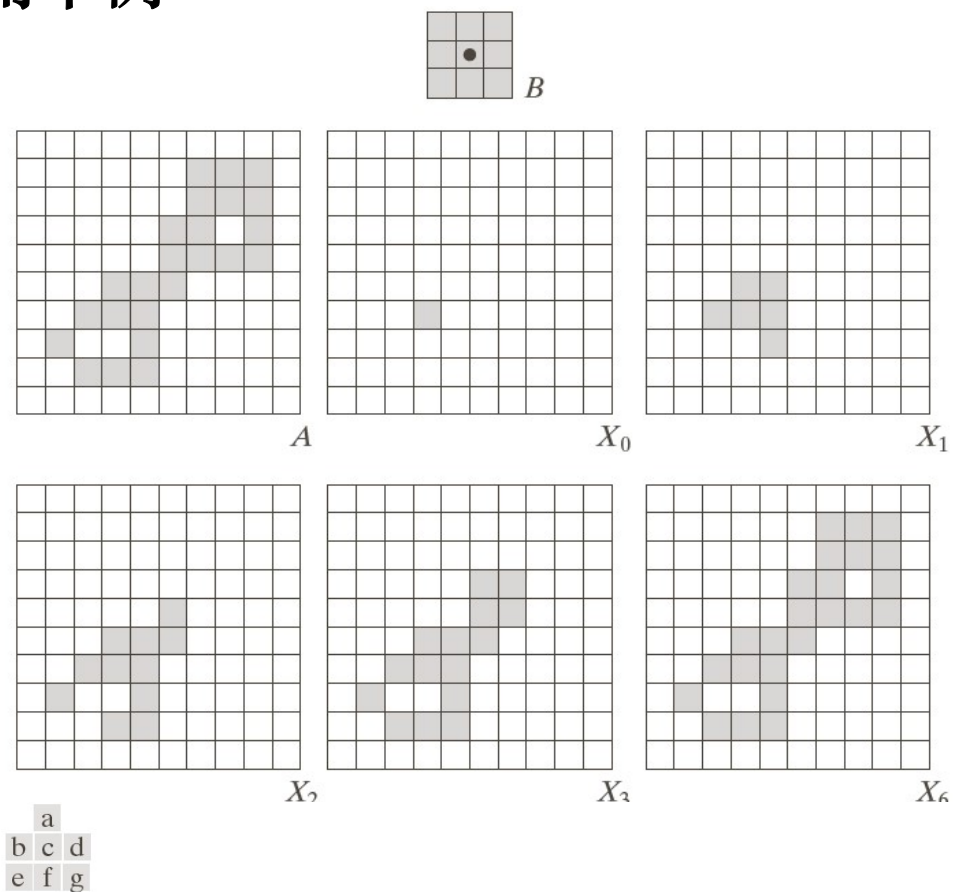


FIGURE 9.17 Extracting connected components. (a) Structuring element. (b) Array containing a set with one connected component. (c) Initial array containing a 1 in the region of the connected component. (d)–(g) Various steps in the iteration of Eq. (9.5-3).

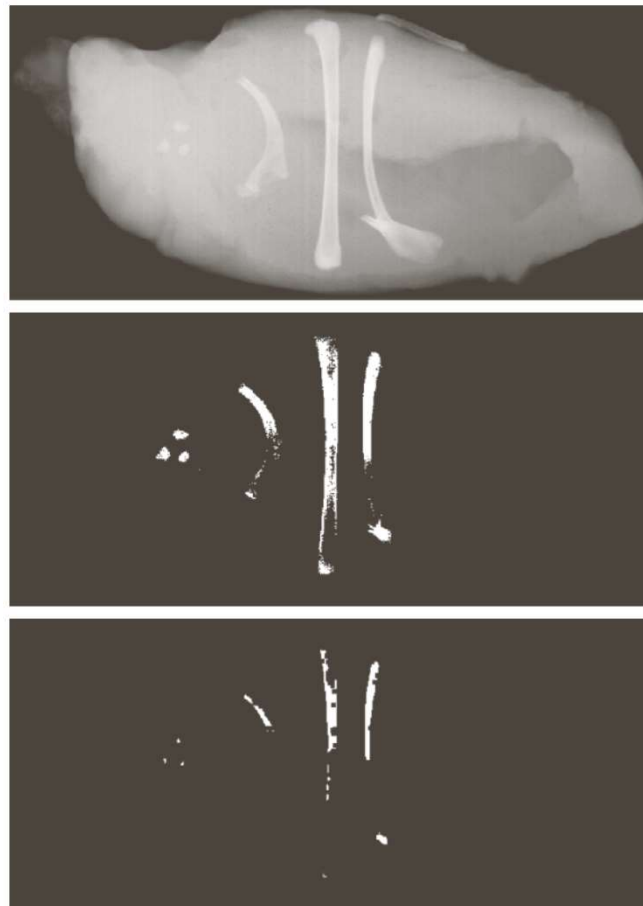
形态学的主要应用

■ 连通分量的提取应用举例

a
b
c d

FIGURE 9.18

(a) X-ray image of chicken filet with bone fragments. (b) Thresholded image. (c) Image eroded with a 5×5 structuring element of 1s. (d) Number of pixels in the connected components of (c). (Image courtesy of NTB Elektronische Geraete GmbH, Diepholz, Germany, www.ntbxray.com.)



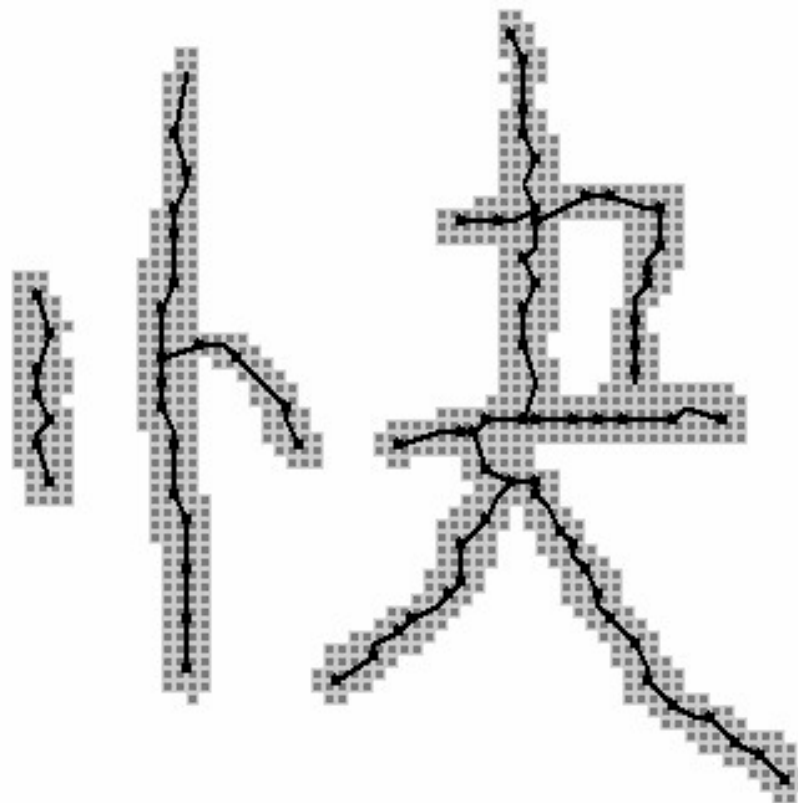
Connected component	No. of pixels in connected comp
01	11
02	9
03	9
04	39
05	<u>133</u>
06	1
07	1
08	<u>743</u>
09	7
10	11
11	11
12	9
13	9
14	<u>674</u>
15	<u>85</u>

骨架

许多图像，其形状足以用其骨架来表示

骨架特点：

- 单像素厚度
- 等距性





扩展至灰度图像

- 膨胀

$$(f \oplus b)(s, t) = \max \{ f(s - x, t - y) + b(x, y) \mid (s - x, t - y) \in D_f; (x, y) \in D_b \}$$

- 腐蚀

$$(f \ominus b)(s, t) = \min \{ f(s + x, t + y) - b(x, y) \mid (s + x, t + y) \in D_f; (x, y) \in D_b \}$$

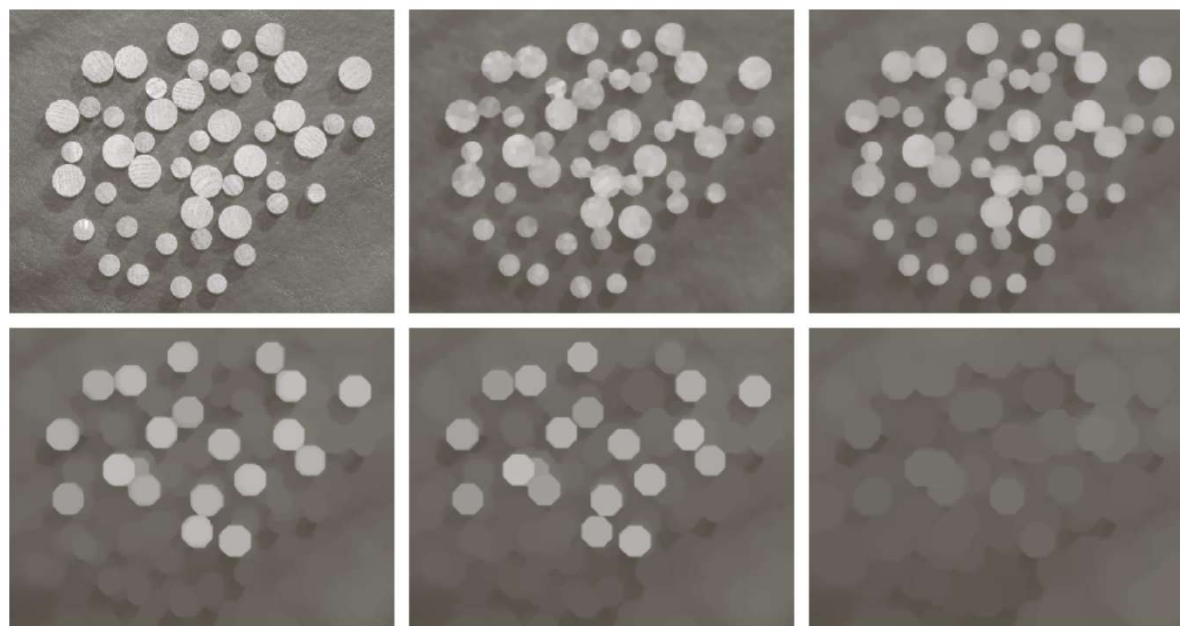
- 开

$$f \circ b = (f \ominus b) \oplus b$$

- 闭

$$f \bullet b = (f \oplus b) \ominus b$$

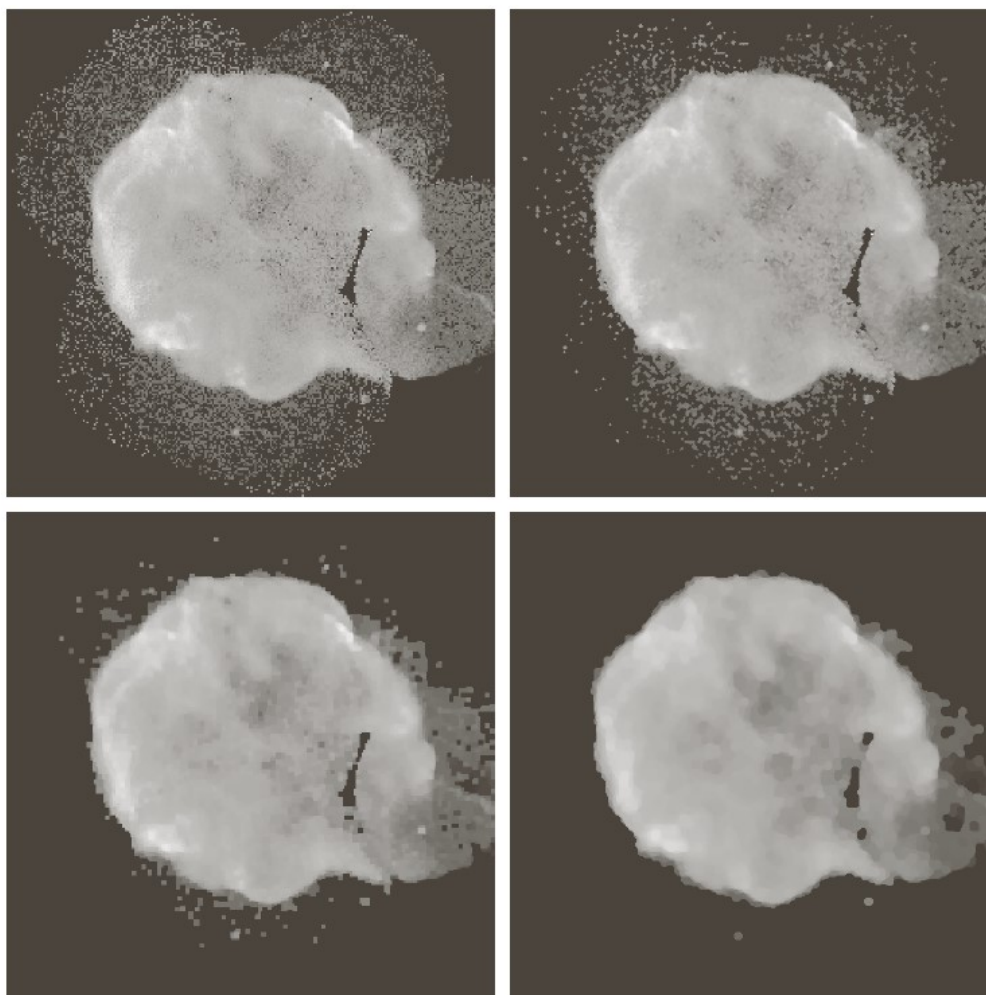
扩展至灰度图像



a	b	c
d	e	f

FIGURE 9.41 (a) 531×675 image of wood dowels. (b) Smoothed image. (c)–(f) Openings of (b) with disks of radii equal to 10, 20, 25, and 30 pixels, respectively. (Original image courtesy of Dr. Steve Eddins, The MathWorks, Inc.)

扩展至灰度图像



a b
c d

FIGURE 9.38

(a) 566×566 image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of performing opening and closing sequences on the original image with disk structuring elements of radii, 1, 3, and 5, respectively. (Original image courtesy of NASA.)



扩展至灰度图像

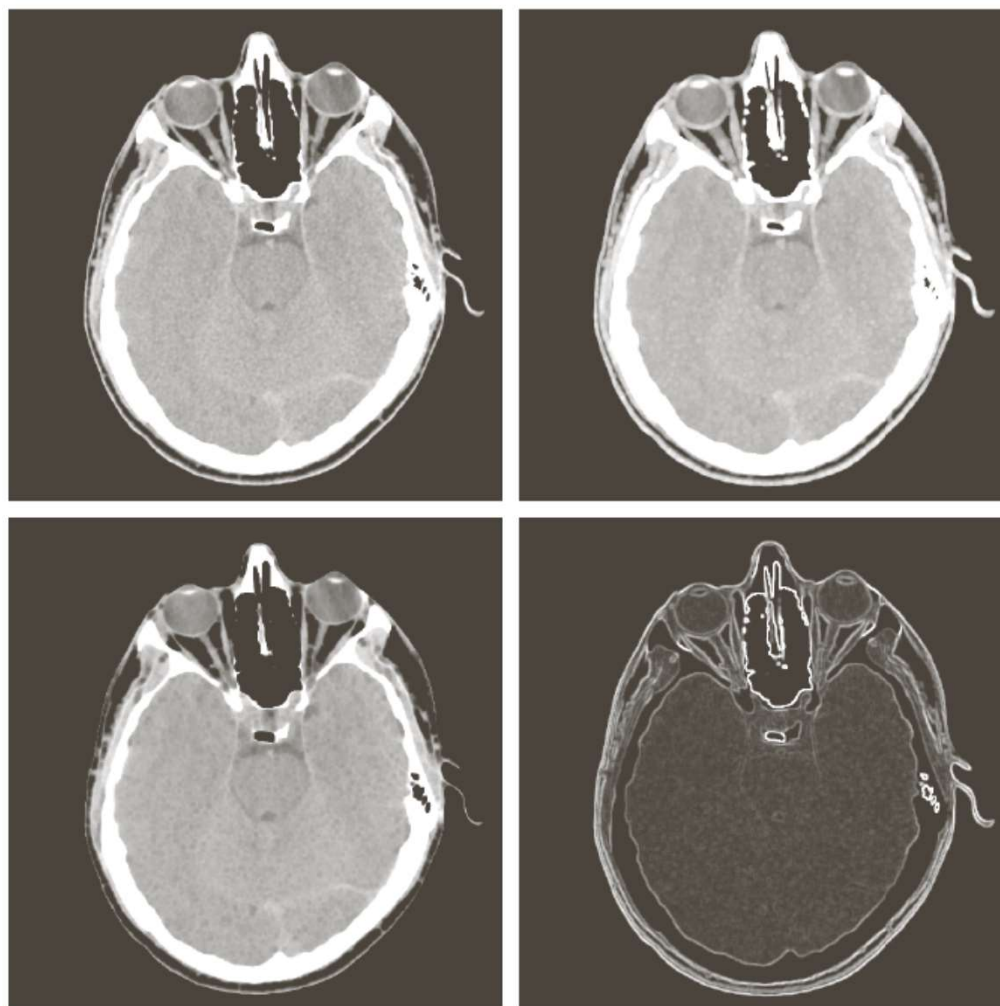
- 形态学梯度

$$g = (f \oplus b) - (f \ominus b)$$

- Top-hat变换

$$h = f - (f \circ b)$$

扩展至灰度图像



a	b
c	d

FIGURE 9.39

(a) 512×512 image of a head CT scan.
(b) Dilation.
(c) Erosion.
(d) Morphological gradient, computed as the difference between (b) and (c).
(Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

扩展至灰度图像

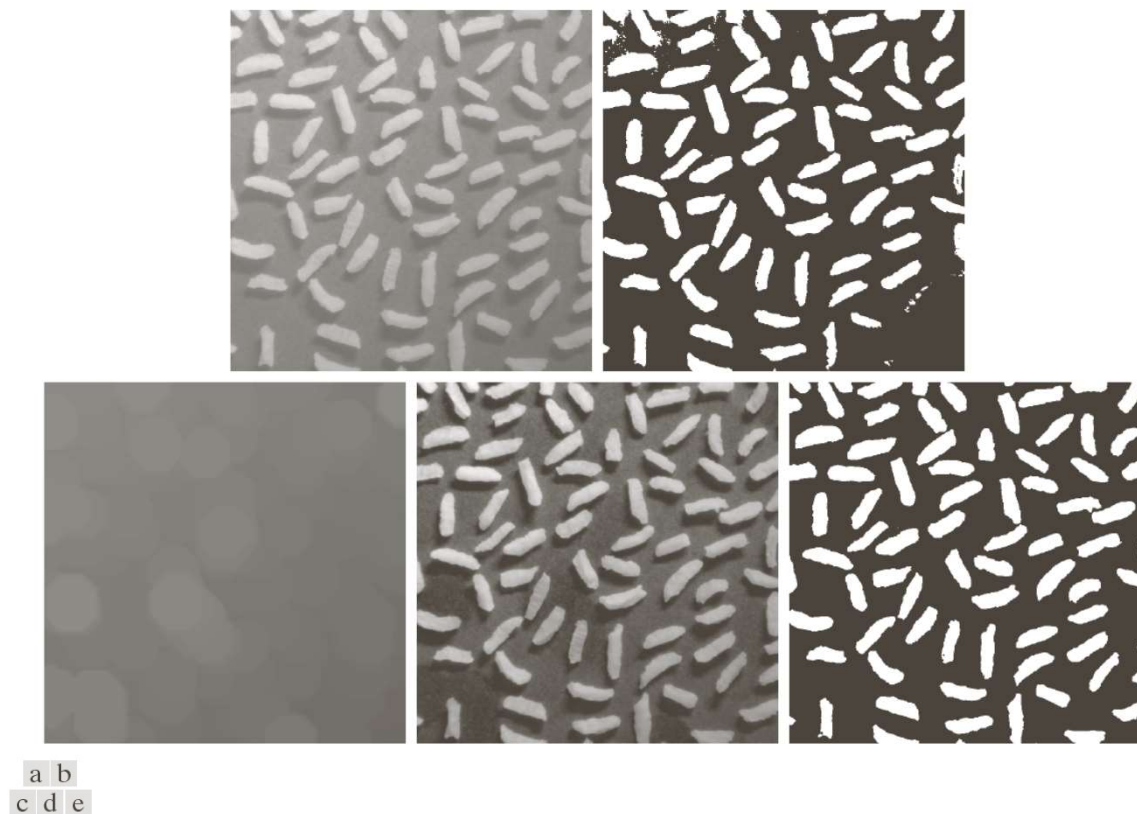


FIGURE 9.40 Using the top-hat transformation for *shading correction*. (a) Original image of size 600×600 pixels. (b) Thresholded image. (c) Image opened using a disk SE of radius 40. (d) Top-hat transformation (the image minus its opening). (e) Thresholded top-hat image.