



第9章 图像的数学形态学处理

主讲教师 张涛

电子信息与电气工程



主要内容

- 9.1 形态学基础
- 9.2 二值形态学的基础运算
- 9.3 二值图像的形态学处理
- 9.4 灰度形态学的基础运算
- 9.5 灰度图像的形态学处理



9.1 形态学基础

- 形态学是建立在严格的数学理论基础上的，其数学基础是集合论。
- 在数学形态学中，用集合来描述目标图像或感兴趣区域，描述图像各部分之间关系，描述目标的结构特点。
- 数学形态学运算就是用结构元素对图像集合进行操作，观察图像中各部分关系，从而提取有用特征进行分析和描述，以达到对图像进行分析和识别的目的。



9.1 形态学基础

(1) 结构元素 (Structuring Element)

在分析目标图像时，需要创建一种几何形态滤波模板，用来收集图像信息，称为结构元素。

注意：结构元素也用集合来描述。

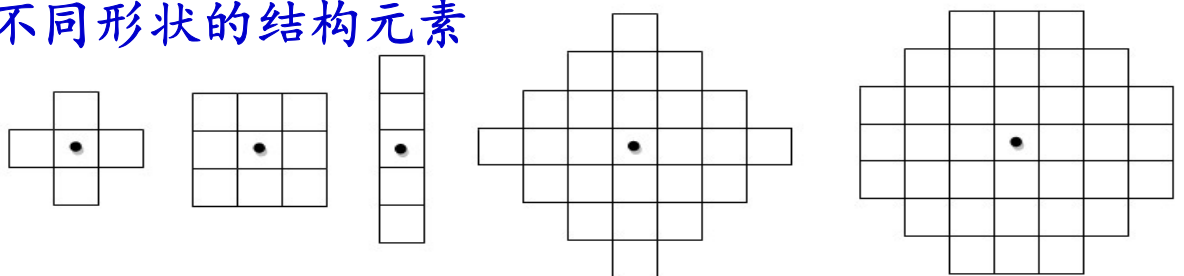
■ 选取结构元素的遵循原则：

- 结构元素的几何结构要比原图像简单且有界，且尺寸要明显小于目标图像的尺寸。
- 结构元素的形状最好具有某种凸性。
- 对于每个结构元素，指定一个原点，作为结构元素参与形态学运算的“参考点”。



9.1 形态学基础

不同形状的结构元素



- 所有形态学处理都基于填充结构元素的概念。
- 所谓的填充，就是用不同的方法把结构元素放在原图像的内部，在结构元素的填充中引出一系列图像的特性。
- 不同的结构元素和不同的形态学变换可以得到图像不同的处理作用结果。



9.1 形态学基础

(2) 例程

■ 函数 `SE = strel (shape , parameters)`

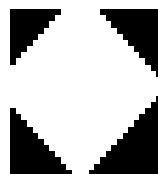
■ 程序

```
SE = strel('diamond',3);
```

```
GN=getnhood(SE);%获取结构元素的邻域
```

```
figure,imshow(GN,[]);
```

■ 效果



0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0
0	0	0	1	0	0	0



9.2 二值形态学的基础运算

9.2.1 基本形态变换

9.2.2 复合形态变换



9.2.1 基本形态变换

二值形态学的基础运算

(1) 膨胀运算定义

集合 X 用结构元素 S 来膨胀记为 $X \oplus S$:

$$X \oplus S = \{x \mid [(\hat{S})_x \cap X] \neq \emptyset\}$$

其含义为：对结构元素 S 作关于原点的映射，所得的映射平移 x ，形成新的集合 $(\hat{S})_x$ ，与集合 X 相交不为空集时结构元素 S 的参考点的集合即为 X 被 S 膨胀所得到的集合。

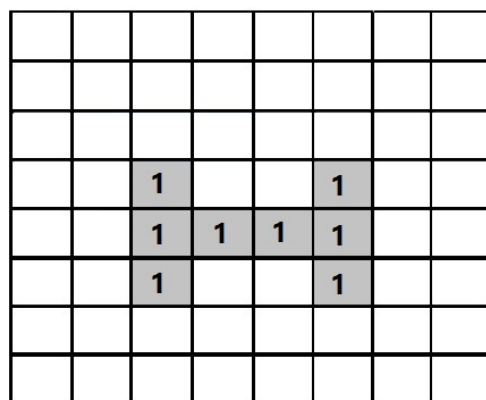


9.2.1 基本形态变换

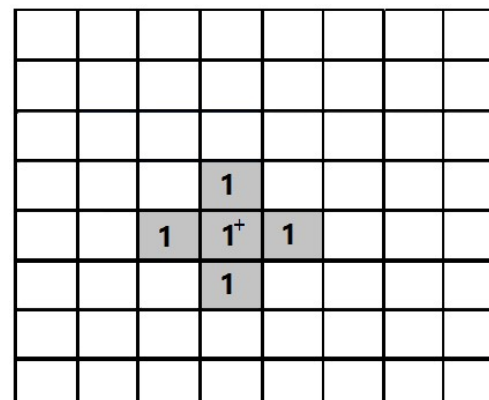
二值形态学的基础运算

(2) 膨胀运算示例一

图 (a) 是一幅二值图像，深色“1”部分为目标集合 X 。图 (b) 中深色“1”部分为结构元素 S (标有“+”处为原点，即结构元素的参考点)。求 $X \oplus S$ 。



(a)

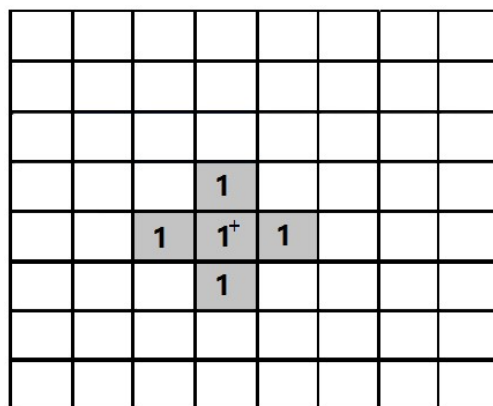


(b)

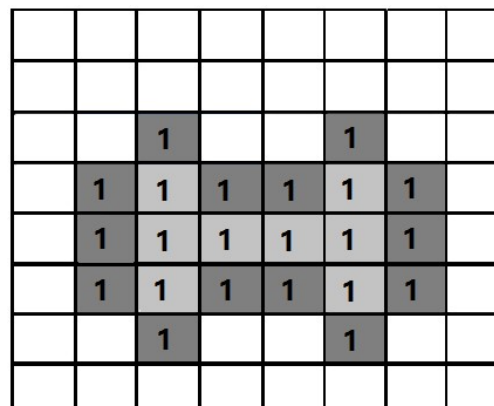


9.2.1 基本形态变换

二值形态学的基础运算



(c)



(d)

解：图(c)中深色“1”部分为结构元素S的映射 \hat{S} 。(d)中深色部分：其中，浅灰色“1”部分表示集合X；深灰色“1”部分表示为膨胀(扩大)部分。则整个深色阴影部分就为集合 $X \oplus S$ 。



9.2.1 基本形态变换

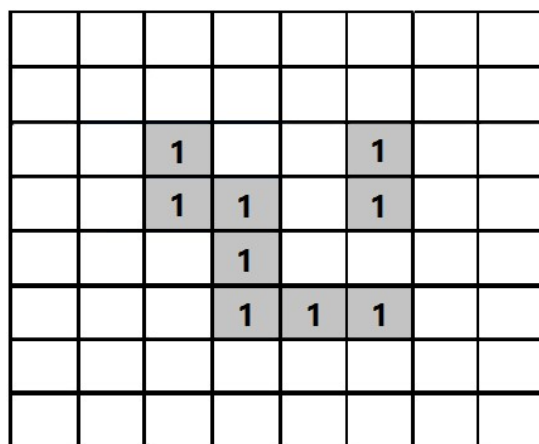
二值形态学的基础运算

(3) 膨胀运算示例二

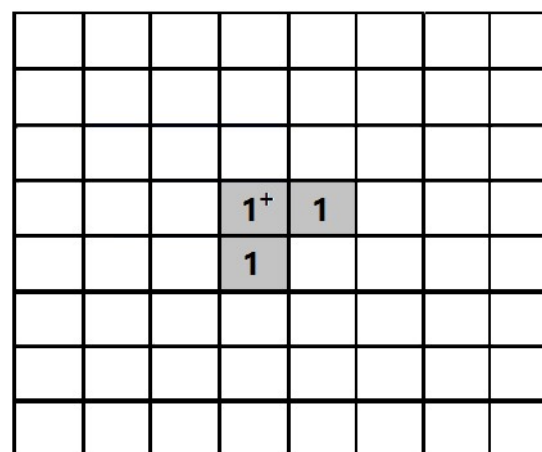
图 (a) 中深色 “1” 部分为目标集合 X 。

图 (b) 中深色 “1” 部分为结构元素 S 。

求 $X \oplus S$ 。



(a)

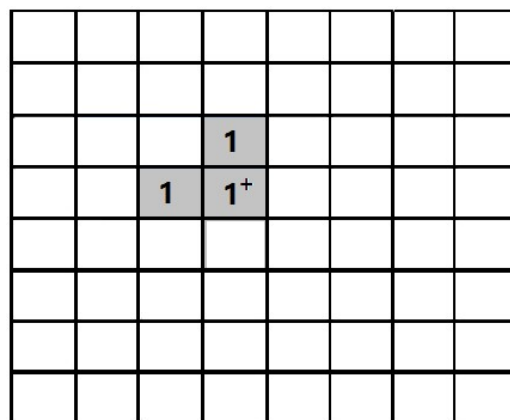


(b)

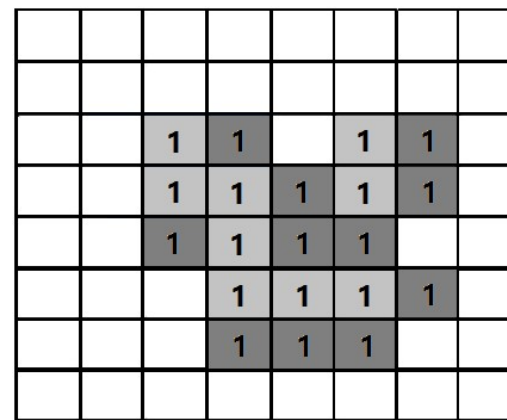


9.2.1 基本形态变换

二值形态学的基础运算



(c)



(d)

解：(c) 将 S 映射为 \hat{S} ，将 \hat{S} 在 X 上移动，记录交集不为空时结构元素参考点位置，得(d)所示深色阴影部分为膨胀后结果。可看出，目标集合经膨胀后不仅面积扩大，而且相邻两个孤立成分连接。



9.2.1 基本形态变换

二值形态学的基础运算

(4) 膨胀运算例程(1)

■ 程序

```
Image=imread('menu.bmp');  
BW=im2bw(Image);  
[h w]=size(BW);  
result=zeros(h,w);  
for x=2:w-1  
    for y=2:h-1  
        for m=-1:1  
            for n=-1:1
```



9.2.1 基本形态变换

二值形态学的基础运算

```
        if BW(y+n,x+m)
            result(y,x)=1;
            break;
        end
    end
end
end
end
figure,imshow(result);title('二值图像膨胀');
```



9.2.1 基本形态变换

二值形态学的基础运算

(4) 膨胀运算例程(2)

■ 函数

IM2 = imdilate(IM,SE,SHAPE):

■ 程序

Image=imread('menu.bmp');

BW=im2bw(Image);

SE=strel('square',3); %创建结构元素

result=imdilate(BW,SE); %膨胀运算

figure,imshow(result);



9.2.1 基本形态变换

二值形态学的基础运算

■ 效果



(a) 原图像



(b) 膨胀图像



9.2.1 基本形态变换

二值形态学的基础运算

(5) 腐蚀运算定义

集合 X 用结构元素 S 来腐蚀记为 $X \ominus S$ ，为：

$$X \ominus S = \{x | (S)_x \subseteq X\}$$

其含义为：若结构元素 S 平移 x 后完全包括在集合 X 中，记录 S 的参考点位置，所得集合为 S 腐蚀 X 的结果。



9.2.1 基本形态变换

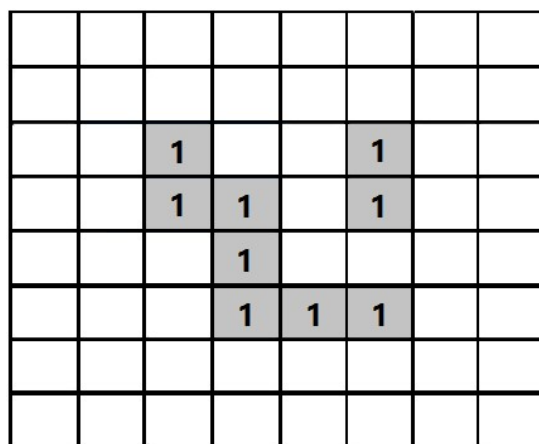
二值形态学的基础运算

(6) 腐蚀运算示例

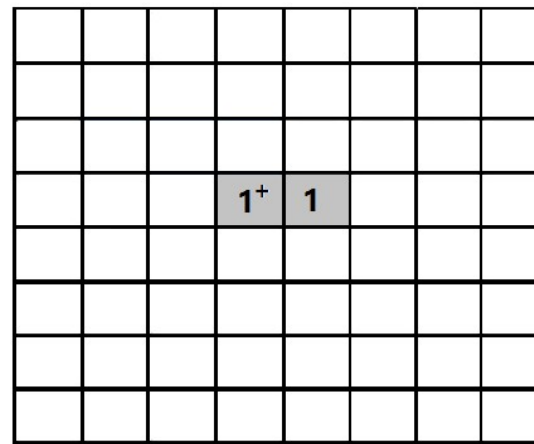
图 (a) 中深色 “1” 部分为目标集合 X 。

图 (b) 中深色 “1” 部分为结构元素 S 。

求 $X \ominus S$ 。



(a)

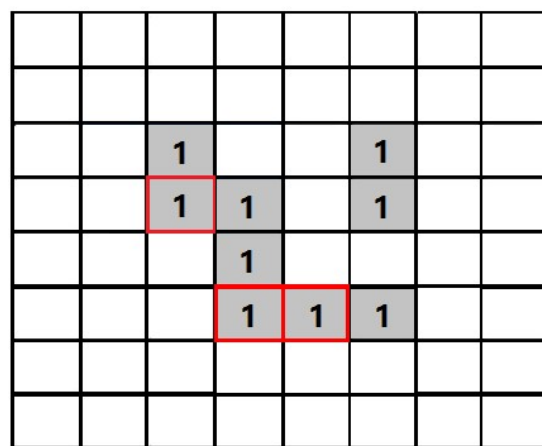


(b)

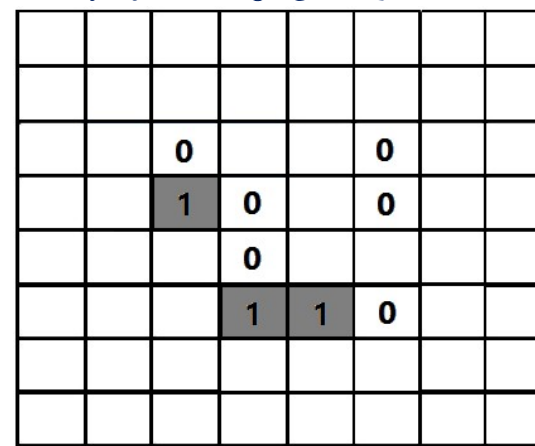


9.2.1 基本形态变换

二值形态学的基础运算



(c)



(d)

解：当 S 参考点位于图(c)中红框“1”部分时， $(S)_x \subseteq X$ ，则 $X \ominus S$ 为图(d)中深灰色“1”部分。白色“0”部分为腐蚀消失部分。



9.2.1 基本形态变换

二值形态学的基础运算

(7) 腐蚀运算例程(1)

■ 程序

```
Image=imread('menu.bmp');  
BW=im2bw(Image);  
[h w]=size(BW);  
result=ones(h,w);  
for x=2:w-1  
    for y=2:h-1  
        for m=-1:1  
            for n=-1:1
```



9.2.1 基本形态变换

二值形态学的基础运算

```
        if BW(y+n,x+m)==0
            result(y,x)=0;
            break;
        end
    end
end
end
end
figure,imshow(result);title('二值图像腐蚀');
```



9.2.1 基本形态变换

二值形态学的基础运算

(7) 腐蚀运算例程(2)

■ 函数

IM2 = imerode(IM,SE,SHAPE):

■ 代码

Image=imread('menu.bmp');

BW=im2bw(Image);

SE=strel('square',3); %创建结构元素

result=imerode(BW,SE); %腐蚀运算

figure,imshow(result);



9.2.1 基本形态变换

二值形态学的基础运算

■ 效果



(a) 原图像



(b) 腐蚀图像



9.2.1 基本形态变换

二值形态学的基础运算

(8) 膨胀和腐蚀的向量定义

■ 向量定义

$$X \oplus S = \{y | y = x + s, x \in X, s \in S\}$$

$$X \ominus S = \{x | (x + s) \in X, x \in X, s \in S\}$$

- **膨胀**为图像 X 中每一点 x 按照结构元素 S 中每一点 s 进行平移的并集。
- **腐蚀**为图像 X 中每一点 x 平移 s 后仍在图像 X 内部的参考点集合。



9.2.1 基本形态变换

二值形态学的基础运算

(9) 示例——膨胀的向量运算

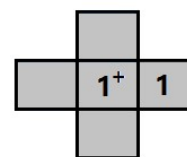
图 (a) 中深色 “1” 部分为目标集合 X 。

图 (b) 中深色 “1” 部分为结构元素 S 。

求用向量运算实现 $X \oplus S$ 。

	0	1	2	3	4	5	6	7
0								
1								
2								
3				1		1		
4				1	1	1		
5				1		1		
6								
7								

(a)



(b)



9.2.1 基本形态变换

二值形态学的基础运算

解：以像素坐标系将 X 、 S 中各点表示为向量：

$$X = \{(2,3), (5,3), (2,4), (3,4), (4,4), (5,4), (2,5), (5,5)\}$$

$$S = \{(0,-1), (-1,0), (0,0), (1,0), (0,1)\}$$

则向量运算膨胀结果为：

$$X \oplus S = \begin{bmatrix} (2,2), & (5,2), & (1,3), & (2,3), & (3,3), & (4,3), \\ (5,3), & (6,3), & (1,4), & (2,4), & (3,4), & (4,4), \\ (5,4), & (6,4), & (1,5), & (2,5), & (3,5), & (4,5), \\ (5,5), & (6,5), & (2,6), & (5,6) \end{bmatrix}$$



9.2.1 基本形态变换

二值形态学的基础运算

(10) 示例——腐蚀的向量运算

图 (a) 中深色 “1” 部分为目标集合 X 。

图 (b) 中深色 “1” 部分为结构元素 S 。

求用向量运算实现 $X \ominus S$ 。

	0	1	2	3	4	5	6	7
0								
1								
2			1			1		
3			1	1		1		
4				1				
5				1	1	1		
6								
7								

(a)

1 ⁺	1
----------------	---

(b)



9.2.1 基本形态变换

二值形态学的基础运算

解：以像素坐标系将 X 、 S 中各点表示为向量：

$$X = \{(2,2), (5,2), (2,3), (3,3), (5,3), (3,4), (3,5), (4,5), (5,5)\}$$

$$S = \{(0,0), (1,0)\}$$



$$\left\{ \begin{array}{ll} x_1 : x_1 + s_2 = (3,2) \notin X & x_2 : x_2 + s_2 = (6,2) \notin X \\ x_4 : x_4 + s_2 = (4,3) \notin X & x_5 : x_5 + s_2 = (6,3) \notin X \\ x_6 : x_6 + s_2 = (4,4) \notin X & x_9 : x_9 + s_2 = (6,5) \notin X \end{array} \right.$$



9.2.1 基本形态变换

二值形态学的基础运算

$$x_3 : \begin{cases} x_3 + s_1 = (2, 3) \in X \\ x_3 + s_2 = (3, 3) \in X \end{cases} \quad x_7 : \begin{cases} x_7 + s_1 = (3, 5) \in X \\ x_7 + s_2 = (4, 5) \in X \end{cases}$$

$$x_8 : \begin{cases} x_8 + s_1 = (4, 5) \in X \\ x_8 + s_2 = (5, 5) \in X \end{cases}$$

因此，向量运算腐蚀结果为：

$$X \ominus S = \{x_3, x_7, x_8\} = \{(2, 3), (3, 5), (4, 5)\}$$



9.2.1 基本形态变换

二值形态学的基础运算

(11) 膨胀和腐蚀的性质

- 性质1：膨胀和腐蚀运算是关于集合补和映射的对偶关系

$$(X \ominus S)^c = X^c \oplus \hat{S}$$

$$(X \oplus S)^c = X^c \ominus \hat{S}$$

- 性质2：膨胀运算具有互换性；
腐蚀运算不具有；

$$(X \oplus S_1) \oplus S_2 = (X \oplus S_2) \oplus S_1$$



9.2.1 基本形态变换

二值形态学的基础运算

(11) 膨胀和腐蚀的性质

- 性质3：膨胀运算具有结合性

若 $S = S_1 \oplus S_2$ ，则有：

$$X \oplus S = X \oplus (S_1 \oplus S_2) = (X \oplus S_1) \oplus S_2$$

- 性质4：膨胀和腐蚀运算具有增长性

$$X \subseteq Y \Rightarrow (X \oplus S) \subseteq (Y \oplus S)$$

$$X \subseteq Y \Rightarrow (X \ominus S) \subseteq (Y \ominus S)$$



9.2.2 复合形态变换

二值形态学的基础运算

一般情况下，膨胀和腐蚀不是互为逆运算的。

膨胀和腐蚀进行级连结合使用，产生新的形态变换，即开运算(Opening)和闭运算(Closing)。

(1) 开、闭运算的定义

- **开运算**：用结构元素对图像**先腐蚀，再膨胀**。

$$X \circ S = (X \ominus S) \oplus S$$

- **闭运算**：用结构元素对图像**先膨胀，再腐蚀**。

$$X \cdot S = (X \oplus S) \ominus S$$



9.2.2 复合形态变换

二值形态学的基础运算

(2) 例程(1)

■ 程序

```
Image=imread('A.bmp');  
BW=im2bw(Image);  
SE=strel('square',3);  
result1=imdilate(imerode(BW,SE),SE);  
result2=imerode(imdilate(BW,SE),SE);  
figure,imshow(result1);title('开运算');  
figure,imshow(result2);;title('闭运算');
```



9.2.2 复合形态变换

二值形态学的基础运算

(2) 例程(2)

■ 函数 `IM2 = imopen(IM,SE):`

`IM2 = imclose(IM,SE):`

■ 程序

```
Image=imread('A.bmp');
```

```
BW=im2bw(Image);
```

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

```
result1=imopen (BW,SE),SE);
```

```
result2=imclose(BW,SE),SE);
```

```
figure,imshow(result1) ;title('开运算');
```

```
figure,imshow(result2); ;title('闭运算');
```



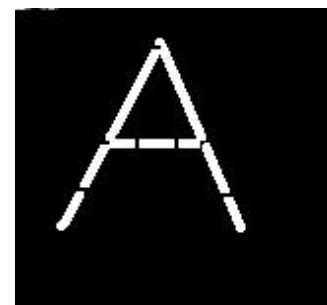
9.2.2 复合形态变换

二值形态学的基础运算

■ 效果



(a) 原图



(b) 开运算结果



(c) 闭运算结果



9.2.2 复合形态变换

二值形态学的基础运算

■ 开运算的效果

- 删除小物体(滤掉突刺)
- 将物体拆分为小物体(切断细长搭接)起分离作用
- 平滑大物体边界而不明显改变它们面积(平滑图像轮廓)

■ 闭运算的效果

- 填充物体的裂缝和小洞
- 连相近物体(搭接短的间断)起连通补接作用
- 平滑大物体边界而不明显改变它们面积(平滑图像轮廓)



9.2.2 复合形态变换

二值形态学的基础运算

(4) 开、闭运算的性质

- 性质1：开运算和闭运算都具有增长性

$$X \subseteq Y \Rightarrow \begin{cases} (X \circ S) \subseteq Y \\ (X \cdot S) \subseteq Y \end{cases}$$

- 性质2：开运算是非外延的，
而闭运算是外延的

$$X \circ S \subseteq X$$

$$X \subseteq X \cdot S$$



9.2.2 复合形态变换

二值形态学的基础运算

(4) 开、闭运算的性质

- 性质3：开运算和闭运算都具有同前性

$$(X \circ S) \circ S = X \circ S$$

$$(X \cdot S) \cdot S = X \cdot S$$

- 性质4：开运算和闭运算都具有对偶性

$$(X \circ S)^c = X^c \cdot \hat{S}$$

$$(X \cdot S)^c = X^c \circ \hat{S}$$



9.3 二值图像的形态学处理

9.3.1 形态滤波

9.3.2 图像的平滑处理

9.3.3 图像的边缘提取

9.3.4 区域填充



9.3.1 形态滤波

二值图像的形态学处理

(1) 原理

- 选择不同形状(如各向同性的圆、十字架、矩形、不同朝向的有向线段等)、不同尺寸的结构元素可以提取图像的不同特征。
- 结构元素的形状和大小会直接影响形态滤波输出结果。



9.3.1 形态滤波

二值图像的形态学处理

(2) 例程——对二值图像特征提取

■ 程序

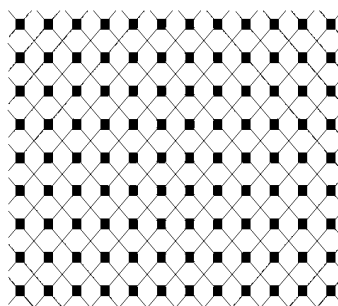
```
I=imread('pattern.jpg');  
BW1=im2bw(I,h);  
BW1=1-BW1;  
se=strel('square',3);  
BW2=1-imopen(BW1,se);  
figure;imshow(BW2);title('矩形块提取');  
se45=strel('line',25,45);  
BW3=1-imopen(BW1,se45);  
figure;imshow(BW3);title('线段提取');
```



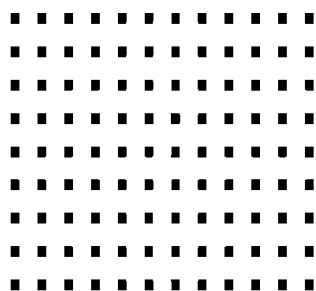
9.3.1 形态滤波

二值图像的形态学处理

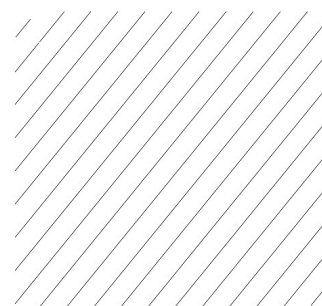
■ 效果



(a) 原始二值图像



(b) 矩形块提取



(c) 45°有向线段提取



9.3.2 图像的平滑处理

二值图像的形态学处理

(1) 原理

- 可通过开运算和闭运算的串行结合来构成形态学噪声滤波器。
- 对二值图像平滑处理的形态学变换为：

$$Y = (X \circ S) \cdot S$$

$$Y = (X \cdot S) \circ S$$

关键为：结构元素的选取



9.3.2 图像的平滑处理

二值图像的形态学处理

(2) 示例——形态学噪声滤波



图像 X



$$X \circ S$$



结构元素 S



$$(X \circ S) \oplus S$$



$$X \ominus S$$



$$(X \ominus S) \cdot S$$

这里 结构元素 S 大于所有噪声孔和噪声块的尺寸。



9.3.2 图像的平滑处理

二值图像的形态学处理

(3) 例程---二值图像的平滑

■ 程序

```
Image=imread('A.bmp');  
BW=im2bw(Image);  
SE=strel('square',3);  
result1=imclose(imopen(BW,SE),SE);  
figure,imshow(result1);  
result2=imopen(imclose(BW,SE),SE);  
figure,imshow(result2);
```



9.3.2 图像的平滑处理

二值图像的形态学处理

■ 效果



图像 A



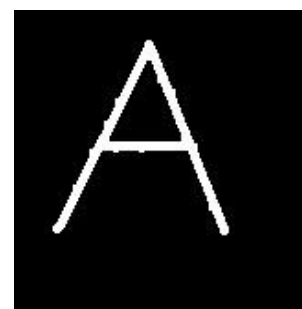
开运算结果



闭运算结果



$(A \circ S) \cdot S$



$(A \cdot S) \circ S$



9.3.3 图像的边缘提取

二值图像的形态学处理

(1) 原理

- 基于形态学提取边缘利用腐蚀运算的特性：
腐蚀运算可以缩小目标，原图像与缩小图像的差即为边界。
- 提取物体的轮廓边缘的形态学变换：

$$\text{内边界:} \quad Y = X - (X \ominus S)$$

$$\text{外边界:} \quad Y = (X \oplus S) - X$$

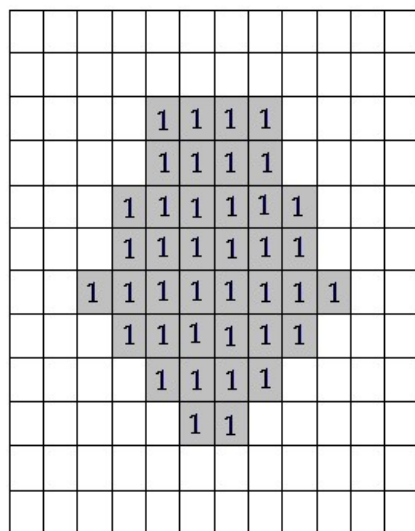
$$\text{形态学梯度:} \quad Y = (X \oplus S) - (X \ominus S)$$



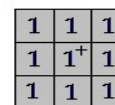
9.3.3 图像的边缘提取

二值图像的形态学处理

(2) 示例——对图像 X 进行边缘提取



图像 X

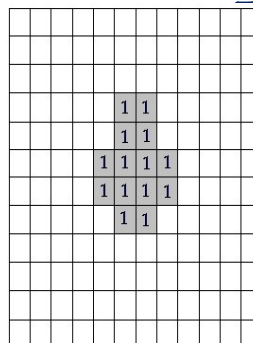


结构元素 S

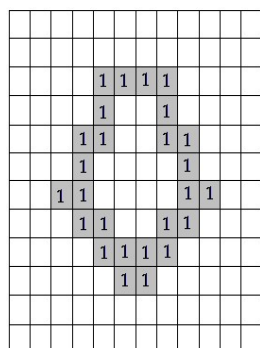


9.3.3 图像的边缘提取

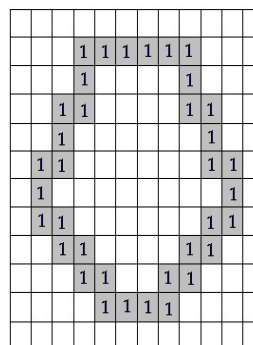
二值图像的形态学处理



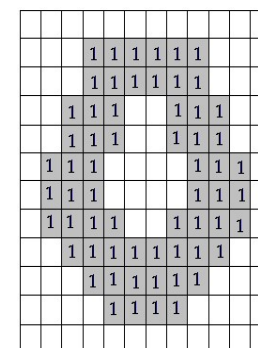
$$X \ominus S$$



$$X - (X \ominus S)$$



$$(X \oplus S) - X$$



$$(X \oplus S) - (X \ominus S)$$



9.3.3 图像的边缘提取

二值图像的形态学处理

(3) 例程(1)---边缘提取

■ 程序

```
Image=imread('menu.bmp');  
BW=im2bw(Image);  
SE=strel('square',3);  
result1=BW-imerode(BW,SE);  
result2=imdilate(BW,SE)-BW;  
result3=imdilate(BW,SE)-imerode(BW,SE);  
figure,imshow(result1);  
figure,imshow(result2);  
figure,imshow(result3);
```



9.3.3 图像的边缘提取

二值图像的形态学处理

(3) 例程(2)——边缘提取

■ 函数

```
IM2=bwperim(IM,conn);
```

■ 程序

```
Image=imread('menu.bmp');
```

```
BW=im2bw(Image);
```

```
result1=bwperim(BW);
```

```
figure,imshow(result1);title('二值图像边缘检测');
```



9.3.3 图像的边缘提取

二值图像的形态学处理

■ 效果



$$X - (X \ominus S)$$



$$(X \oplus S) - X$$



$$(X \oplus S) - (X \ominus S)$$



9.3.4 区域填充

二值图像的形态学处理

(1) 原理

- 区域填充的形态学变换为

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

- A 表示区域边界点集合, k 为迭代次数
- 取边界内某一点 p ($p = x_0$)为起点, 利用上面的公式作迭代运算。
- 当 $x_k = x_{k+1}$ 时停止迭代。
- X_k 和 A 的补集 A^c 的交集就包括了图像边界线所包围的填充区域。



9.3.4 区域填充

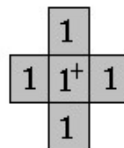
二值图像的形态学处理

(2) 示例---区域填充

边界点用灰色表示，赋值为1；所有非边界点是白色部分，赋值为0。

	1	1	1	1	1	1	1		
	1	0	0	0	0	0	0	1	
	1	0	1	1	1	1	0	0	1
	1	0	1	0	0	0	1	0	1
	1	0	1	1	1	1	0	0	1
	1	0	0	0	0	0	0	1	
	1	0	1	1	0	0	1		
	1	0	1		1	0	0	1	
	1	1	1			1	1	1	

原始图像



结构元素

	1	1	1	1	1	1	1		
	1	0	0	0	0	0	0	1	
	1	0	1	1	1	1	0	0	1
	1	0	1	0	0	0	1	0	1
	1	0	1	1	1	1	0	0	1
	1	0	0	0	0	0	0	1	
	1	0	1	1	0	0	1		
	1	0	1		1	0	0	1	
	1	0	1			1	0	0	1
	1	1	1			1	1	1	

起点 p

	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	0	0	0	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	

填充后图像



9.3.4 区域填充

二值图像的形态学处理

(3) 例程---区域填充

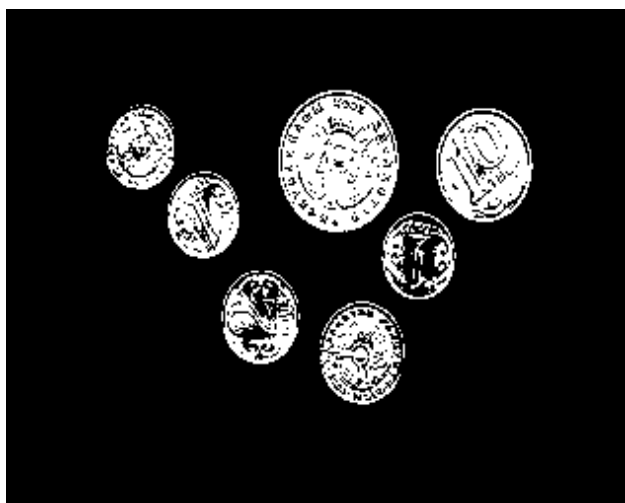
- 函数 `IM2=imfill(IM,'holes')`:
`IM2=imfill(IM,locations,conn):`
`IM2=imfill(i) :`
- 程序
`Image=imread('coin.bmp');`
`BW=im2bw(Image);`
`imshow(BW); title('二值图像');`
`result1=imfill(BW,'holes');`
`figure,imshow(result1);title('二值图像的区域填充');`



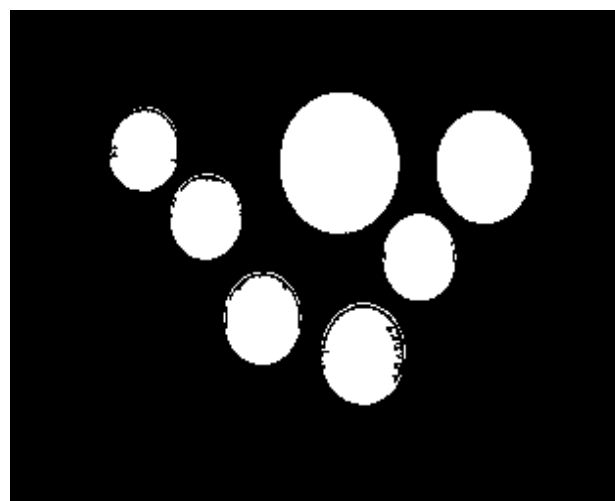
9.3.4 区域填充

二值图像的形态学处理

■ 效果



(a) 原二值图像



(b) 区域填充



9.4 灰度形态学的基础运算

9.4.1 膨胀运算和腐蚀运算

9.4.2 开运算和闭运算



9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

(1) 膨胀运算定义

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

其中， D_f 、 D_b 分别为输入图像 $f(x, y)$ 和结构元素 $b(x, y)$ 的定义域。

- 灰度图像膨胀的含义是把图像 $f(x, y)$ 的每一点反向平移图像 x, y ，平移后与 $b(x, y)$ 相加，在 x, y 取所有值的结果中求最大。



9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

(2) 膨胀运算示例

一幅图像 f ，以 3×3 的方形结构元素 b 对其进行膨胀。（注意：结构元素 b 中标记 $()$ 的为参考点位置）

$$f = \begin{bmatrix} 1 & 2 & 2 & 1 & 1 \\ 1 & 3 & 5 & 4 & 2 \\ 2 & 4 & 3 & 3 & 3 \\ 1 & 2 & 5 & 2 & 1 \\ 3 & 1 & 2 & 1 & 3 \end{bmatrix}$$

$$b = \begin{bmatrix} 1 & 2 & 3 \\ 4 & (5) & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

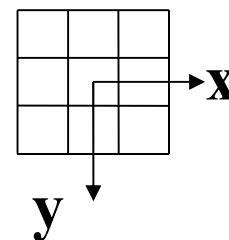


9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

解：结构元素**b**的定义域：中心点为原点，采用像素坐标系

$$D_b = \left\{ (-1, -1), (0, -1), (1, -1), (-1, 0), (0, 0), (1, 0), (-1, 1), (0, 1), (1, 1) \right\}$$



以灰度图像**f**中点 $(s, t) = (2, 2)$ 为例，

■ 平移： $(s - x, t - y)$

$$\{(3, 3), (2, 3), (1, 3), (3, 2), (2, 2), (1, 2), (3, 1), (2, 1), (1, 1)\}$$



9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

■ 相加: $f(s-x, t-y) + b(x, y)$

$$f(3,3) + b(-1,-1) = 2 + 1 = 3$$

$$f(2,3) + b(0,-1) = 5 + 2 = 7$$

⋮

$$f(2,1) + b(0,1) = 5 + 8 = 13$$

$$f(1,1) + b(1,1) = 3 + 9 = 12$$

■ 求最大: 对应上列的所有值取最大为13,
(2,2) 点膨胀后为13



9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

■ 最后膨胀结果为：

$$g = \begin{bmatrix} 1 & 2 & 2 & 1 & 1 \\ 1 & 10 & 11 & 11 & 2 \\ 2 & 12 & 13 & 14 & 3 \\ 1 & 12 & 13 & 12 & 1 \\ 3 & 1 & 2 & 1 & 3 \end{bmatrix}$$

■ 灰度图像的膨胀运算过程：
结构元素先做关于参考点的映射：

$$\hat{b} = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

把映射后的结构元素 \hat{b} 作为模板在图像上移动，模板覆盖区域内，像素值与 \hat{b} 值对应相加，求最大。



9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

- 灰度膨胀运算会产生以下两种效果：
 - 如果在结构元素所定义的邻域中其值都为正，膨胀后 $f \oplus b$ 的值比 f 值大，因此图像会比输入图像亮；
 - 输入图像中暗细节的部分是否在膨胀中被削减或去除，取决于结构元素的形状以及结构元素的值。



9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

(3) 腐蚀运算定义

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

D_f , D_b 是 $f(x, y)$ 和 $b(x, y)$ 的定义域。

- 灰度图像腐蚀的含义是把 $f(x, y)$ 的每一点平移 x, y , 平移后与 $b(x, y)$ 相减, 在 x, y 取所有值的结果中求最小。



9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

(4) 腐蚀运算示例

一幅图像 f ，以 3×3 的方形结构元素 b 对其进行腐蚀。（注意：结构元素 b 中标记 $()$ 的为参考点位置）

$$f = \begin{bmatrix} 1 & 2 & 2 & 1 & 1 \\ 1 & 3 & 5 & 4 & 2 \\ 2 & 4 & 3 & 3 & 3 \\ 1 & 2 & 5 & 2 & 1 \\ 3 & 1 & 2 & 1 & 3 \end{bmatrix} \quad b = \begin{bmatrix} 1 & 2 & 3 \\ 4 & (5) & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

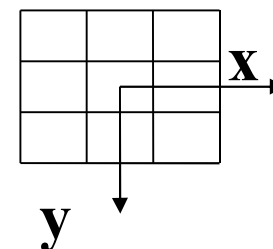


9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

解：结构元素 b 的定义域：中心点为原点，采用像素坐标系

$$D_b = \left\{ \begin{array}{l} (-1, -1), (0, -1), (1, -1), (-1, 0), \\ (0, 0), (1, 0), (-1, 1), (0, 1), (1, 1) \end{array} \right\}$$



以灰度图像 f 中点 $(s, t) = (2, 2)$ 为例，

■ 平移： $(s + x, t + y)$

$$\{(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)\}$$



9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

■ 相减: $f(s+x, t+y) - b(x, y)$

$$f(1,1) - b(-1,-1) = 3 - 1 = 2$$

$$f(2,1) - b(0,-1) = 5 - 2 = 3$$

⋮

$$f(2,3) - b(0,1) = 5 - 8 = -3$$

$$f(3,3) - b(1,1) = 2 - 9 = -7$$

■ 求最小: 对应上列的所有值取最小为-7,
(2,2) 点腐蚀后为-7



9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

■ 最后腐蚀结果为：

$$g = \begin{bmatrix} 1 & 2 & 2 & 1 & 1 \\ 1 & -6 & -6 & -6 & 2 \\ 2 & -6 & -7 & -8 & 3 \\ 1 & -7 & -8 & -7 & 1 \\ 3 & 1 & 2 & 1 & 3 \end{bmatrix}$$

■ 灰度图像的腐蚀运算过程：

把结构元素***b***作为模板在图像上移动，模板覆盖区域内，像素值与***b***值对应相减，求最小。



9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

- 灰度腐蚀运算会产生以下两种效果：
 - 如果在结构元素所定义的邻域中其值都为正，腐蚀后 $f \ominus b$ 的值比 f 值小，因此图像会比输入图像暗；
 - 如果输入图像中亮细节部分的尺寸比结构元素小，则腐蚀后明亮细节将会削弱，削弱程度与该亮细节周围的灰度值和结构元素的形状以及结构元素的值有关。



9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

(5) 例程

■ 程序

```
Image=(imread('maleman.gif'));  
se=strel('ball',5,5); %创建球形结构元素  
result1=imdilate(Image,se); %膨胀灰度图像  
result2=imerode(Image,se); %腐蚀灰度图像  
figure,imshow(result1);title('膨胀后的图像');  
figure,imshow(result2);title('腐蚀后的图像');
```



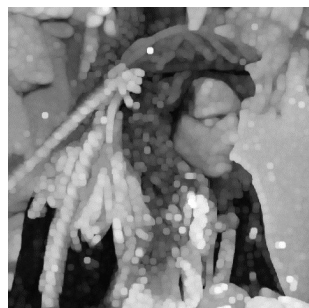
9.4.1 膨胀运算和腐蚀运算

灰度形态学的基础运算

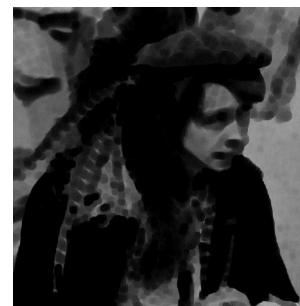
■ 效果



(a) 原始图像



(b) 膨胀后的图像



(c) 腐蚀后的图像



9.4.2 开运算和闭运算

灰度形态学的基础运算

(1) 开、闭运算定义

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

$f(x, y)$ 是一个曲面，采用球状结构元素。

- 进行开运算，相当于让球从曲面的下表面滚过，去掉比结构元素小的亮细节。
- 进行闭运算，相当于让球从曲面的上表面滚过，去掉比结构元素小的暗细节。
- 灰度图像进行开闭运算可达到滤波的目的。



9.4.2 开运算和闭运算

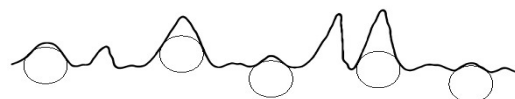
灰度形态学的基础运算



(a) 原始连续图像



(b) 球状结构元素



(c) 求开运算的几个位置



(d) 灰度开运算结果



(e) 求闭合运算的几个位置



(f) 灰度闭运算结果

灰度开、闭运算示意图



9.4.2 开运算和闭运算

灰度形态学的基础运算

(2) 例程

■ 程序

```
Image=imread('maleman.gif');  
[h w]=size(Image);  
Image1=Image;    Image2=Image;  
k1=0.1;          k2=0.3;  
a1=rand(h,w)<k1;  a2=rand(h,w)<k2;  
Image1(a1&a2)=0;  Image2(a1&~a2)=255;  
figure,imshow(Image1);title('椒噪声图像');  
figure,imshow(Image2);title('盐噪声图像');
```



9.4.2 开运算和闭运算

灰度形态学的基础运算

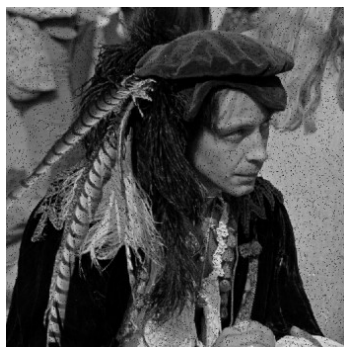
```
se=strel('disk',2);  
result1=imclose(Image1,se);  
result2=imopen(Image2,se);  
figure,imshow(result1);title('闭运算图像');  
figure,imshow(result2);title('开运算图像');
```



9.4.2 开运算和闭运算

灰度形态学的基础运算

■ 效果



(a) 椒噪声图像



(b) 闭运算结果



(c) 盐噪声图像



(d) 开运算结果

9.5 灰度图像的形态学处理



9.5.1 形态学平滑

9.5.2 形态学梯度

9.5.3 Top-hat和Bottom-hat



9.5.1 形态学平滑

灰度图像的形态学处理

(1) 定义

由先开后闭或先闭后开的形态学组合运算实现:

$$(f \circ b) \cdot b, (f \cdot b) \circ b$$

- 可以获得对图像的平滑处理，最终效果是可以去掉或减弱图像中特别亮的小亮斑和特别暗的小暗斑。
- 经灰度形态学平滑滤波后图像变得模糊。



9.5.1 形态学平滑

灰度图像的形态学处理

(2) 例程

■ 程序

```
Image=imread ('maleman.gif');  
Image1=imnoise(Image,'salt& pepper',0.04);  
se=strel('disk',2);  
result1=imopen(imclose(Image1,se),se);  
result2=imclose(imopen(Image1,se),se);  
figure,imshow(result1);title('先闭后开图像');  
figure,imshow(result2);title('先开后闭图像');
```



9.5.1 形态学平滑

灰度图像的形态学处理

■ 效果



(a) 椒盐噪声

先开后闭和先闭后开
均可去除椒盐噪声，
原因在于噪声点小于
结构元素。



(b) $f \bullet b \circ b$



(c) $f \circ b \bullet b$



9.5.2 形态学梯度

灰度图像的形态学处理

(1) 定义

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

经形态学图像梯度处理，灰度变化更加尖锐，且采用对称结构元素获得的形态学梯度比用Sobel算子较少受边缘方向影响。



边缘提取



形态学梯度





9.5.2 形态学梯度

灰度图像的形态学处理

(2) 例程---边缘检测

■ 程序

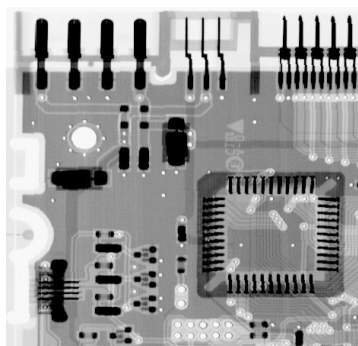
```
Image=imread('circard.jpg');  
SE=strel('disk',1);  
H1=[-1 -2 -1;0 0 0;1 2 1]; H2=[-1 0 1;-2 0 2;-1 0 1];  
R1=imfilter(Image,H1);  
R2=imfilter(Image,H2);  
result1=abs(R1)+abs(R2);  
figure,imshow(result1),title('Sobel图像');  
result2=imdilate(Image,SE)-imerode(Image,SE);  
figure,imshow(result2);
```



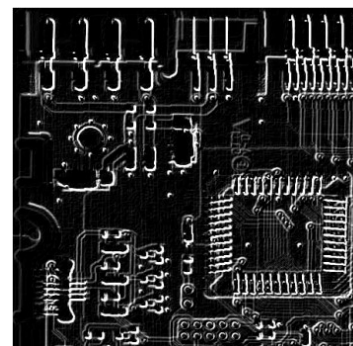
9.5.2 形态学梯度

灰度图像的形态学处理

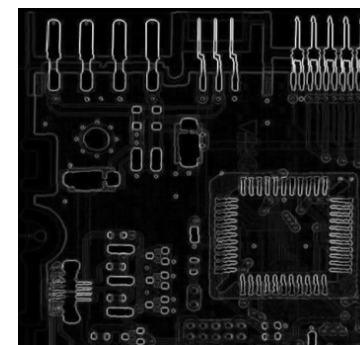
■ 效果



(a) 原图



(b) Sobel算子检测



(c) 形态学梯度算子检测



9.5.3 Top-hat和Bottom-hat变换

灰度图像的形态学处理

(1) 定义

- Top-hat变换: $TPH(f) = f - (f \circ b)$
- Bottom-hat变换: $BPH(f) = (f \cdot b) - f$
 - 它们都可以检测到图像中变化较大地方。
 - Top-hat(高帽)变换对在较暗背景中求亮的像素聚集体非常有效, 称波峰检测器。
 - Bottom-hat (低帽)变换对在较亮背景中求暗的像素聚集体非常有效, 称波谷检测器。



9.5.3 Top-hat和Bottom-hat变换

灰度图像的形态学处理

(2) 例程

■ 函数

□ 形态学Top-hat变换

- $IM2 = imtophat(IM, SE);$
- $IM2 = imtophat(IM, NHOOD);$

□ 形态学Bottom-hat变换

- $IM2 = imbothat(IM, SE);$
- $IM2 = imbothat(IM, NHOOD);$

■ 程序

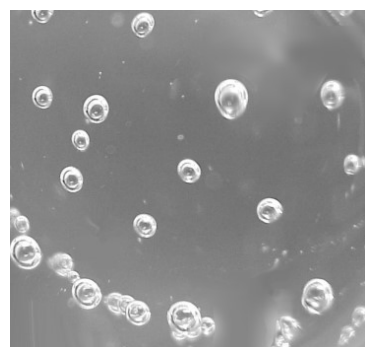
```
Image1=imread('cell.jpg');  
Image2=imread('clock.bmp');  
se=strel('disk',23); %选取半径为2圆盘结构元素  
result1=imtophat(Image1,se);  
result2=imbothat(Image2,se);  
figure,imshow(result1);title('Top-hat变换');  
figure,imshow(result2);title('Bottom-hat变换');  
rr1=imadjust(result1); %进行灰度线性拉伸  
rr2=imadjust(result2); %进行灰度线性拉伸  
figure,imshow(rr1);title('基于Top-hat对比度增强');  
figure,imshow(rr2);title('基于Bottom-hat对比度增强');
```



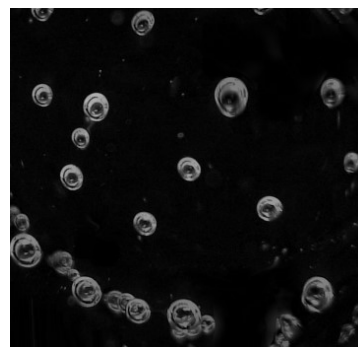
9.5.3 Top-hat和Bottom-hat变换

灰度图像的形态学处理

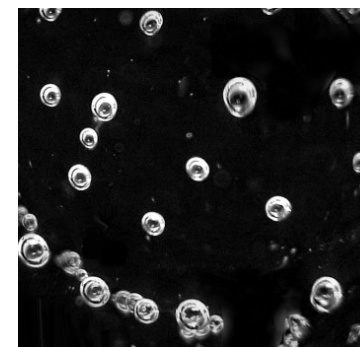
效果



(a) cell图像



(b) Top-hat变换



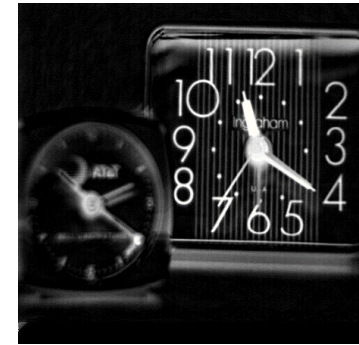
(c) 线性拉伸



(d) clock图像



(e) Bottom-hat变换



(f) 线性拉伸



编程实践

- 9.1 打开一幅灰度图像，设计方形结构元素，对其进行膨胀、腐蚀、开、闭运算。
- 9.2 打开一幅灰度图像，对其进行边缘检测，并利用数学形态学方法进行处理，以便获取完整边界。