



华北电力大学
NORTH CHINA ELECTRIC POWER UNIVERSITY

6 形态学图像处理



主要内容

- 形态学相关概念
 - ✓ 数学形态学的发展历史
 - ✓ 数学形态学基本思想
- 形态学基本运算
 - ✓ 集合论中的基本概念
 - ✓ 二值图像中的基本逻辑操作



主要内容

- 膨胀与腐蚀
 - ✓ 膨胀
 - ✓ 腐蚀
- 开操作与闭操作
 - ✓ 开操作
 - ✓ 闭操作
- 形态学的主要应用
 - ✓ 边界提取
 - ✓ 区域填充
 - ✓ 梯度



6.1 形态学相关概念

- 形态学 (morphology)
 - ✓ 形态学是生物学的一个分支，是一门用于研究动物和植物的形态和结构的学科。
- 数学形态学 (mathematical morphology)
 - ✓ 数学形态学是以形态学为基础对图像进行分析的数学工具，并已在图像分析、计算机视觉、模式识别和信号处理等方面得到了广泛的应用。



6.1.1 数学形态学发展历史

- 1964年，法国巴黎矿业学院马瑟荣（Matheron）和其学生塞拉（Serra）对铁矿核的定量岩石学分析，从而提出该理论。
- 1982年，Serra出版了《Image Analysis and Mathematical Morphology》，将数学形态学真正用于图像处理和分析。
- 90年代至今，在模式识别，编码，运动分析，运动景物描述、放射医学、工业控制等方面取得进展，及用于数值函数的形态学算子开发等。



6.1.2 数学形态学思想

- 用具有一定形态的结构元素去度量和提取图像中的对应形状以达到对图像分析和识别的目的。
- 形态学图像处理表现为一种邻域运算形式。
- 一种特殊定义的邻域称之为“结构元素”在每个像素位置上它与二值图像对应的区域进行特定的逻辑运算，逻辑运算的结果为输出图像的相应像素。
- 形态学运算的效果取决于结构元素的大小、内容以及逻辑运算的性质



6.1.2 数学形态学思想

- 用途
 - ✓ 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，可以实现噪声滤除、特征提取、边缘检测、图像分割、形状识别、纹理分析等
- 基本的运算包括
 - ✓ 腐蚀和膨胀
 - ✓ 开、闭运算



6.2 形态学基本运算

- 数学形态学的语言是集合论
 - ✓ 集合表示图像中的不同对象，比如在二值图像中，所有的黑色像素构成一个集合，用来表示目标（背景）。
 - ✓ 二值图像中，集合是二维整数空间 (Z^2) 的元素，即集合中的每一个元素都是一个二维向量，是图像中某一像素的空间坐标(x,y)。



6.2.1 集合论中的基本概念

- 属于、不属于、空集

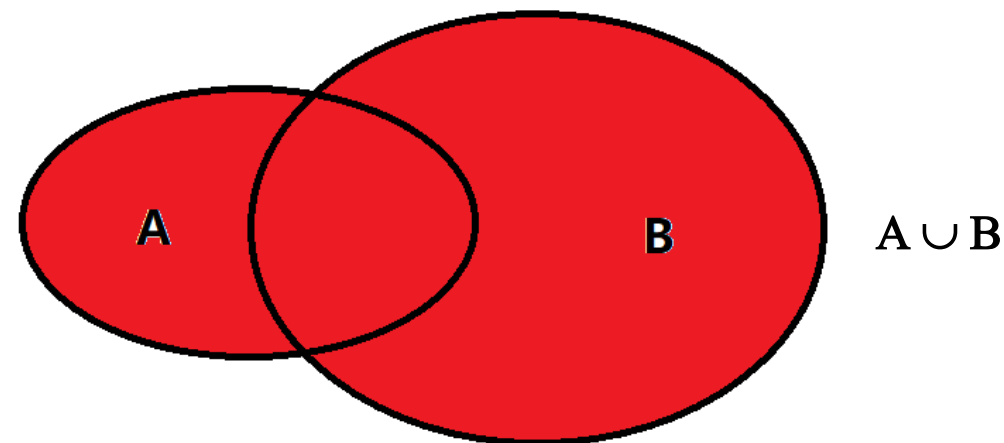
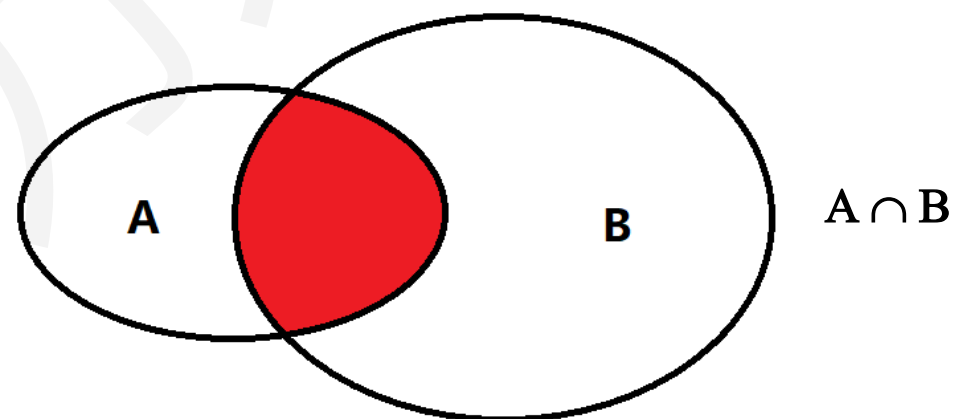
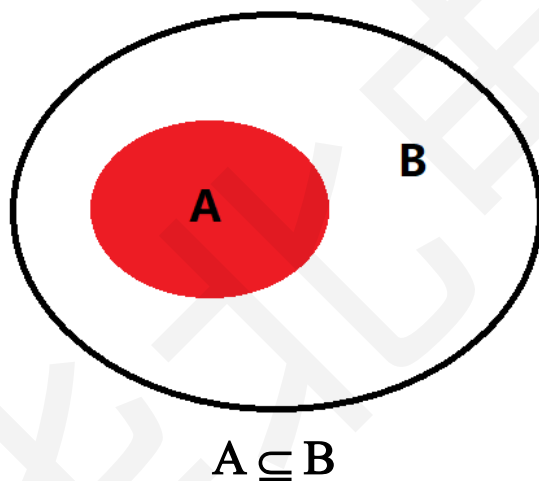
✓ 令 A 是 Z^2 中的一个集合，如果 α 是其中的一个元素，称 α 属于 A ，并记作： $\alpha \in A$ ，否则，称 α 不属于 A ，记为： $\alpha \notin A$ ，如 A 中没有任何元素，称 A 为空集： \emptyset



6.2.1 集合论中的基本概念

- 子集、并集、交集

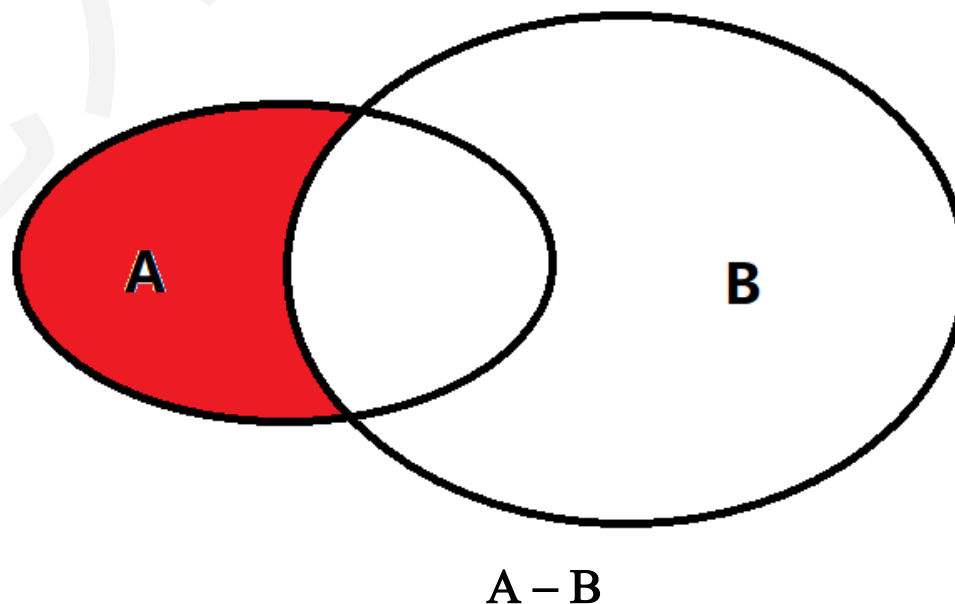
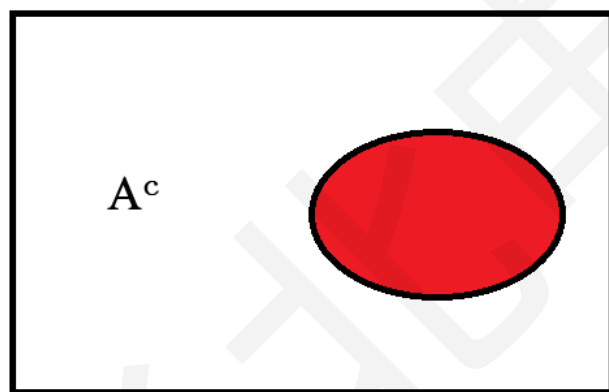
✓ $A \subseteq B$, $X = A \cup B$, $X = A \cap B$



6.2.1 集合论中的基本概念

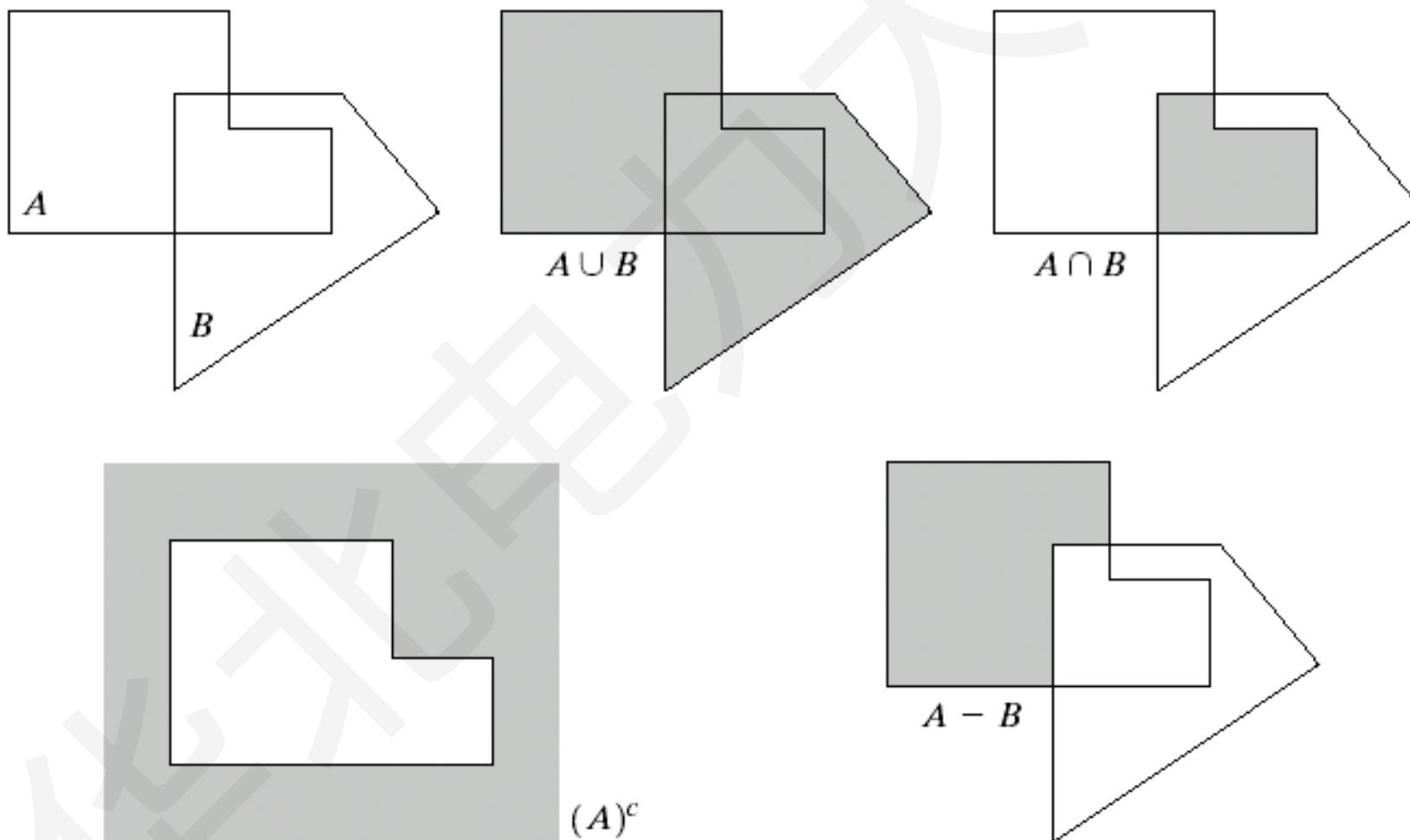
- 不相连（互斥）、补集、差集

✓ $A \cap B = \emptyset$, $A^c = \{ \alpha \mid \alpha \notin A \}$, $A - B = \{ \alpha \mid \alpha \in A, \alpha \notin B \} = A \cap B^c$





6.2.1 集合论中的基本概念





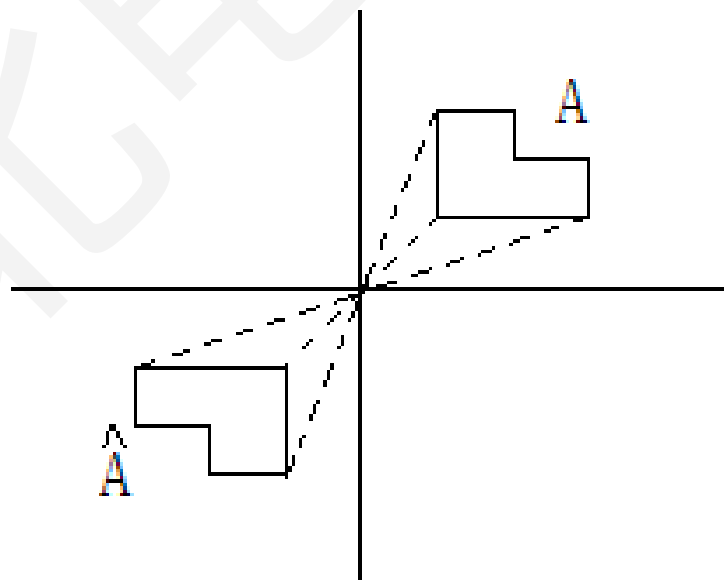
6.2.1 集合论中的基本概念

- 集合的反射

- ✓ 由集合A中所有元素相对于原点的反射元素组成的集合称为集合A的反射，记为 \hat{A}

- ✓ 相对某个中心点

$$\hat{A} = \{x \mid x = -a, a \in A\}$$





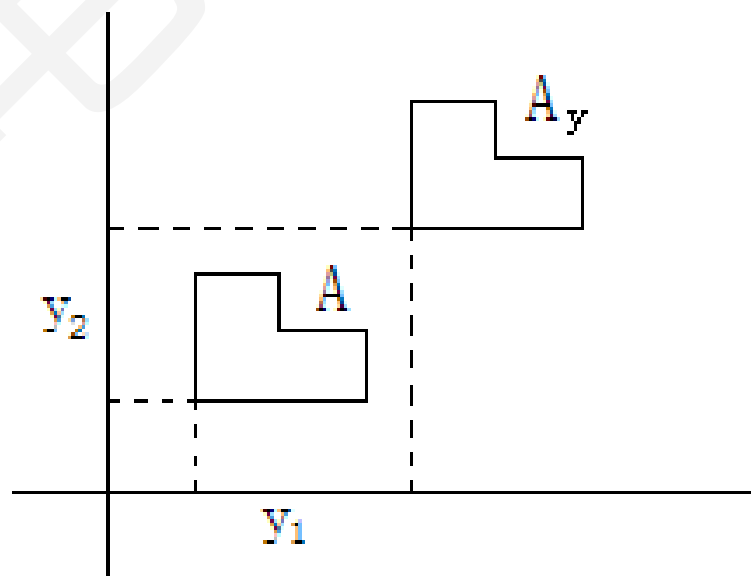
6.2.1 集合论中的基本概念

- 集合的平移

✓ 由集合A中所有元素平移 $y=(y_1, y_2)$ 后组成的元素集合称为集合A的平移，记为

$$(A)_y = \{x \mid x = a + y, a \in A\}$$

✓ 相对原点

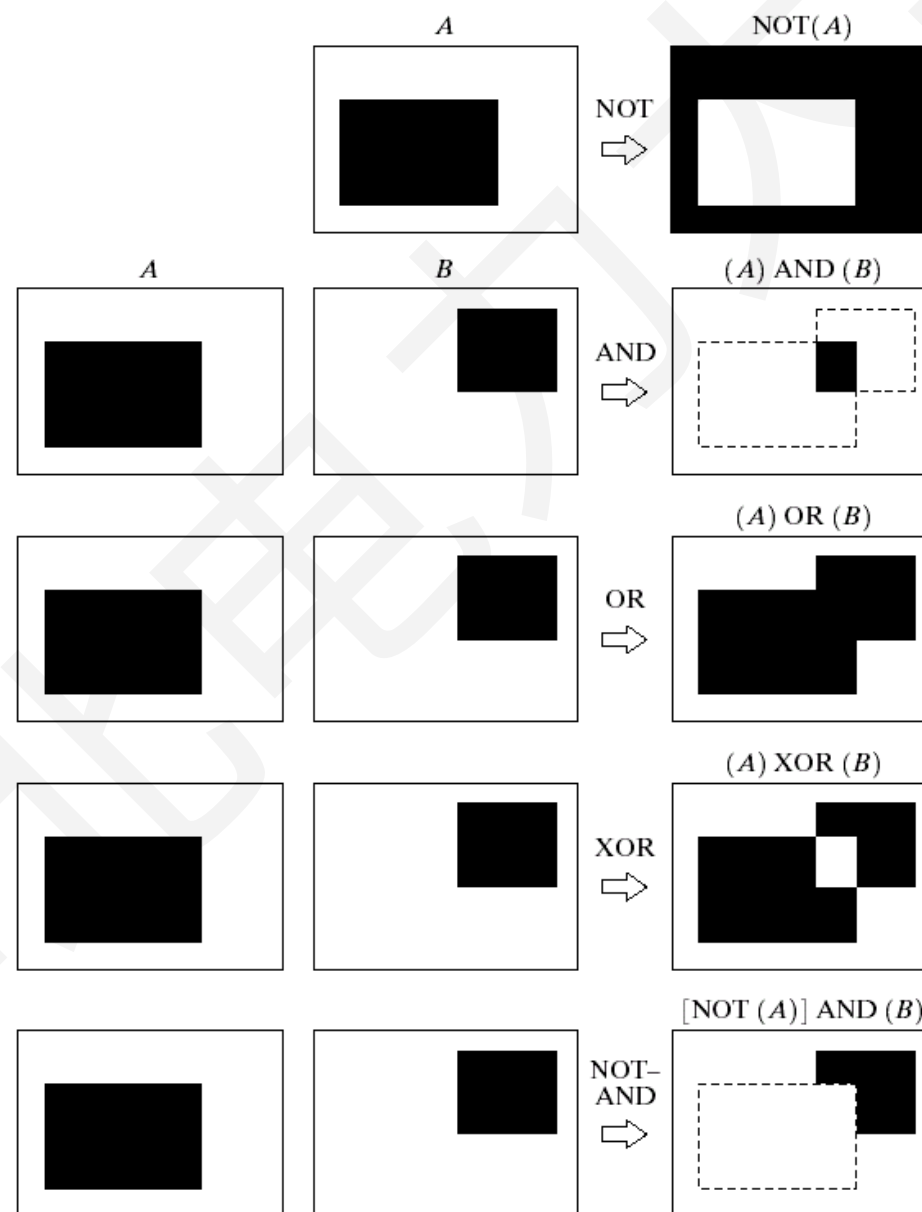




6.2.2 二值图像中的基本逻辑操作

- 最基本的逻辑运算：与、或、非、异或
 - ✓ 由尽管逻辑操作与集合操作间存在一一对应的关系，但逻辑操作只是针对二值图像

6.2.2 二值图像中的基本逻辑操作





6.3 膨胀与腐蚀

- 膨胀和腐蚀是数学形态学算法的基础。
 - ✓ 在二值形态学中，被处理的二值图像称为**目标图像**，一般用集合**A**表示。
 - ✓ 用数学形态学方法在对图像进行处理时，需要有一个**结构元素**，用集合**B**来表示。
 - ✓ 结构元素B的尺寸**明显小于**图像的尺寸，是比较小的图像像素的集合。
 - ✓ 二值形态学运算过程就是在图像中移动结构元素，将结构元素与其下面重叠部分的图像进行交、并等集合运算。



6.3.1 膨胀

- 膨胀 (dilation) : 使图像变大, 目标图像A被结构元素B膨胀可定义为:

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

- ✓ 先对结构元素做关于原点的映射, 再将其映射平移 z 。膨胀的结果是所有 z 的集合, 平移 z 后, B的映射与A至少有一个元素是重叠的。
- ✓ 膨胀后的结果仍是一个像素坐标点的集合。



6.3.1 膨胀

1 原点

- 操作过程

✓ 每当结构元素 \hat{B} 在目标图像A上平移后，结构元素 \hat{B} 与其覆盖的子图像中至少有一个元素相交时，就将目标图像中与结构元素 \hat{B} 的原点对应的那个位置的像素值置为“1”，否则置为0。

			1		1
	1			1	
	1	1			
		1	1		
		1	1		

目标图像

		1			
		1	1		

结构元素

	1	1			
		1			

\hat{B}

6.3.1 膨胀


			1		1
	1			1	
	1	1			
		1	1		
		1	1		

	1	1			
		1			

	1		1		1
	1			1	
	1	1			
		1	1		
		1	1		

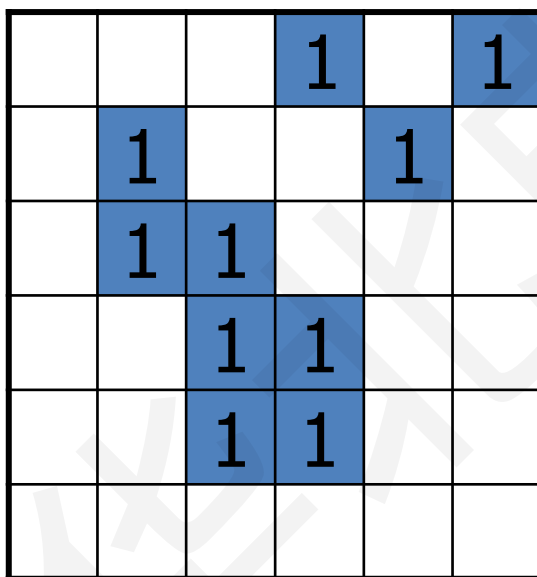
	1	?	1		1
	1			1	
	1	1			
		1	1		
		1	1		

	1		1	1	1
	1	1		1	1
	1	1	1		
		1	1	1	
		1	1	1	

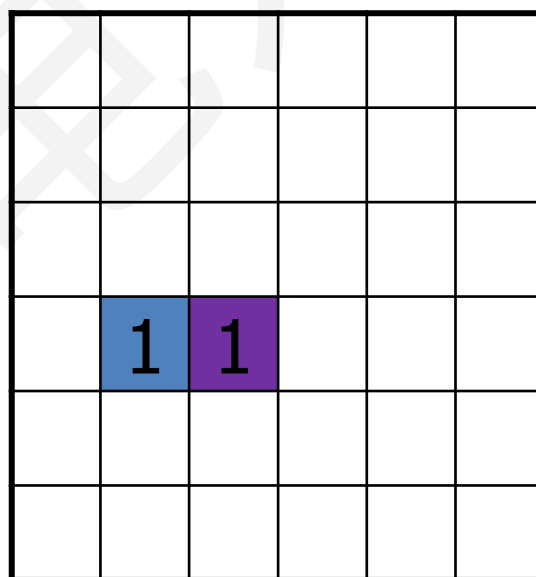
 膨胀出来的元素

6.3.1 膨胀

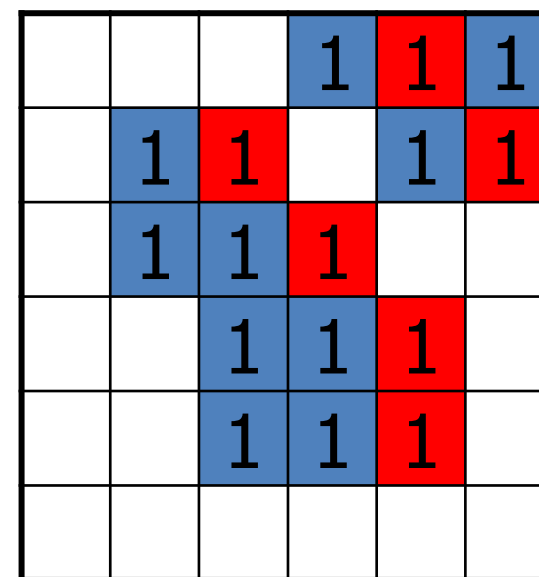
- 结构元素形状对膨胀运算结果的影响
 - ✓ 当目标图像不变，但所给的结构元素的形状改变时，膨胀运算的结果会发生改变。



目标图像

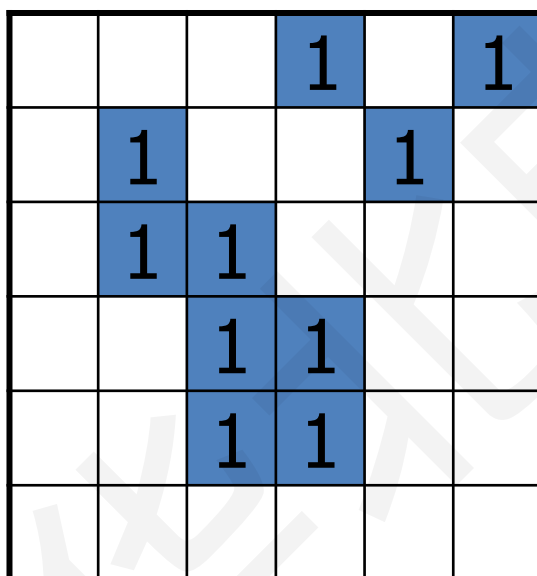


\hat{B}

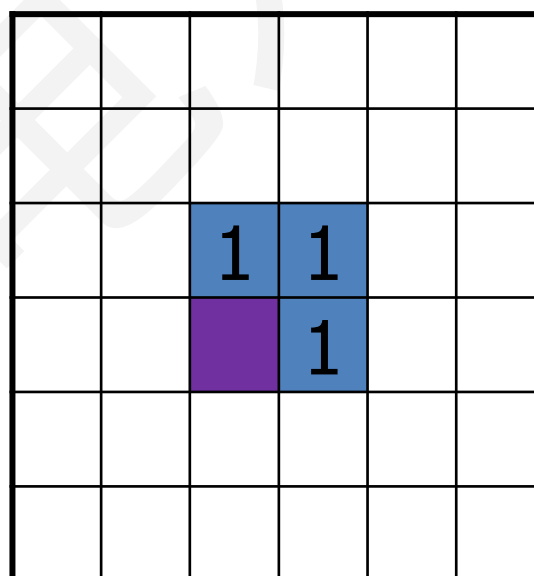


6.3.1 膨胀

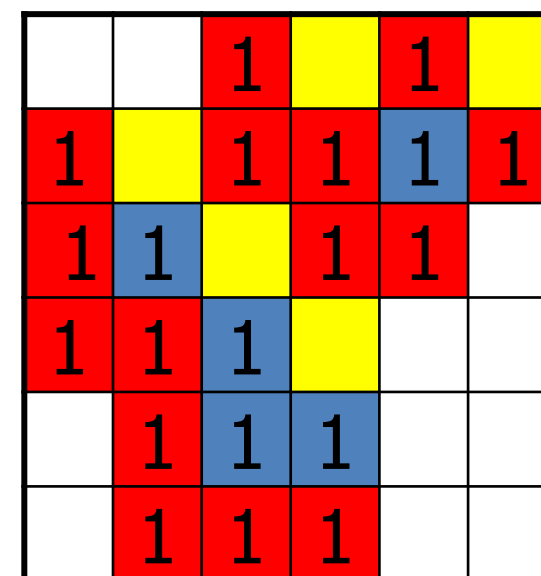
- 结构元素形状对膨胀运算结果的影响
 - ✓ 当目标图像不变，结构元素的形状不变，而其原点位置改变时，膨胀运算的结果会发生改变。



目标图像

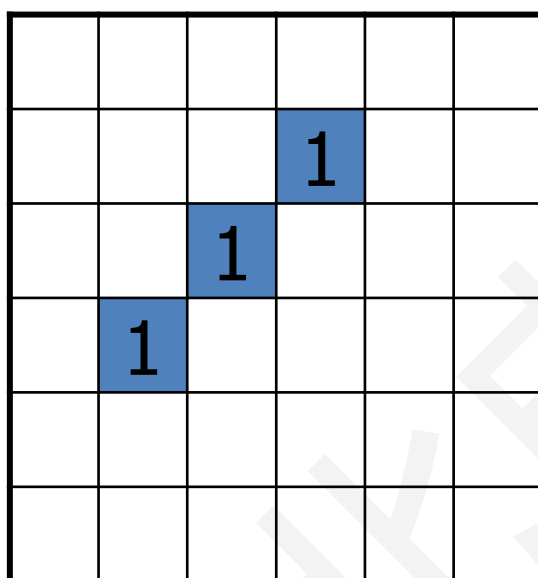


\hat{B}

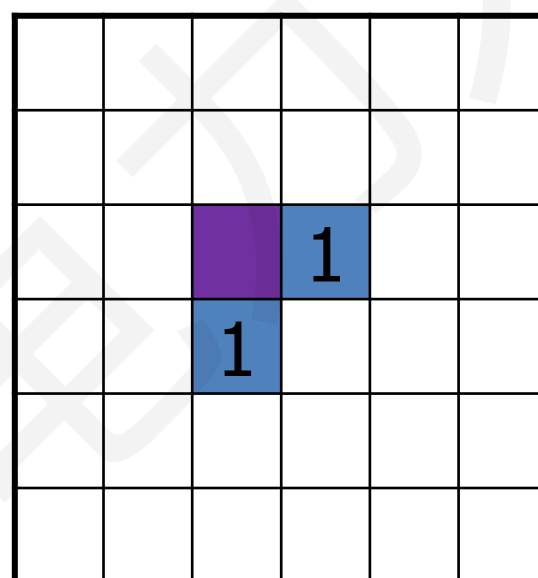


目标图像中消失的元素

6.3.1 膨胀

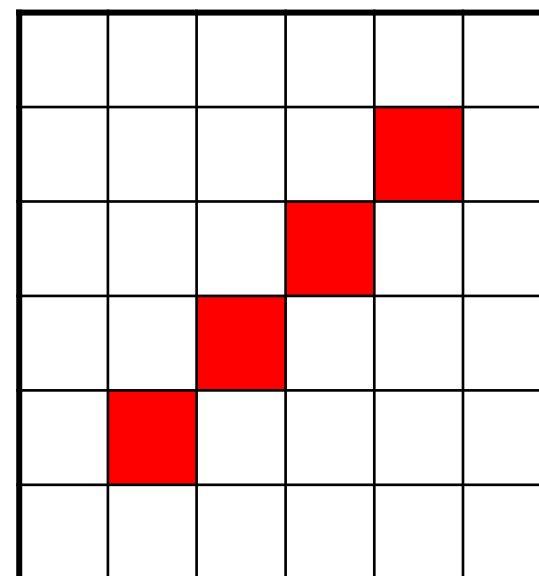
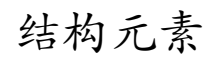


目标图像

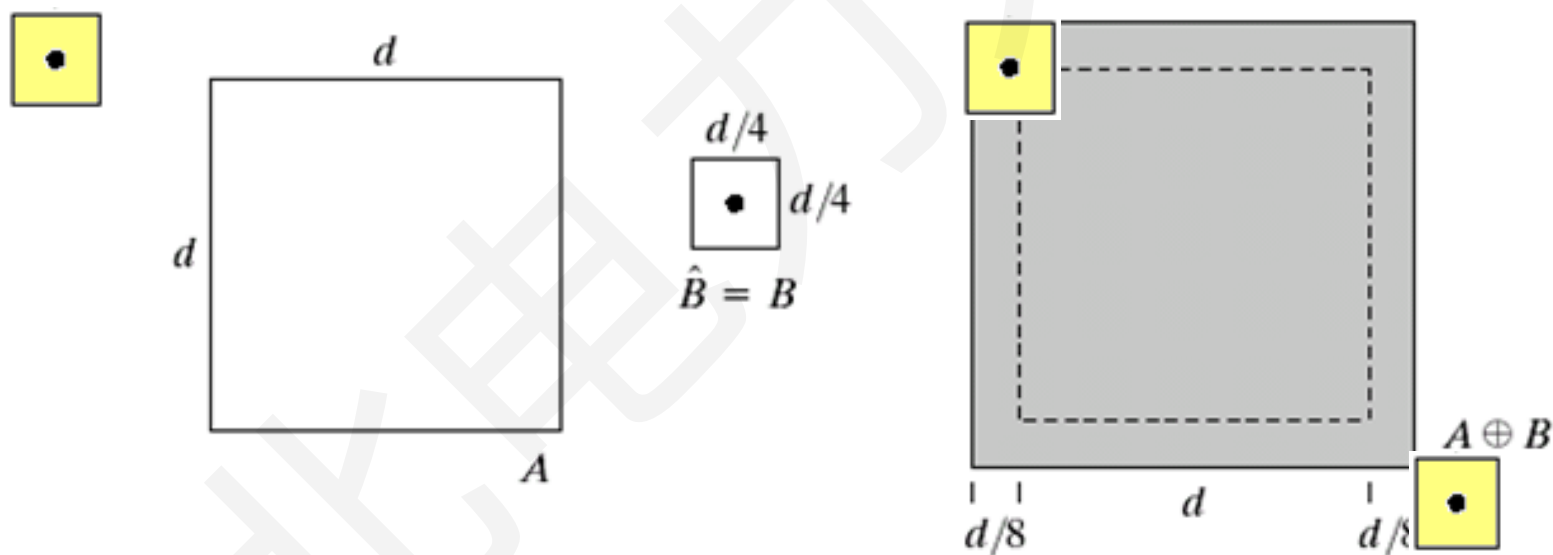


结构元素

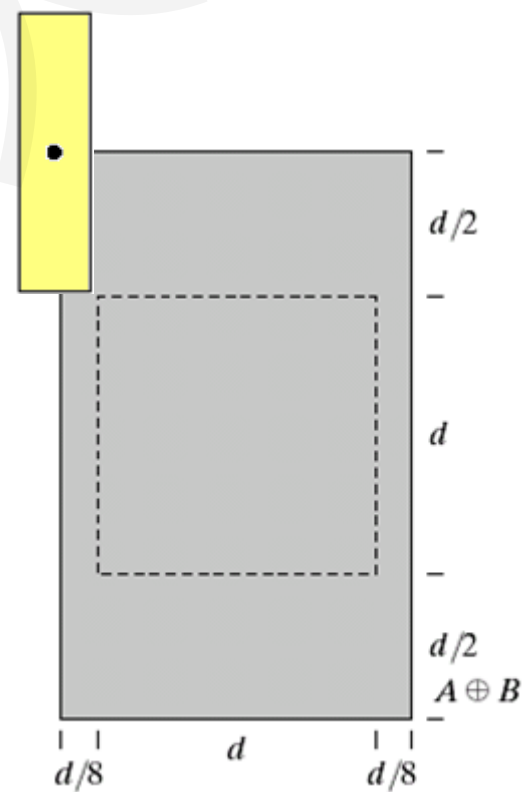
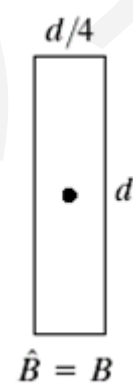
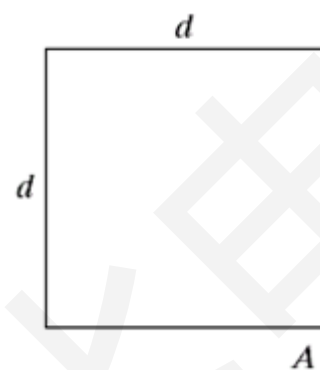
?



6.3.1 膨胀



6.3.1 膨胀

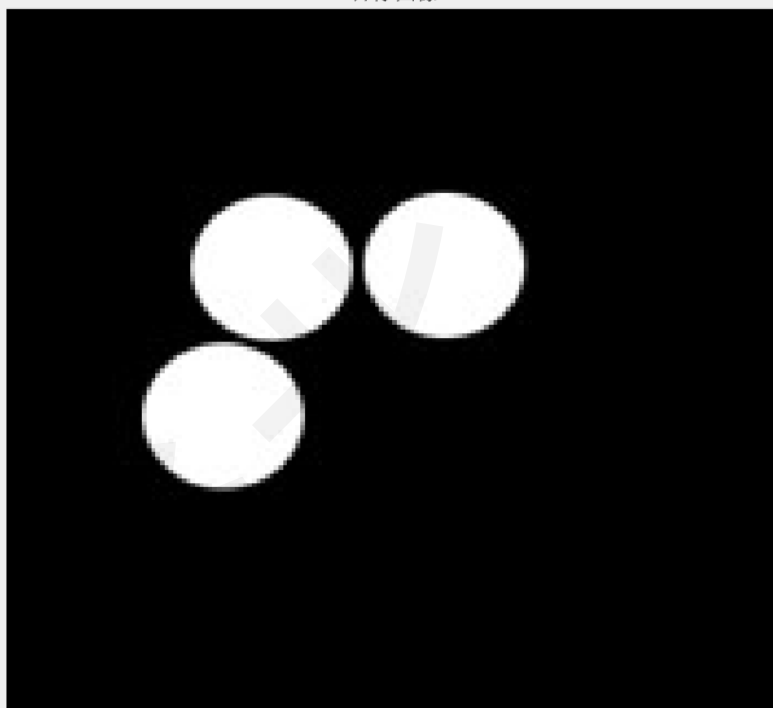


6.3.1 膨胀

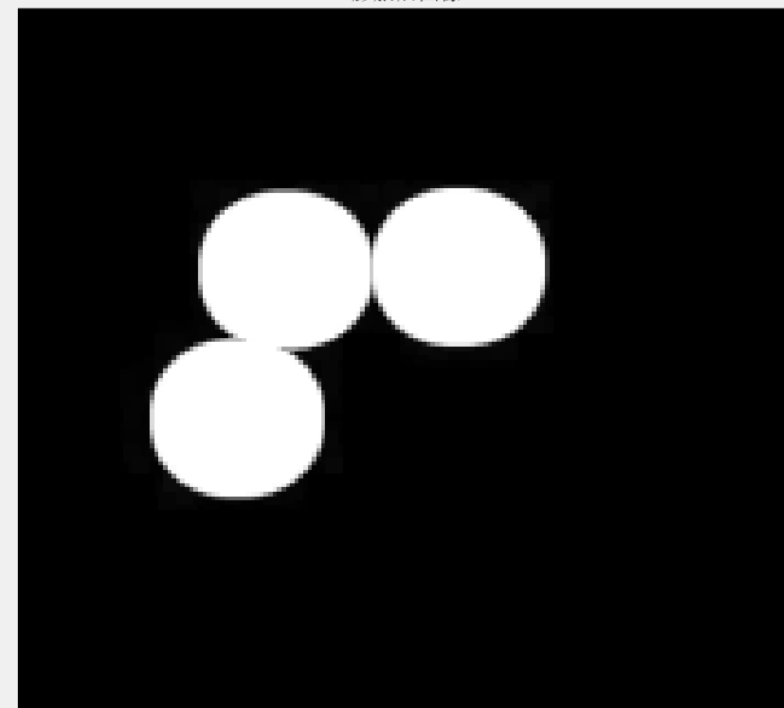
- 膨胀的应用

✓ 填充图像中相对于结构元素较小的小孔，连接相邻的物体，同时它对图像有扩张作用。

目标图像



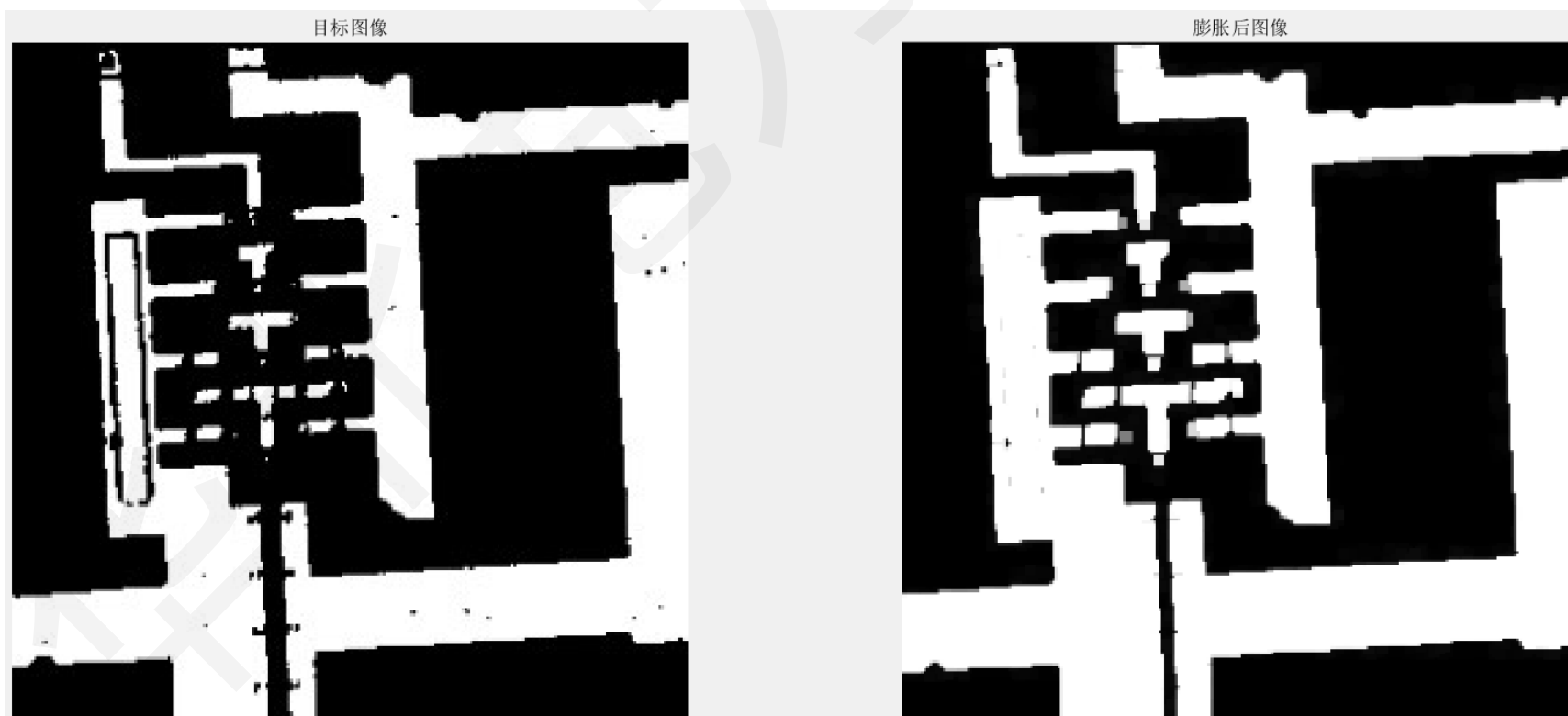
膨胀后图像



6.3.1 膨胀

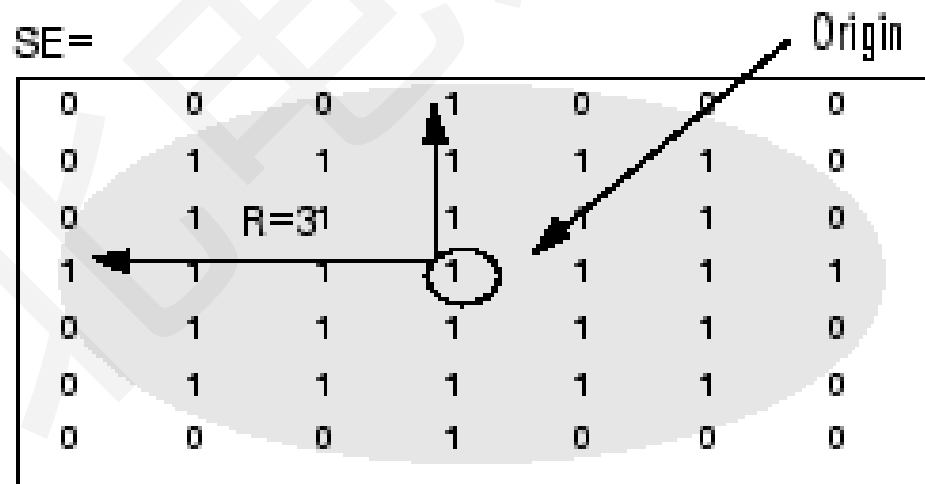
- 膨胀的应用

✓ 填充图像中相对于结构元素较小的小孔，连接相邻的物体，同时它对图像有扩张作用。



6.3.2 matlab对膨胀操作的支持

- 膨胀: $Im = \text{imdilate}(I, se, n)$
 - ✓ I 是目标图像, se 是结构元素, n 膨胀次数。
- 结构元素定义: $se = \text{strel}(\text{shape}, \text{parameters})$
 - ✓ shape 是结构元素的形状, parameters 具体参数。





6.3.2 matlab对膨胀操作的支持

- ✓ `se = strel("line",len,deg)` 创建一个关于邻域中心对称的线性结构元素，长度约为 `len`，角度约为 `deg`。
- ✓ `se = strel("rectangle",[m n])` 创建一个大小为 `[m n]` 的矩形结构元素。
- ✓ `se = strel("square",w)` 创建一个宽度为 `w` 个像素的正方形结构元素。
- ✓ `se = strel("diamond",r)` 创建一个菱形结构元素，其中 `r` 指定从结构元素原点到菱形各点的距离。



6.3.2 matlab对膨胀操作的支持

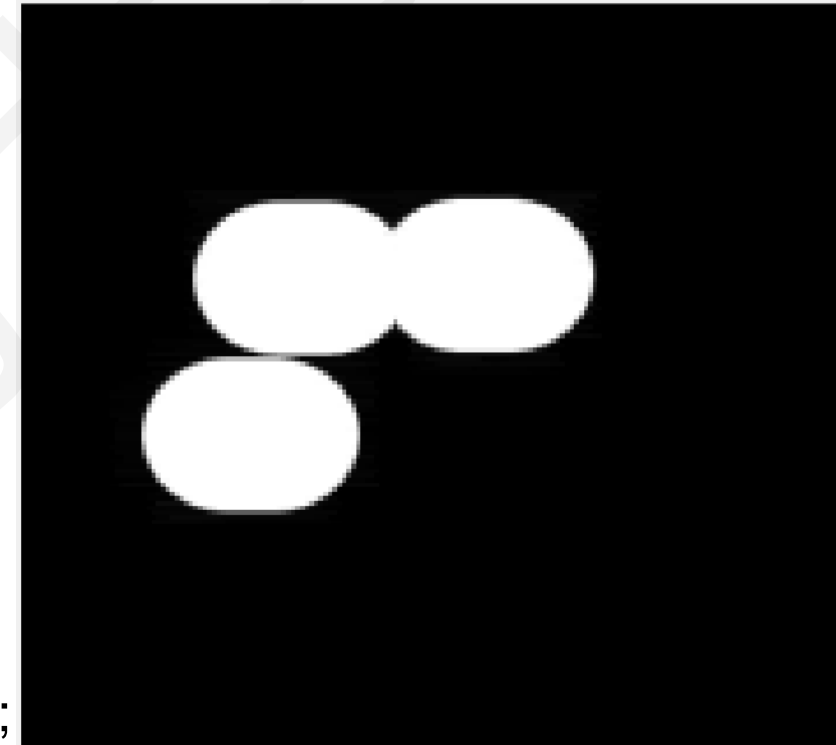
```
I1 = imread('3circles.jpg');  
I1 = rgb2gray(I1);  
I1 = im2double(I1);  
  
se = strel('square',4); %结构元素  
I2 = imdilate(I1, se); %膨胀  
figure(1);  
subplot(1,2,1);imshow(I1);title('目标图像');  
subplot(1,2,2);imshow(I2);title('膨胀后图像');
```



6.3.2 matlab对膨胀操作的支持

```
I1 = imread('3circles.jpg');  
I1 = rgb2gray(I1);  
I1 = im2double(I1);  
  
se = strel('line',10,0); %结构元素  
I2 = imdilate(I1, se); %膨胀  
figure(1);  
subplot(1,2,1);imshow(I1);title('目标图像');  
subplot(1,2,2);imshow(I2);title('膨胀后图像');
```

膨胀后图像





6.3.3 腐蚀

- 腐蚀 (erosion) : 使图像变小, 目标图像A被结构元素B腐蚀可定义为:

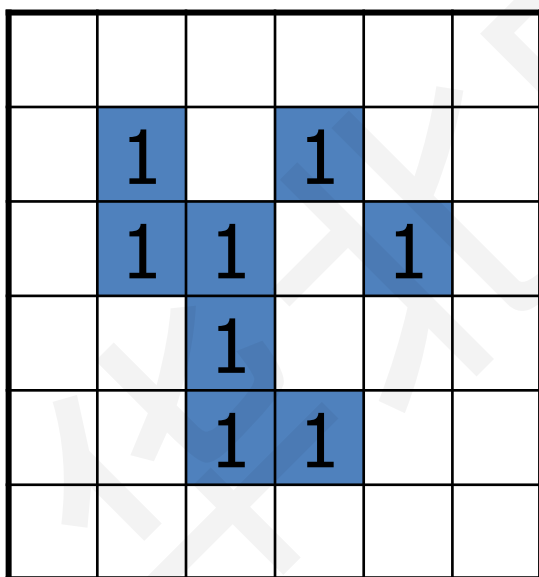
$$A \ominus B = \{x \mid (B)_x \subseteq A\}$$

- ✓ 用B来腐蚀A得到的集合是B完全包含在A中时B的原点位置的集合。
- ✓ 就是在目标图像中标出那些与结构元素相同的子图像的原点位置的像素。

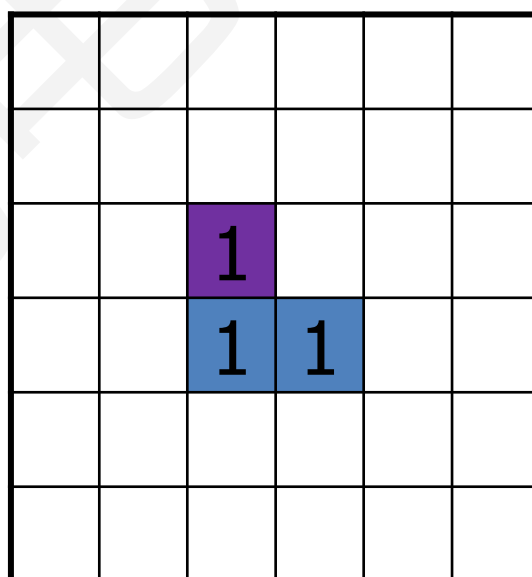
6.3.3 腐蚀

• 操作过程

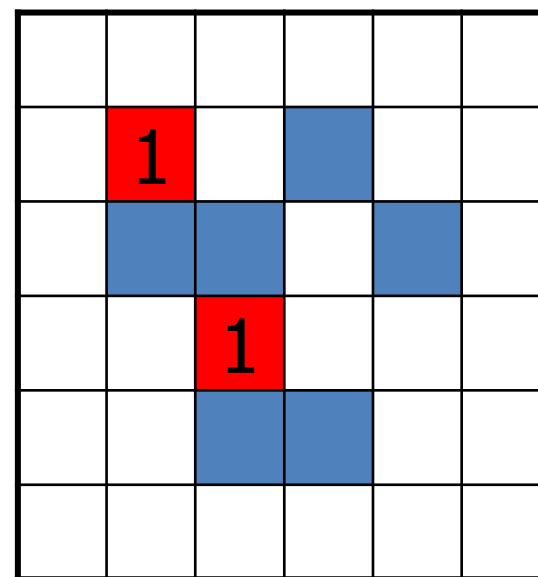
✓ 每当在目标图像A中找到一个与结构元素B相同的子图像时，就把该子图像中与B的原点位置对应的那个像素位置标注为1，图像A上标注出的所有这样的像素组成的集合，即为腐蚀运算的结果。



目标图像

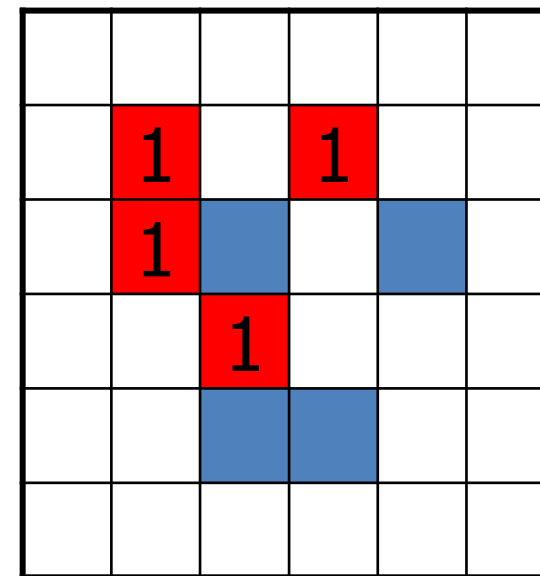
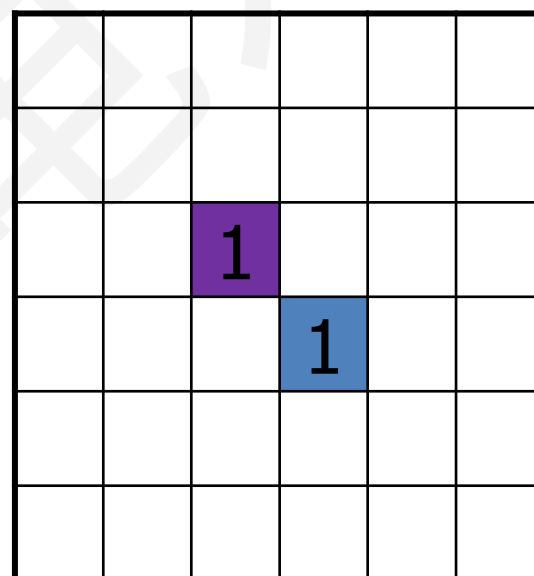
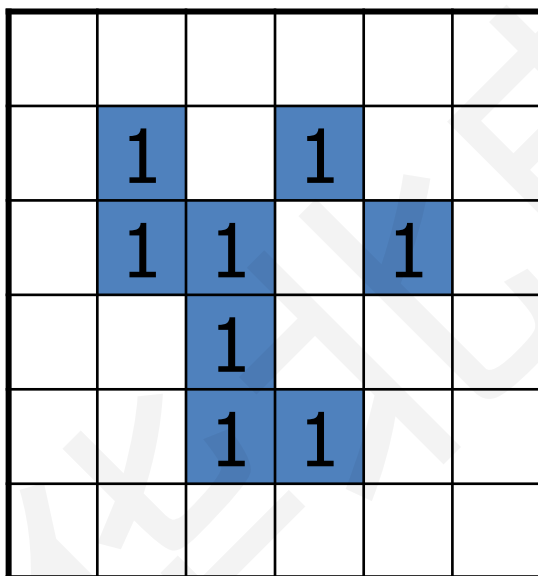


结构元素



6.3.3 腐蚀

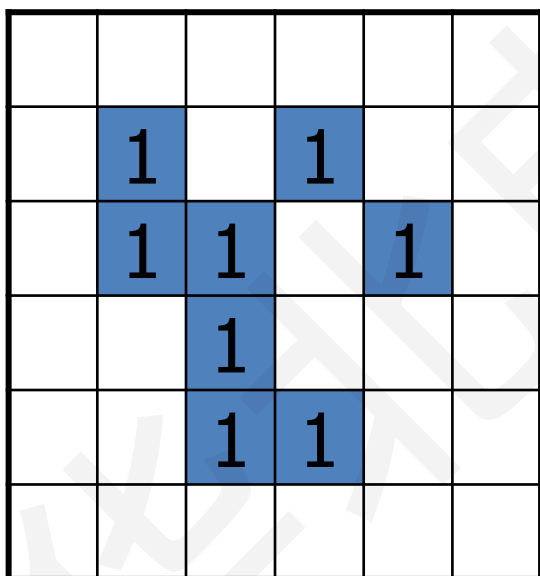
- 结构元素形状对腐蚀运算结果的影响
 - ✓ 当目标图像不变，但所给的结构元素的形状改变时，腐蚀运算的结果会发生改变。



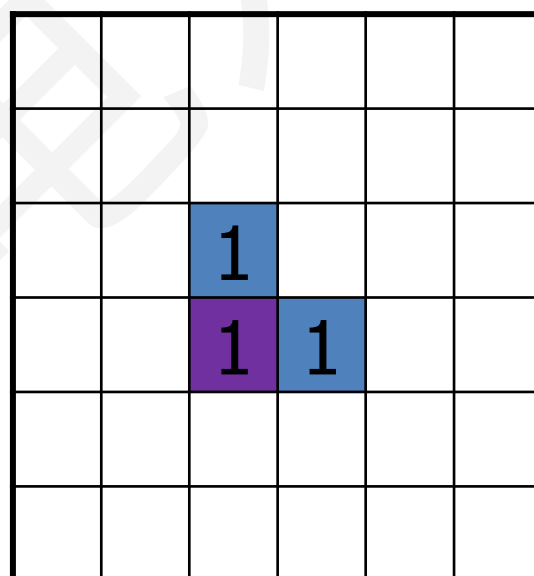
目标图像

6.3.3 腐蚀

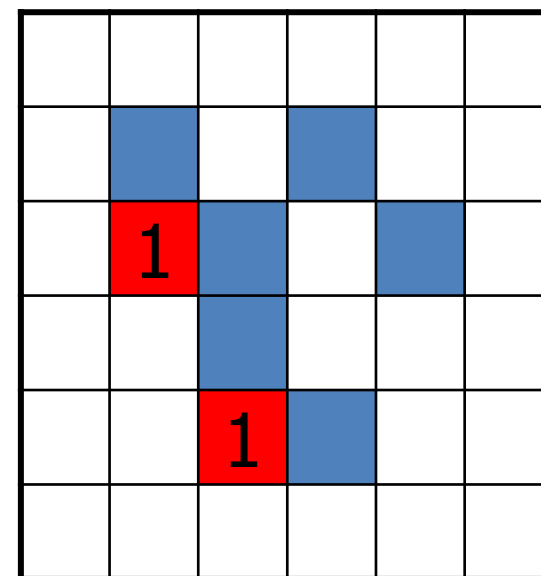
- 结构元素形状对腐蚀运算结果的影响
 - ✓ 当目标图像不变，结构元素的形状不变，而其原点位置改变时，腐蚀运算的结果会发生改变。



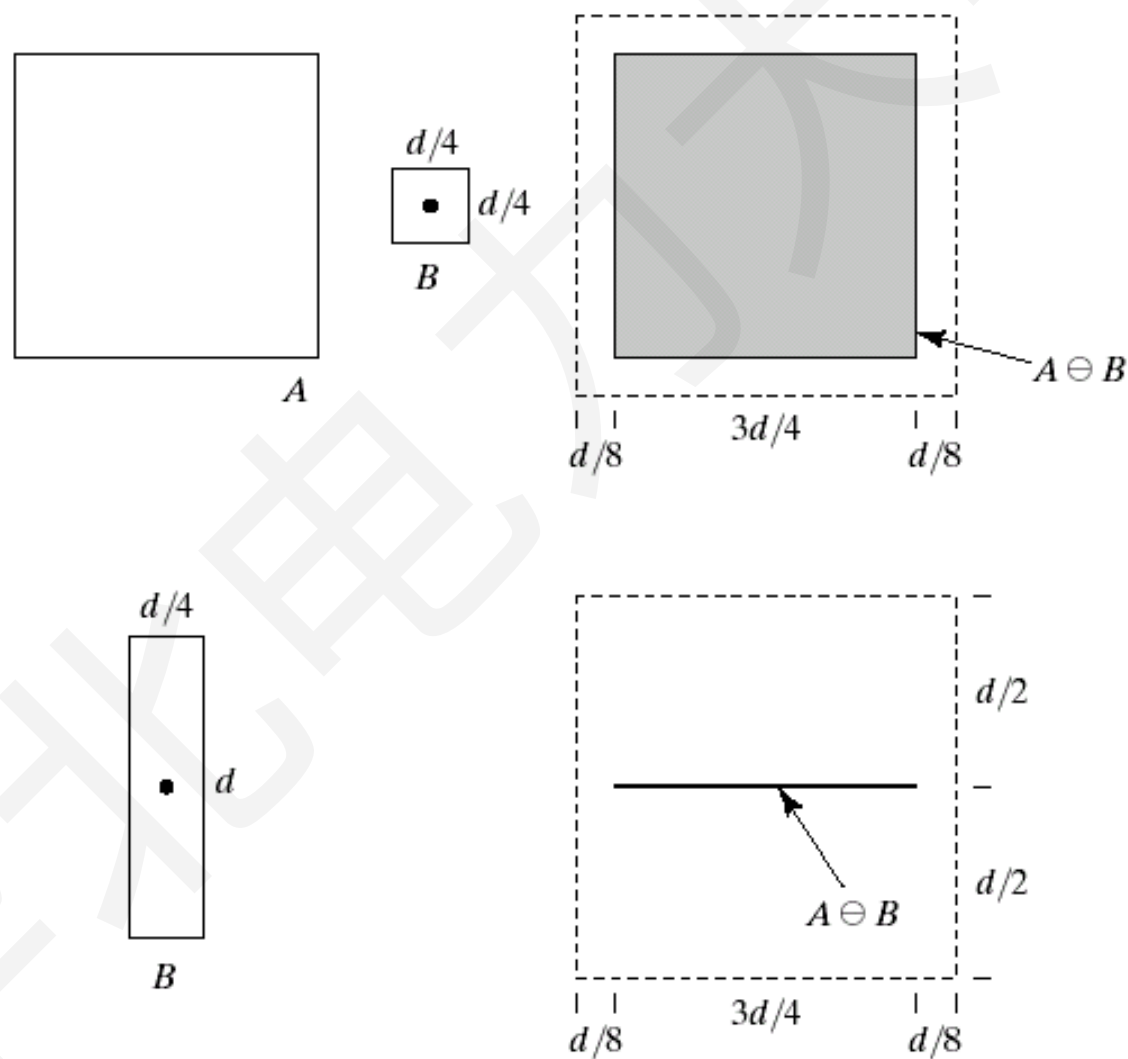
目标图像



结构元素



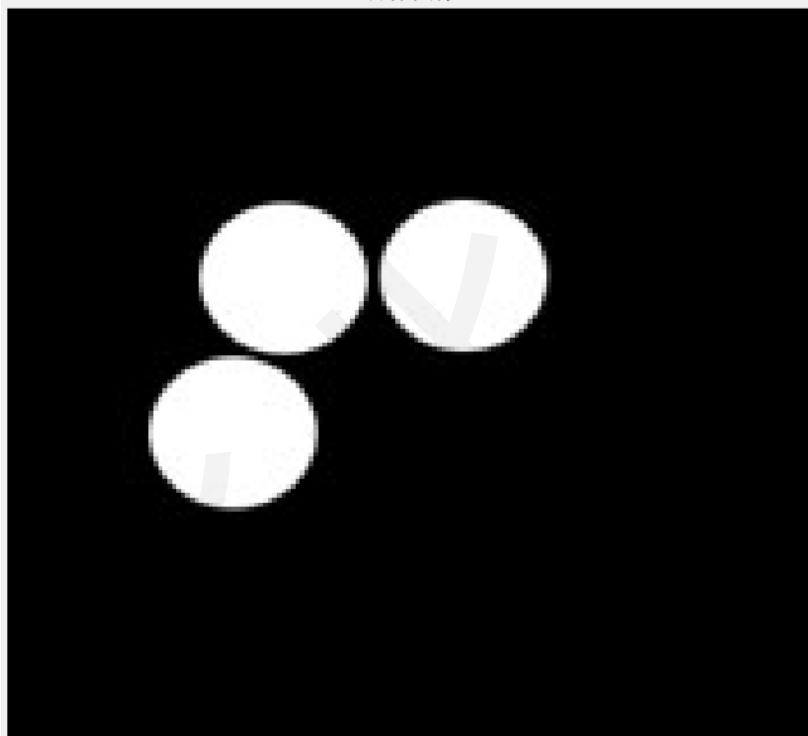
6.3.3 腐蚀



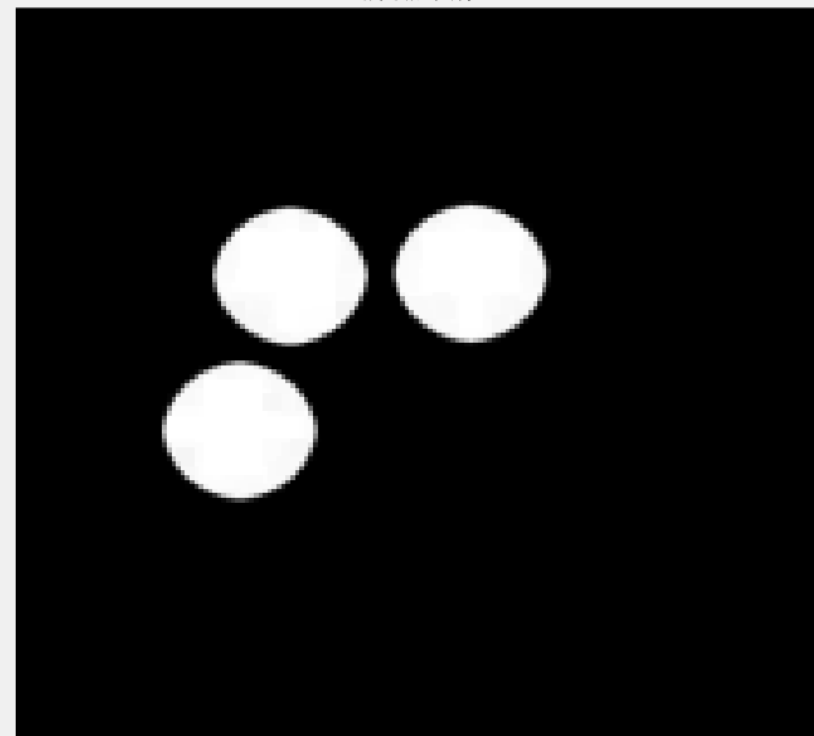
6.3.3 腐蚀

- 腐蚀的应用
 - ✓ 消除图像中比结构元素小的成分，去除物体之间的粘连，消除图像中的小颗粒噪声。

目标图像



腐蚀后图像

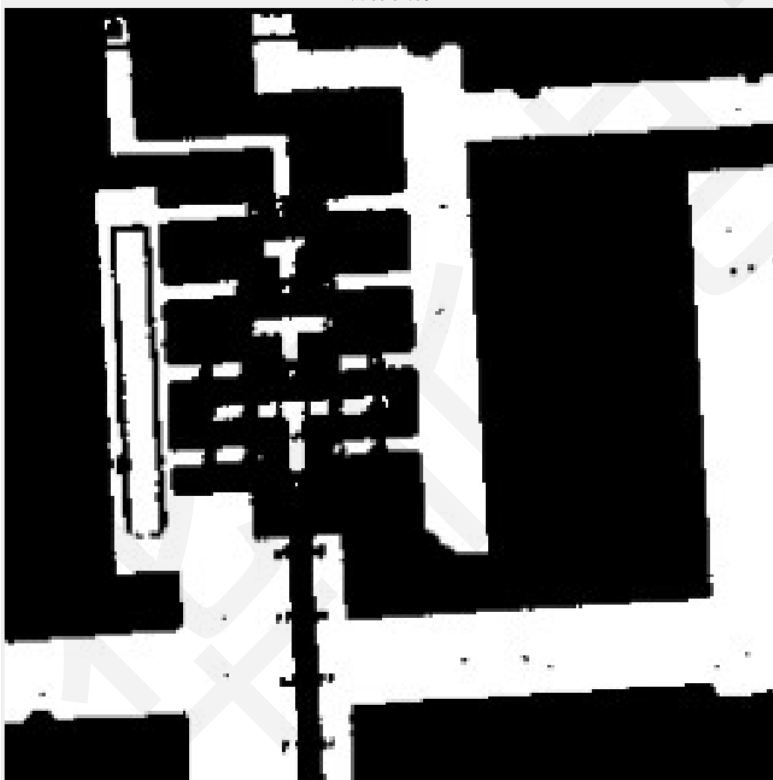


6.3.3 腐蚀

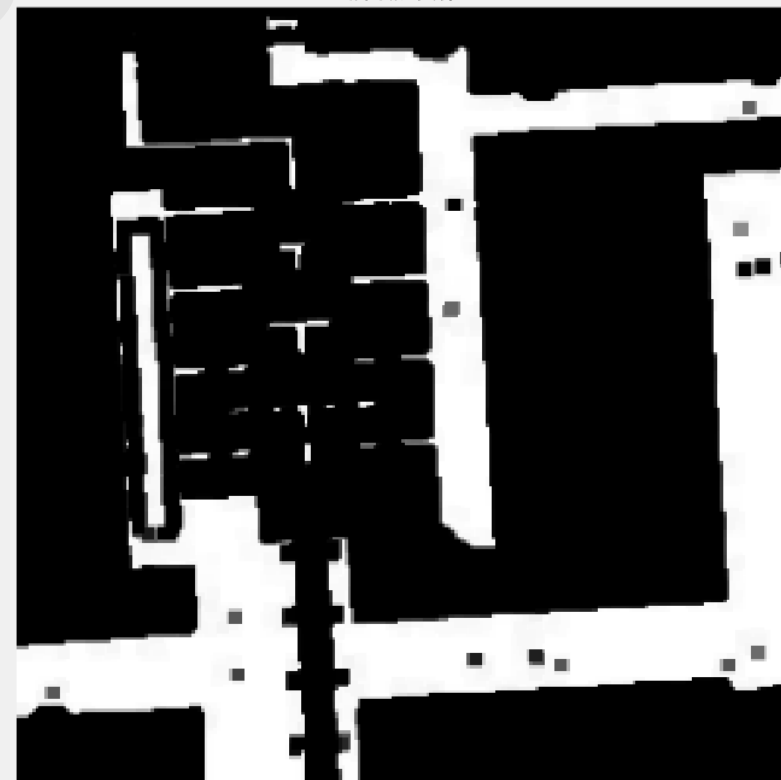
- 腐蚀的应用

✓ 消除图像中比结构元素小的成分，去除物体之间的粘连，消除图像中的小颗粒噪声。

目标图像

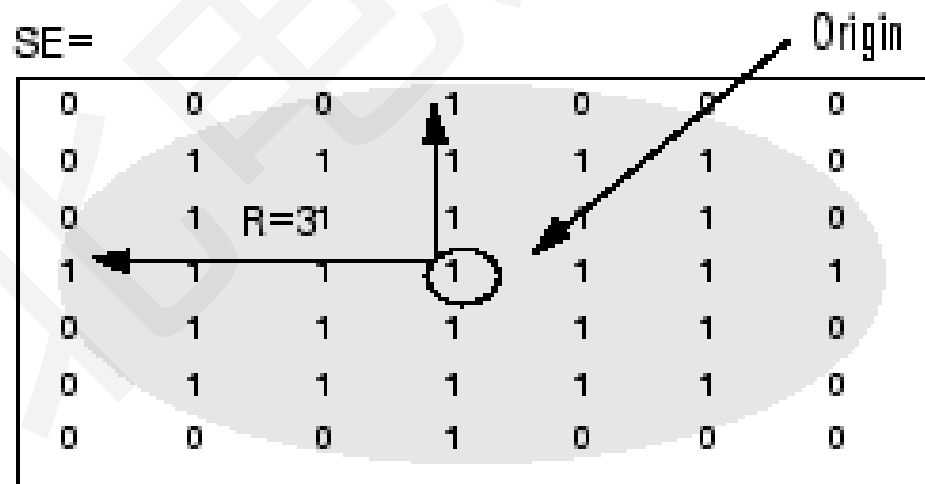


腐蚀后图像



6.3.4 matlab对腐蚀操作的支持

- 膨胀: $Im = \text{imerode}(I, se, n)$
 - ✓ I 是目标图像, se 是结构元素, n 腐蚀次数。
- 结构元素定义: $se = \text{strel}(\text{shape}, \text{parameters})$
 - ✓ shape 是结构元素的形状, parameters 具体参数。





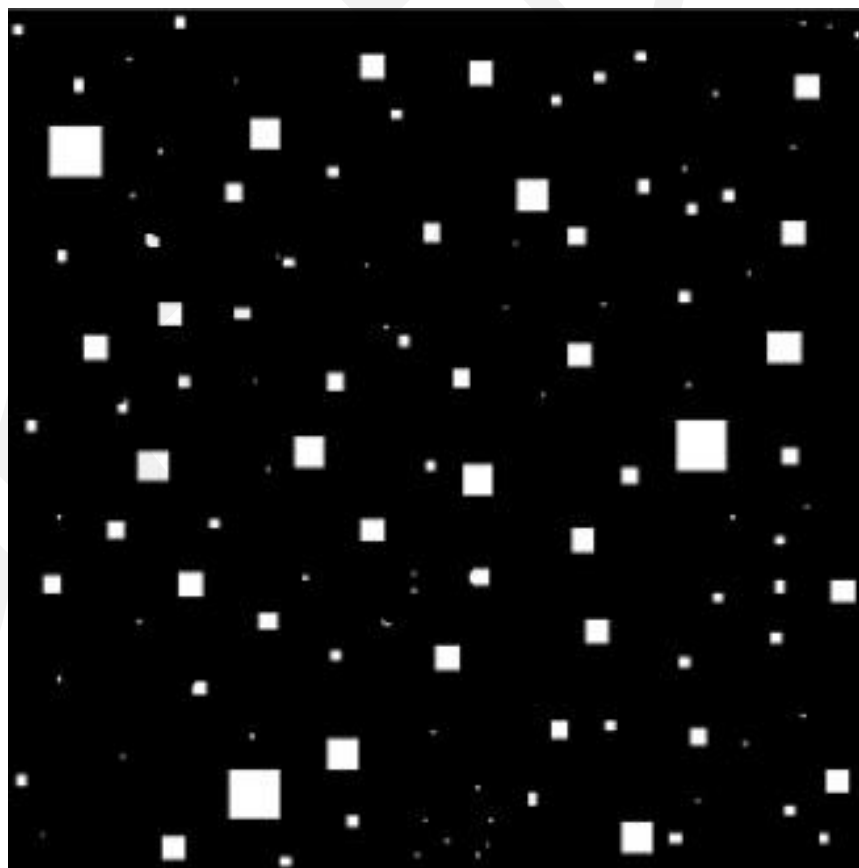
6.3.4 matlab对腐蚀操作的支持

```
I1 = imread('3circles.jpg');  
I1 = rgb2gray(I1);  
I1 = im2double(I1);  
  
se = strel('square',4); %结构元素  
I2 = imerode(I1, se); %腐蚀  
figure(1);  
subplot(1,2,1);imshow(I1);title('目标图像');  
subplot(1,2,2);imshow(I2);title('腐蚀后图像');
```



6.3.4 matlab对腐蚀操作的支持

- 下图中包含边长为3, 5, 7, 9, 11, 13, 1, 5, 17的若干正方形, 如何只保留边长最大的正方形?

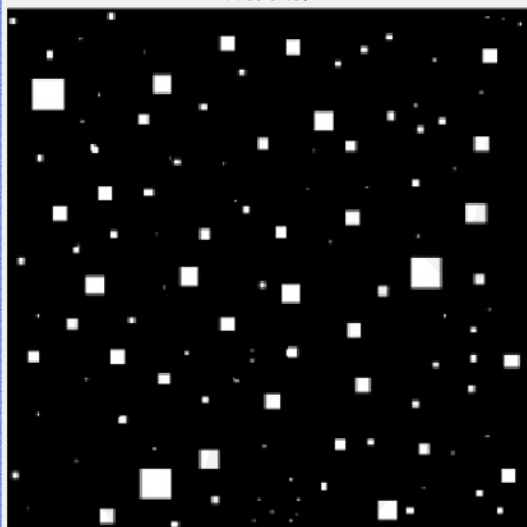




6.3.4 matlab对腐蚀操作的支持

- 下图中包含边长为3, 5, 7, 9, 11, 13, 1, 5, 17的若干正方形, 如何只保留边长最大的正方形?

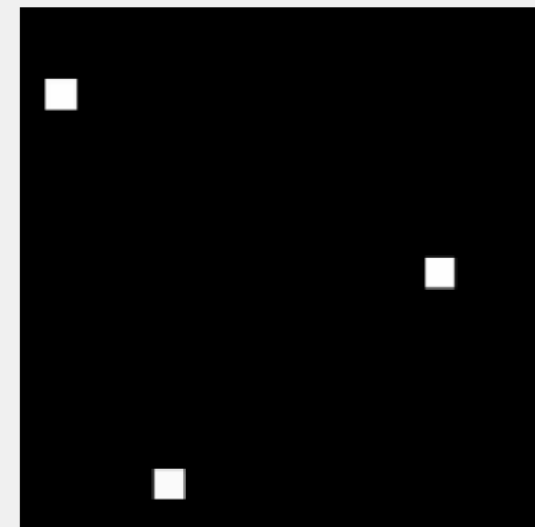
目标图像



15*15正方形腐蚀



15*15正方形膨胀



6.3.4 matlab对腐蚀操作的支持

- 也可以对灰度图进行膨胀和腐蚀，膨胀后图像变亮，腐蚀后图像变暗，两者相减提取边缘。

目标图像



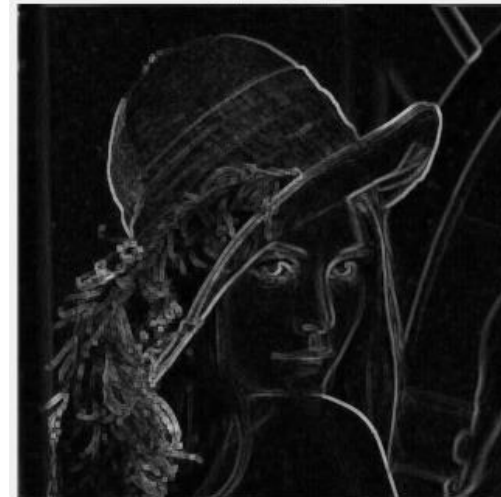
disk腐蚀



disk膨胀

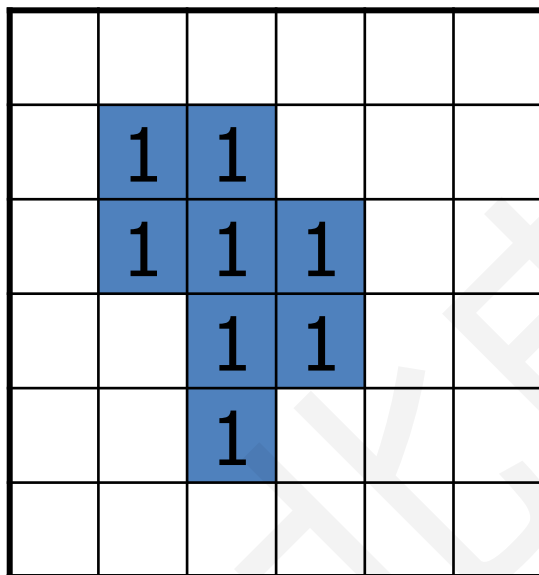


膨胀-腐蚀

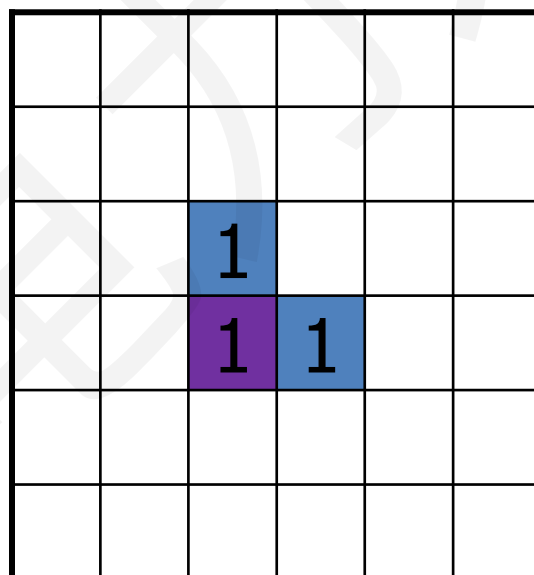


6.3.5 膨胀与腐蚀的对偶性

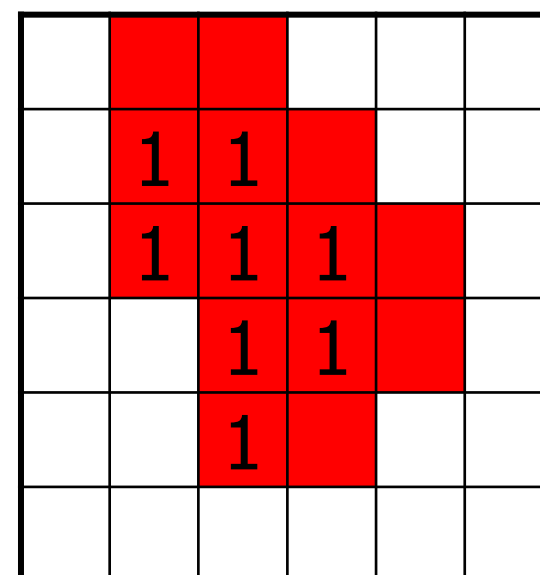
- 对目标图像膨胀和腐蚀



目标图像A



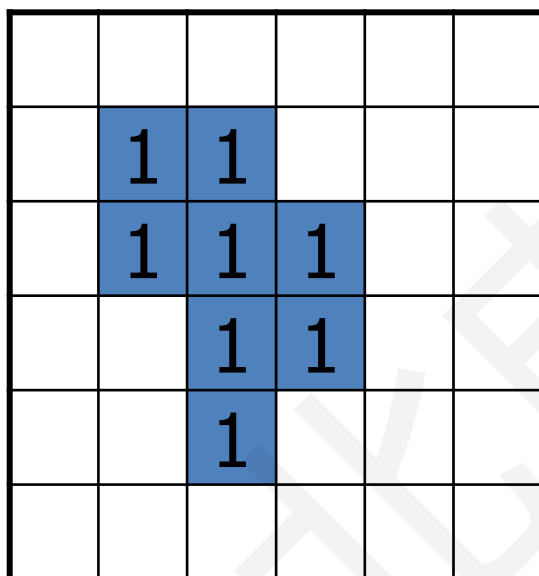
结构元素B



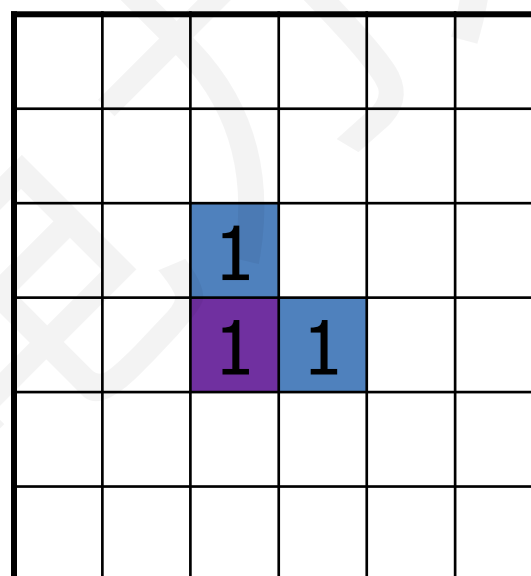
膨胀结果 $A \oplus B$

6.3.5 膨胀与腐蚀的对偶性

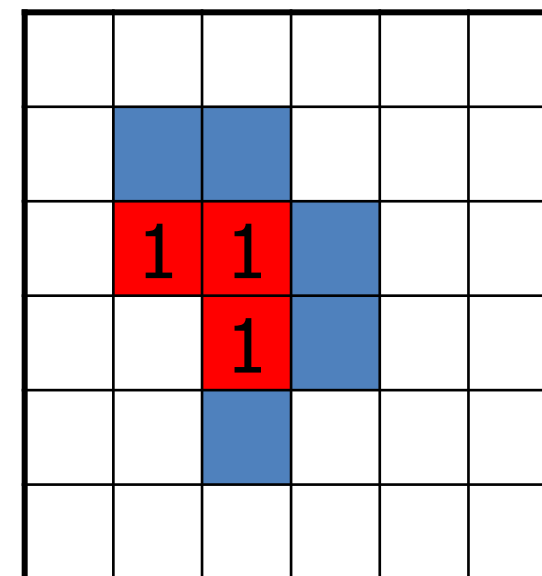
- 对目标图像膨胀和腐蚀



目标图像A



结构元素B



腐蚀结果 $A \ominus B$



6.3.5 膨胀与腐蚀的对偶性

- 对目标图像膨胀和腐蚀

1	1	1	1	1	1
1			1	1	1
1				1	1
1	1			1	1
1	1		1	1	1
1	1	1	1	1	1

目标图像 A^c

		1	1		
			1		

结构元素 \hat{B}

1	1	1	1	1	1
1			1	1	1
1				1	1
1	1			1	1
1	1		1	1	1
1	1	1	1	1	1

膨胀结果 $A^c \oplus \hat{B}$

6.3.5 膨胀与腐蚀的对偶性

- 对目标图像膨胀和腐蚀

1	1	1	1	1	1
1			1	1	1
1				1	1
1	1			1	1
1	1		1	1	1
1	1	1	1	1	1

目标图像 A^c

		1	1		
			1		

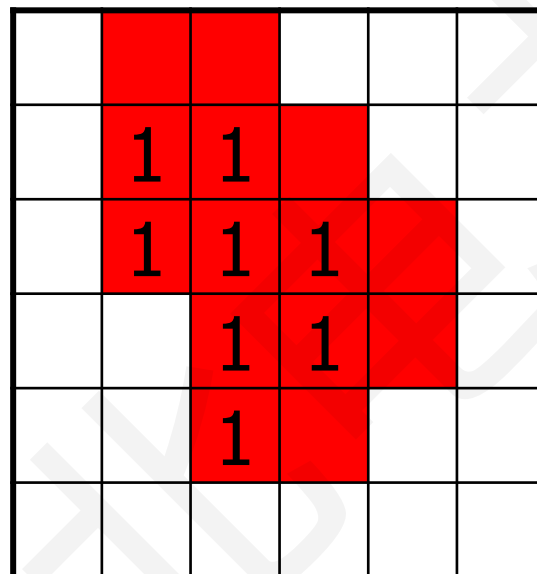
结构元素 \hat{B}

1			1	1	1
1				1	1
1					1
1	1				1
1	1			1	1
1	1	1	1	1	1

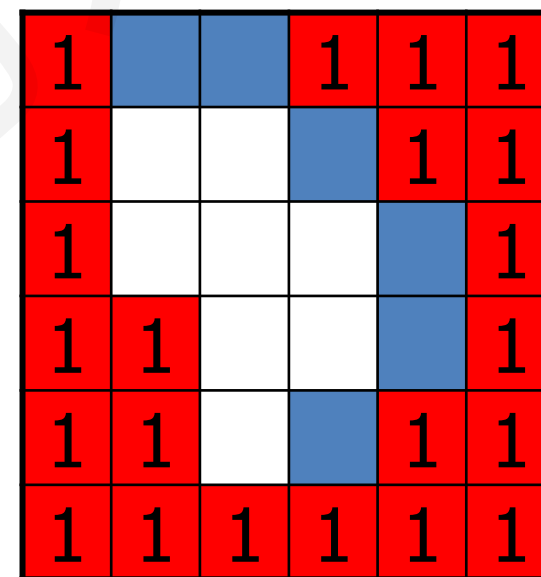
腐蚀结果 $A^c \ominus \hat{B}$

6.3.5 膨胀与腐蚀的对偶性

- 对目标图像的膨胀运算，相当于对图像背景



膨胀结果 $A \oplus B$

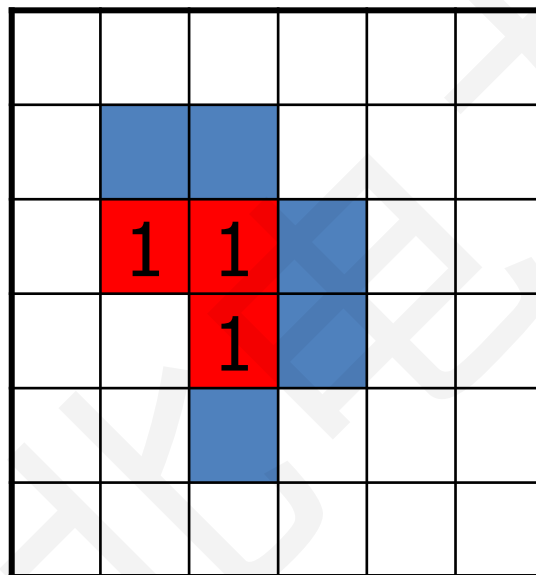


腐蚀结果 $A^c \ominus \hat{B}$

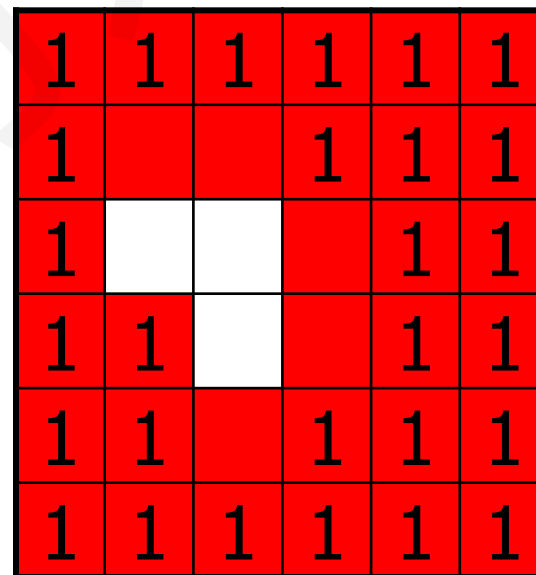
$$(A \oplus B)^c = A^c \ominus \hat{B}$$

6.3.5 膨胀与腐蚀的对偶性

- 对目标图像的腐蚀运算，相当于对图像背景



腐蚀结果 $A \ominus B$

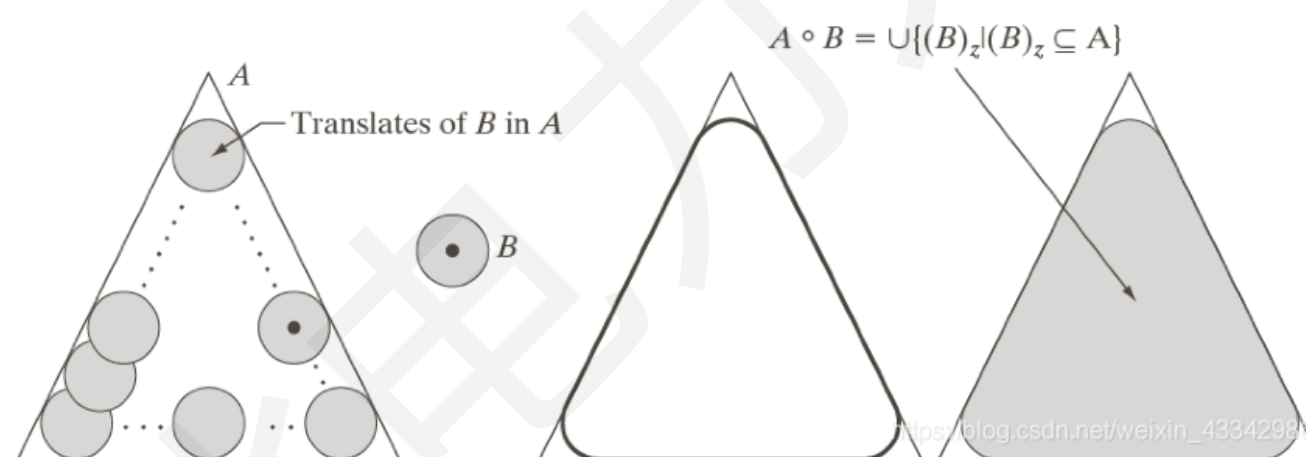


膨胀结果 $A^c \oplus \hat{B}$

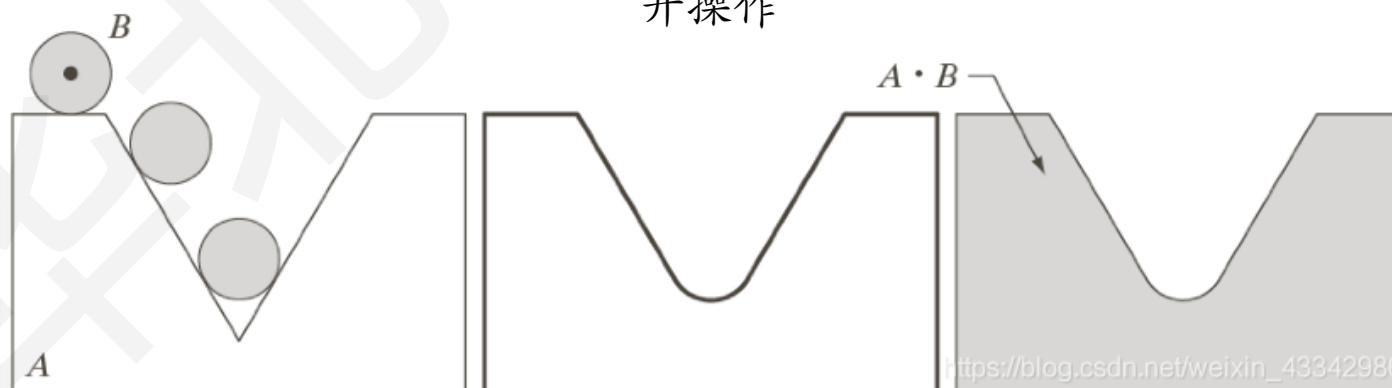
$$(A \ominus B)^c = A^c \oplus \hat{B}$$

6.4 开操作与闭操作

- 开操作与闭操作是基于膨胀和腐蚀的组合，目的是使图像平滑。



开操作



闭操作



6.4.1 开操作

- 使图像的轮廓变得光滑，断开狭窄的间断和消除细的突出物。

✓ 使用结构元素B对集合A进行开操作，定义为

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

✓ 先腐蚀、后膨胀。



6.4.1 开操作

$$B$$

1	
1	1

$$\hat{B}$$

1	1
	1

	1		1		
	1	1		1	
		1			
		1	1		

	1				
		1			

$A \ominus B$

	1				
	1	1			
		1			
		1	1		

$(A \ominus B) \oplus B$

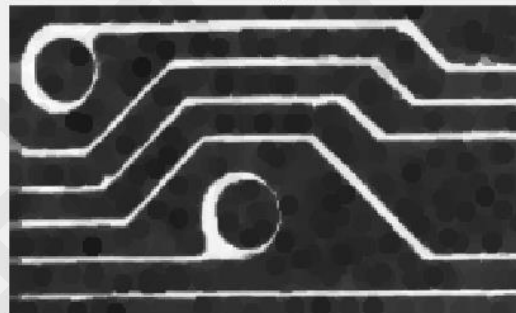


6.4.2 matlab对开操作的支持

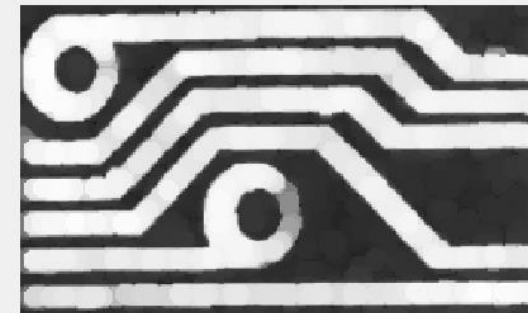
目标图像



disk腐蚀



disk膨胀



开操作可以去除一部分噪声。



6.4.2 matlab对开操作的支持

```
I1 = imread('plate.jpg');
```

```
I1 = rgb2gray(I1);
```

```
I1 = im2double(I1);
```

```
S = strel('disk',5);
```

```
I2 = imerode(I1, S); %先腐蚀
```

```
I3 = imdilate(I2, S); %后膨胀
```

```
figure(1);
```

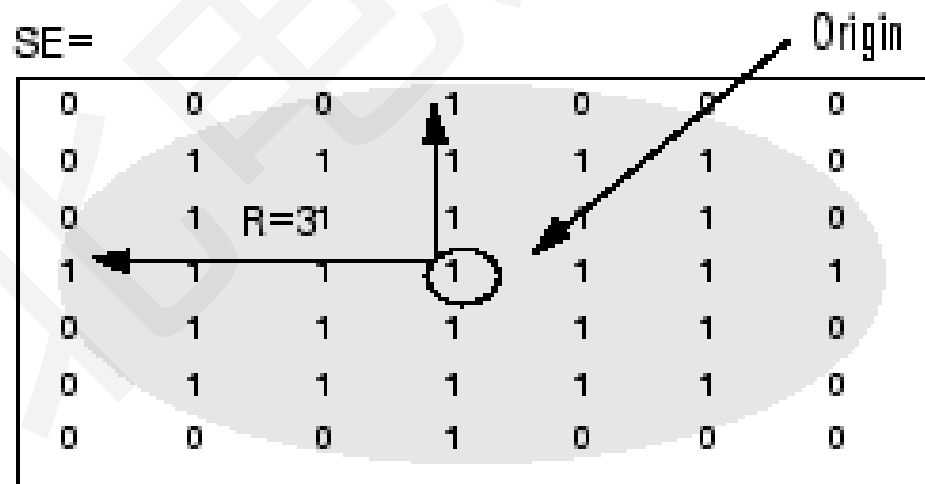
```
subplot(1,3,1);imshow(I1);title('目标图像');
```

```
subplot(1,3,2);imshow(I2);title('disk腐蚀');
```

```
subplot(1,3,3);imshow(I3);title('disk膨胀');
```

6.4.2 matlab对开操作的支持

- 开操作: $I_m = \text{imopen}(I, se)$
 - ✓ I 是目标图像, se 是结构元素。
- 结构元素定义: $se = \text{strel}(\text{shape}, \text{parameters})$
 - ✓ shape 是结构元素的形状, parameters 具体参数。

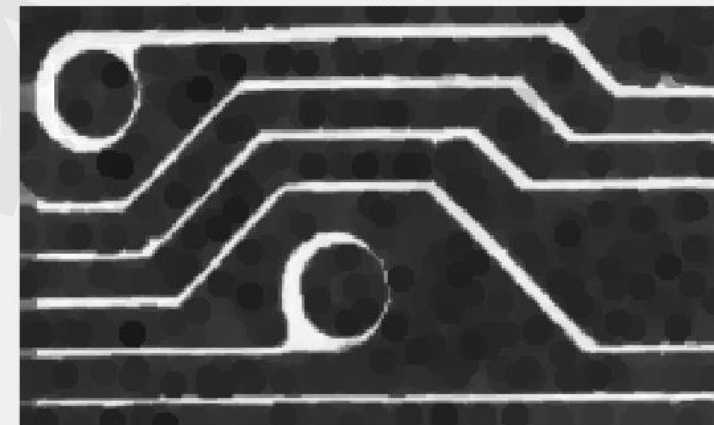


6.4.2 matlab对开操作的支持

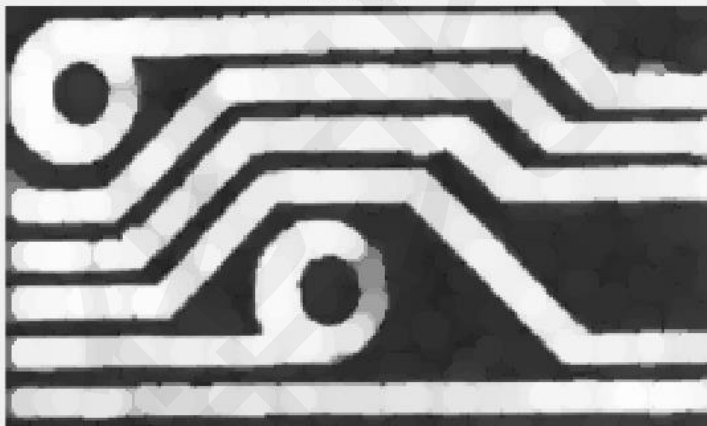
目标图像



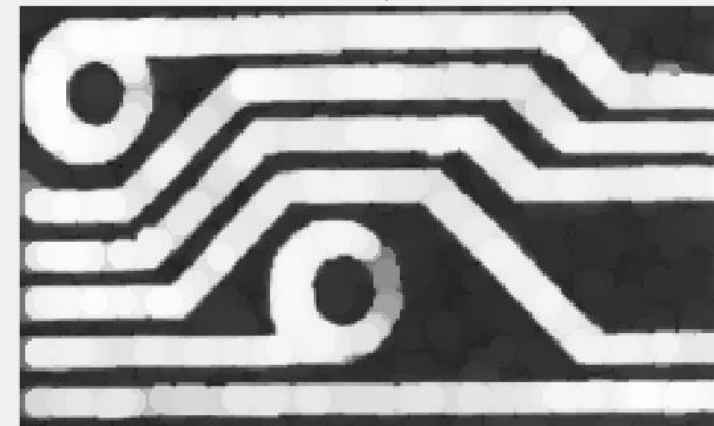
disk腐蚀



disk膨胀



直接open





6.4.2 matlab对开操作的支持

```
I1 = imread('plate.jpg');  
I1 = rgb2gray(I1);  
I1 = im2double(I1);  
  
S = strel('disk',5);  
I2 = imerode(I1, S); %先腐蚀  
I3 = imdilate(I2, S); %后膨胀  
I4 = imopen(I1, S); %开操作  
  
figure(1);  
subplot(2,2,1);imshow(I1);title('目标图像');  
subplot(2,2,2);imshow(I2);title('disk腐蚀');  
subplot(2,2,3);imshow(I3);title('disk膨胀');  
subplot(2,2,4);imshow(I4);title('直接open');
```



6.4.3 闭操作

- 同样使图像的轮廓变得光滑，但与开操作相反，它能消弥狭窄的间断和长细的鸿沟，消除小的孔洞，并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

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

✓ 先膨胀、后腐蚀。

6.4.3 闭操作

$$B$$

1	
1	1

$$^{\wedge}B$$

1	1
	1

	1		1		
	1	1		1	
		1			
		1	1		

	1		1		
	1	1	1	1	
	1	1	1	1	1
		1	1		
		1	1	1	

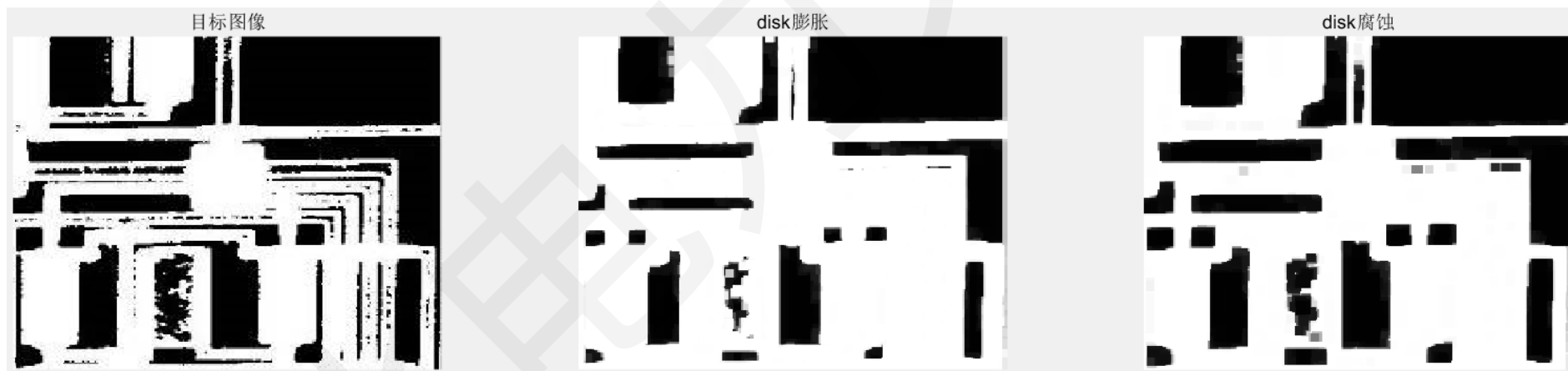
$$A \oplus B$$

	1		1		
	1	1	1	1	
		1			
		1	1		

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



6.4.4 matlab对闭操作的支持



闭操作可以去去除内部的空洞。



6.4.4 matlab对闭操作的支持

```
I1 = imread('plate.jpg');
```

```
I1 = rgb2gray(I1);
```

```
I1 = im2double(I1);
```

```
S = strel('disk',3);
```

```
I2 = imdilate (I1, S); %先膨胀
```

```
I3 = imerode (I2, S); %后腐蚀
```

```
figure(1);
```

```
subplot(1,3,1);imshow(I1);title('目标图像');
```

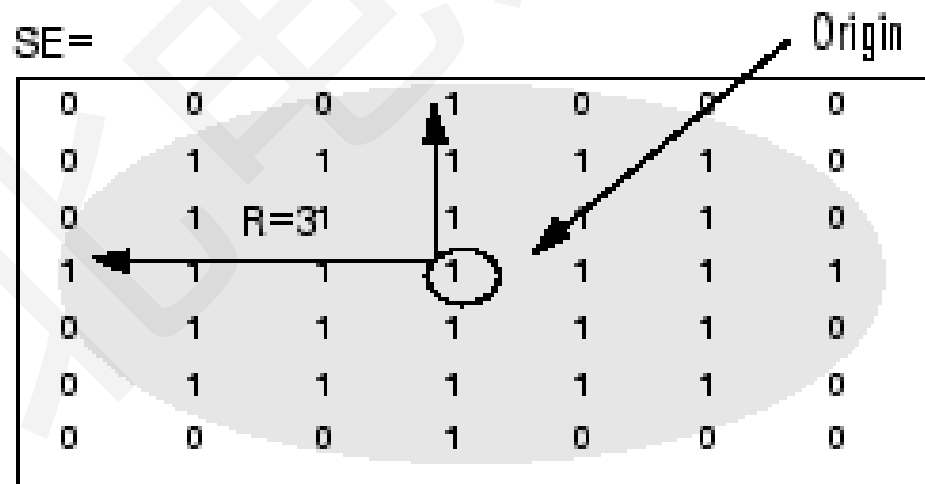
```
subplot(1,3,2);imshow(I2);title('disk膨胀');
```

```
subplot(1,3,3);imshow(I3);title('disk腐蚀');
```



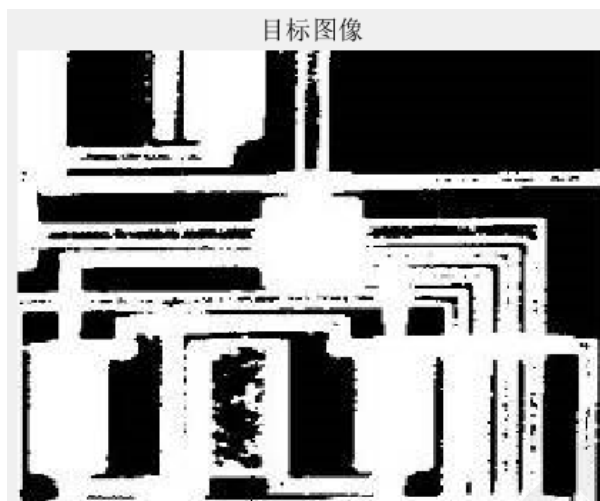
6.4.4 matlab对闭操作的支持

- 闭操作: $I_m = \text{imclose}(I, se)$
 - ✓ I 是目标图像, se 是结构元素。
- 结构元素定义: $se = \text{strel}(\text{shape}, \text{parameters})$
 - ✓ shape 是结构元素的形状, parameters 具体参数。





6.4.4 matlab对闭操作的支持





6.4.4 matlab对闭操作的支持

```
I1 = imread('plate.jpg');
```

```
I1 = rgb2gray(I1);
```

```
I1 = im2double(I1);
```

```
S = strel('disk',3);
```

```
I2 = imdilate (I1, S); %先膨胀
```

```
I3 = imerode (I2, S); %后腐蚀
```

```
I4 = imclose(I1, S); %闭操作
```

```
figure(1);
```

```
subplot(2,2,1);imshow(I1);title('目标图像');
```

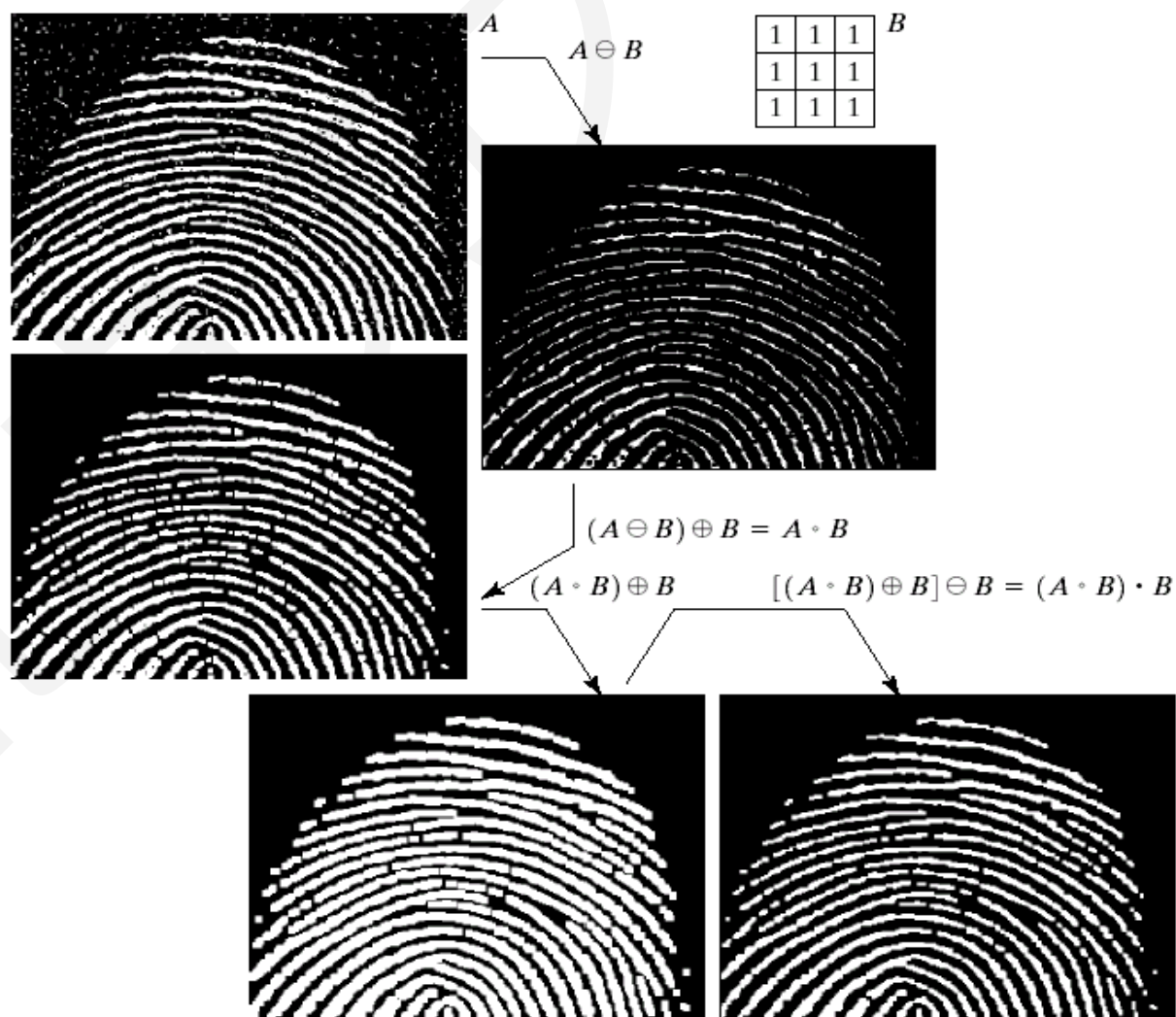
```
subplot(2,2,2);imshow(I2);title('disk膨胀');
```

```
subplot(2,2,3);imshow(I3);title('disk腐蚀');
```

```
subplot(2,2,4);imshow(I4);title('闭操作');
```

6.4.5 开/闭操作的应用

- 先开后闭，开消除噪声，闭修复开运算造成的指纹断裂。





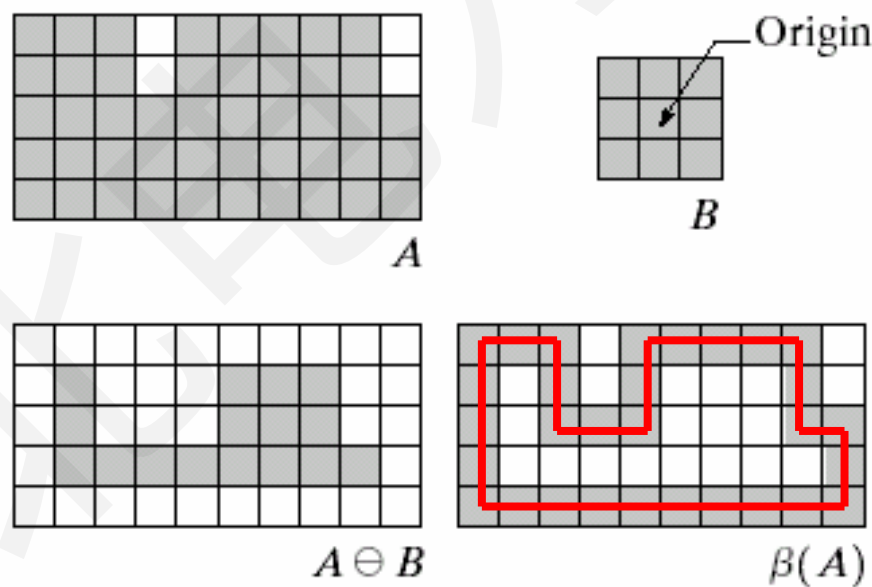
6.5 形态学的主要应用

- 处理图像的类型：二值图像
- 作用：提取对于描绘和表达形状有用的图像成分
 - ✓ 主要应用：边界提取、区域填充、连通分量提取、凸壳和细化以及骨架等。

6.5.1 边界提取

- 用B对A腐蚀，然后用A减去腐蚀得到，B是结构元素。

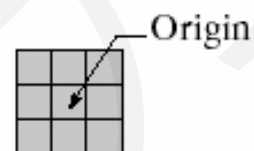
$$\beta(A) = A - (A \ominus B)$$



6.5.1 边界提取

- 用B对A腐蚀，然后用A减去腐蚀得到，B是结构元素。

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



结构元素B





6.5.1 边界提取

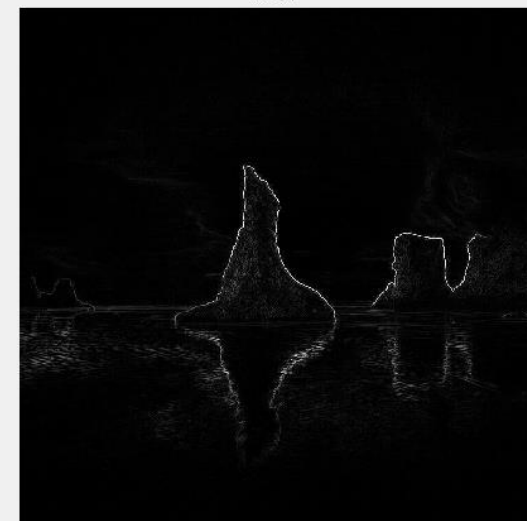
目标图像



腐蚀



边界





6.5.1 边界提取

```
I1 = imread('bandw.jpg');
```

```
I1 = im2double(I1);
```

```
S = strel('square',3);
```

```
I2 = imerode(I1, S); %腐蚀操作
```

```
I3 = I1 - I2; %边界图像
```

```
figure(1);
```

```
subplot(1,3,1);imshow(I1);title('目标图像');
```

```
subplot(1,3,2);imshow(I2);title('腐蚀');
```

```
subplot(1,3,3);imshow(I3);title('边界');
```



6.5.2 区域填充

- 区域填充的公式为：

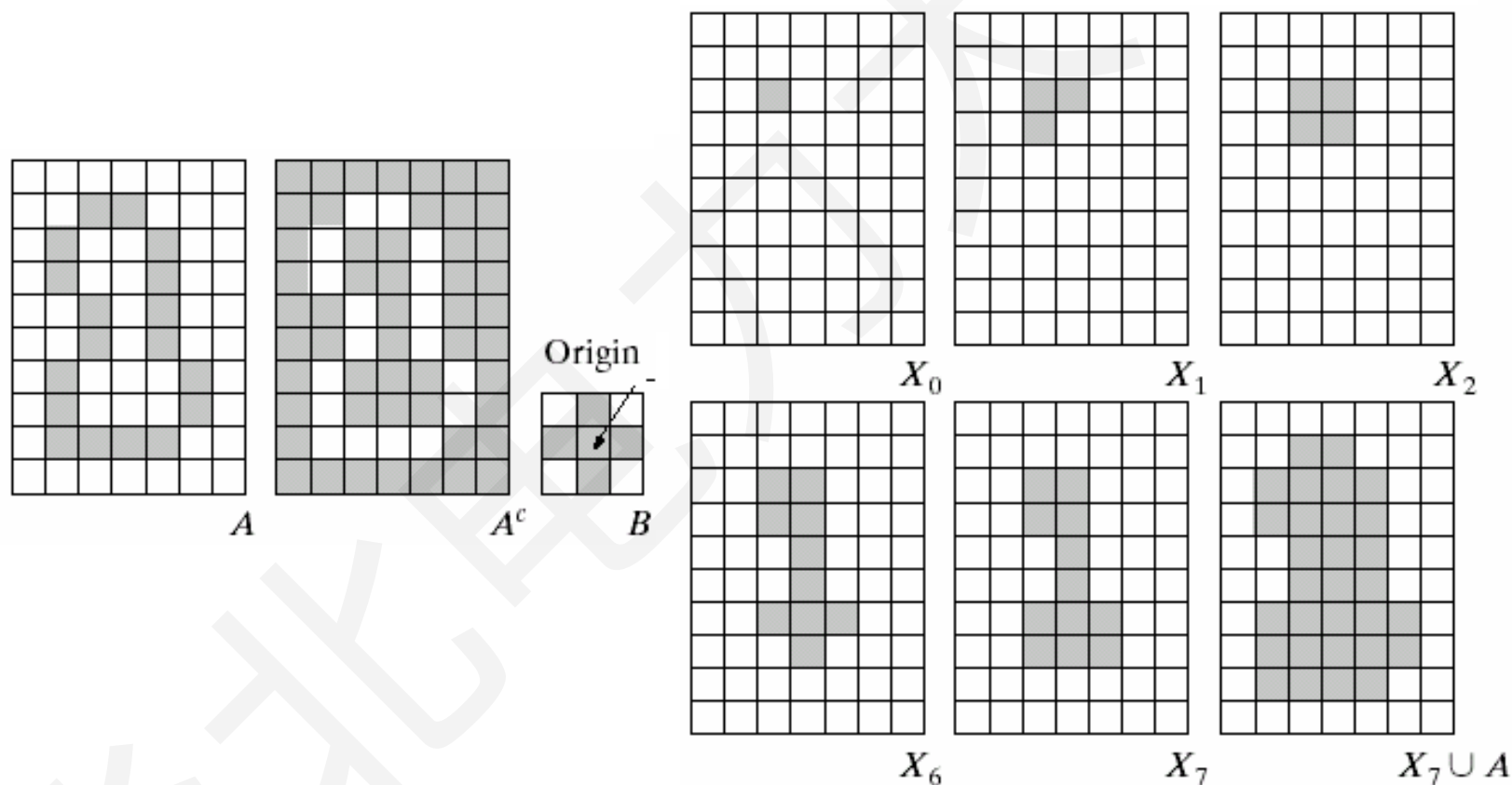
$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots$$

$X_0 = p$ ，如果 $X_k = X_{k-1}$ ，则算法在迭代的第 k 步结束。

X_k 和 A 的并集包含被填充的集合和它的边界。

P 是在边界内的初始点，种子。

6.5.2 区域填充 $X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots$

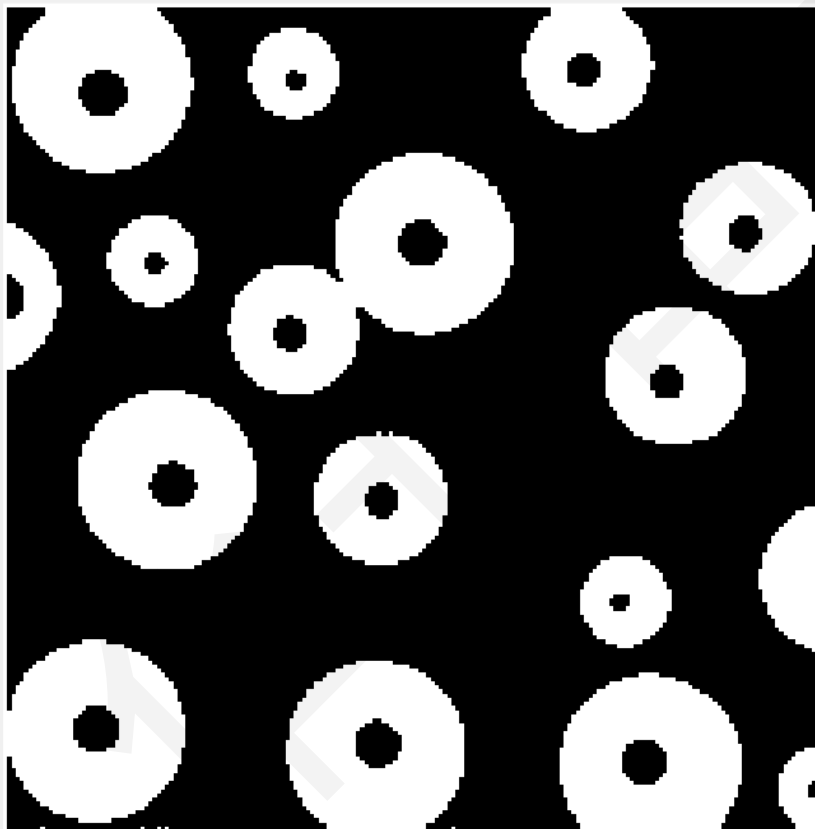


4连通边界内部区域是8连通的，8连通边界内部区域则是4连通的。因此，填充4连通边界的结构元素应选择**3×3正方形**，而填充8连通边界的结构元素应该选择**3×3十字架结构**。

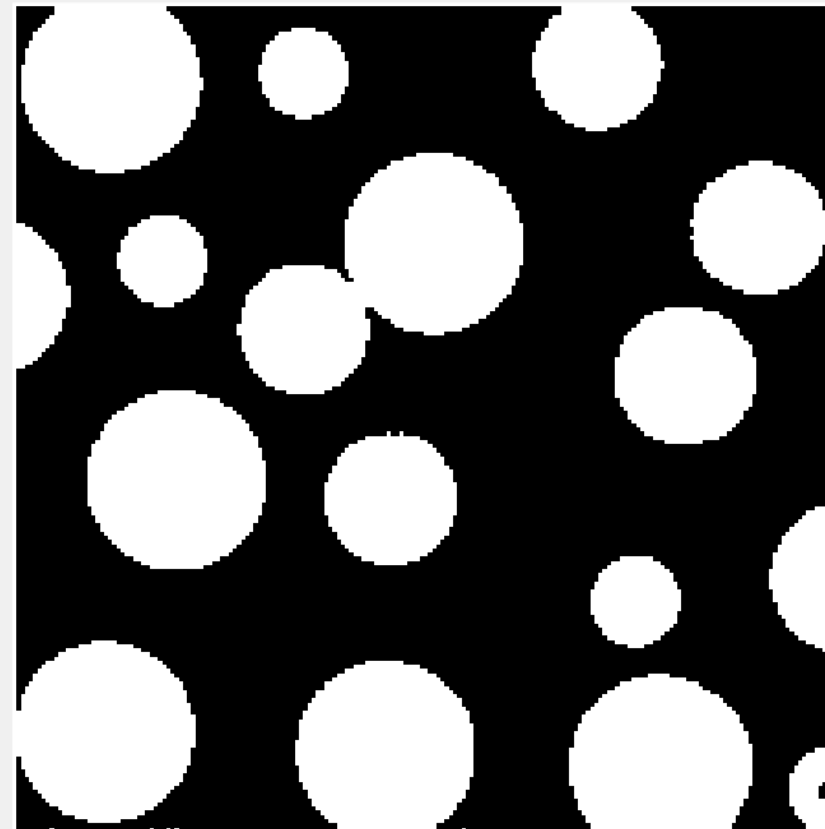
6.5.2 区域填充

- $BW2 = \text{imfill}(BW, 'holes');$
✓ BW 和 $BW2$ 都是二值图像。

目标图像



区域填充





6.5.2 区域填充

```
I1 = imread('circleandhole.jpg');  
I1 = im2bw(I1); %转换为二值图像  
  
I2 = imfill(I1, 'holes'); % 区域填充  
  
figure(1);  
subplot(1,2,1);imshow(I1);title('目标图像');  
subplot(1,2,2);imshow(I2);title('区域填充');
```

6.5.3 梯度

- 图像的梯度可表示为： $g = (A \oplus B) - (A \ominus B)$

