

图象工程（中）

图 象 分 析

（第4版）

章毓晋

清华大学电子工程系 100084 北京

第4单元 数学工具

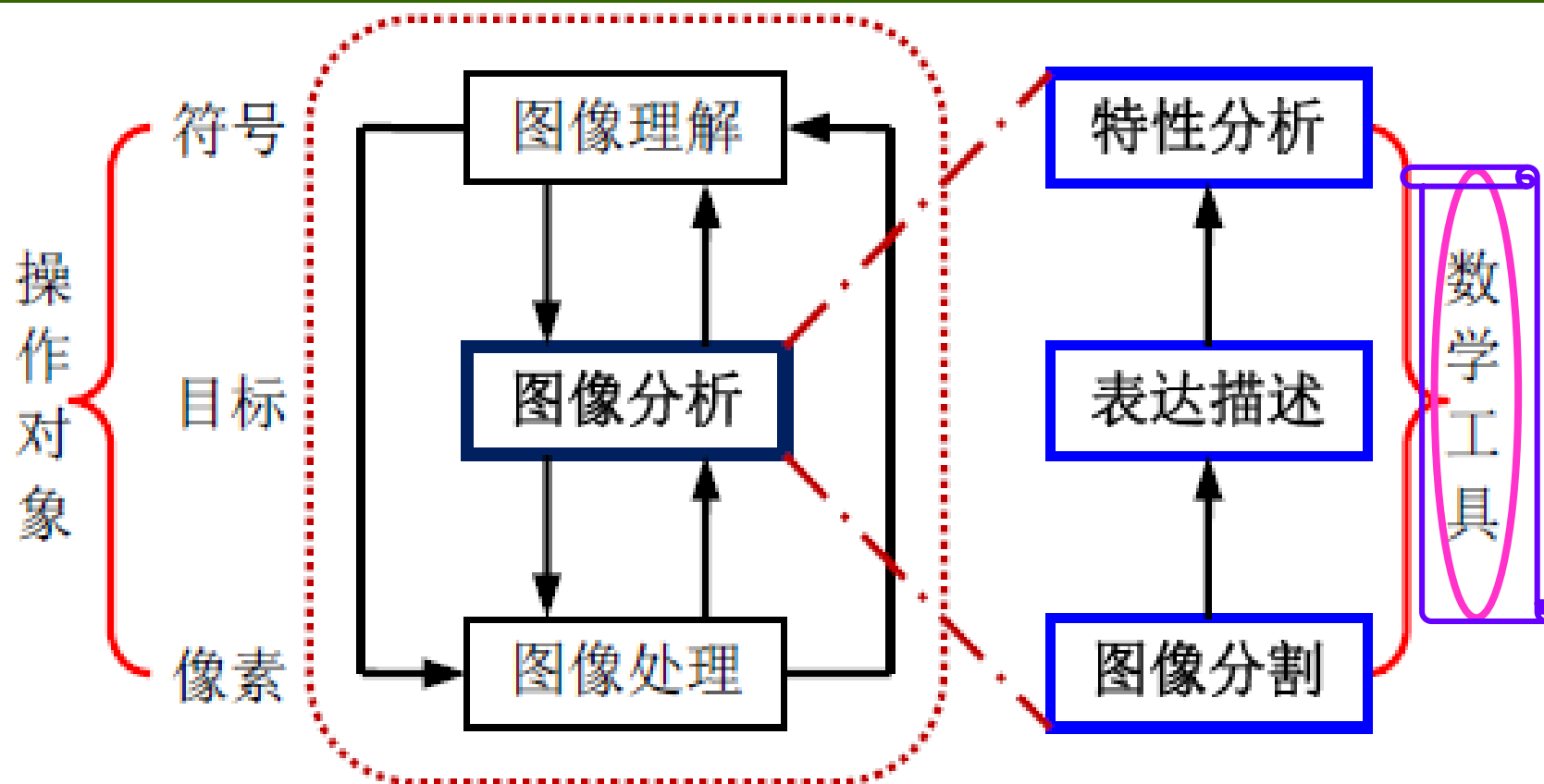
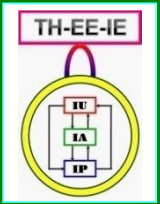


图 1.2.2 图像分析主要功能模块



第4单元 数学工具

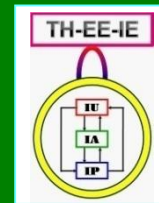
- 第13章 数学形态学：二值
- 第14章 数学形态学：灰度
- 第15章 图象识别

数学及其他学科理论和工具的支持

数学形态学表示以形态为基础对图象进行分析的数学工具

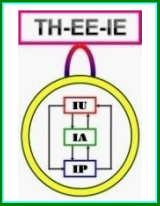
操作对象可以是二值或灰度图象

图象模式识别（简称图象识别）



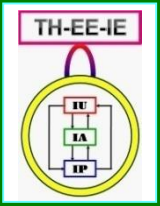
第14章 数学形态学：灰度

- 初期数学形态学方法仅可应用于二值图象
- 需将灰度图象先进行二值化
- 上世纪七十年代末八十年代初人们开始研究灰度形态学，使得数学形态学方法可应用于各种灰度图象（及彩色图象）
- 推广到更一般的概念——图象代数。图象代数可将各种形态学嵌在其内，而且还比形态学包含更多的集合运算和操作



第14章 数学形态学：灰度

- 14.1 灰度图象的排序
- 14.2 灰度形态学基本运算
- 14.3 灰度形态学组合运算
- 14.4 灰度形态学实用算法
- 14.5 图象代数



14.1 灰度图象的排序

- ✓ 对灰度图象讨论数学形态学的方法时不仅要考虑空间位置还要考虑灰度值的大小
- ✓ 以1-D的信号为例讨论（易推广到2-D图象）
- 一个信号 $f(x)$ 的支撑区/定义域为

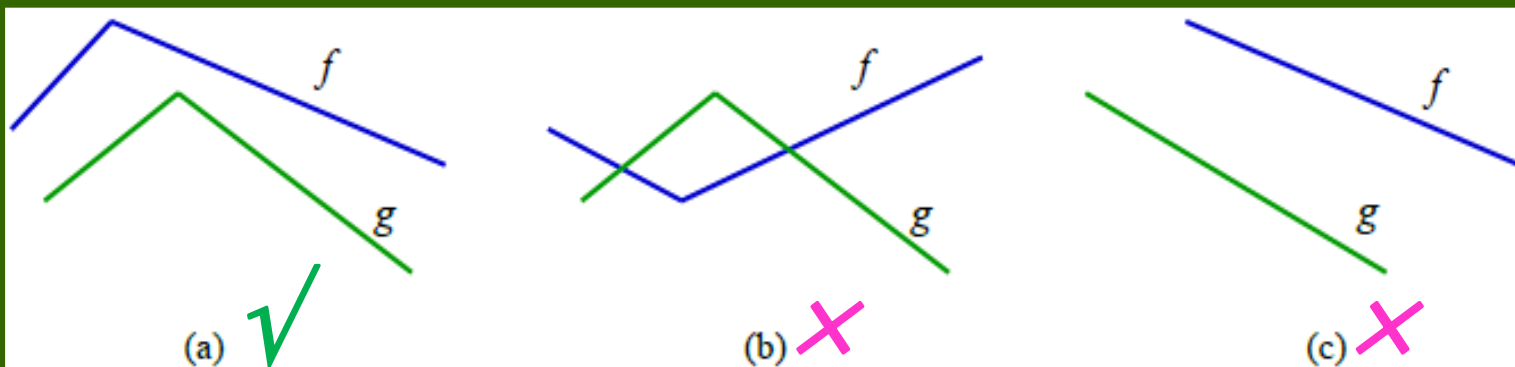
$$D[f] = \{x : f(x) > -\infty\}$$

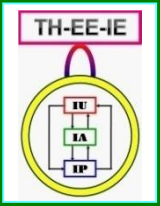
- ✓ 如果对所有的 x 都有 $g(x) \leq f(x)$ ，就说 $g(x)$ 在 $f(x)$ 的下方，并记为 $g \perp f$

14.1 灰度图象的排序

- ✓ 当且仅当 $D[g] \subset D[f]$ 且 x 属于两个信号的共同支撑区，即也有 $x \in D[g]$ 时，才能有 $g \perp f$
- 图(a)中 $g(x)$ 在不在 $f(x)$ 的下方？ $g \perp f$
- 图(b)中 $g(x)$ 在不在 $f(x)$ 的下方？ $g(x) > f(x)$
- 图(c)中 $g(x)$ 在不在 $f(x)$ 的下方？ $D[g] \not\subset D[f]$

例





14.1 灰度图象的排序

- ✓ 二值信号：交集和并集操作
- ✓ 灰度信号：最小和最大操作
- 两个信号 $f(x)$ 和 $g(x)$ 的最小值记为 $(f \wedge g)(x)$

$$(f \wedge g)(x) = \min \{f(x), g(x)\}$$

- 如果 $x \in D[f] \cap D[g]$ ，那么 $(f \wedge g)(x)$ 是 $f(x)$ 和 $g(x)$ 的最小值，否则 $(f \wedge g)(x) = -\infty$

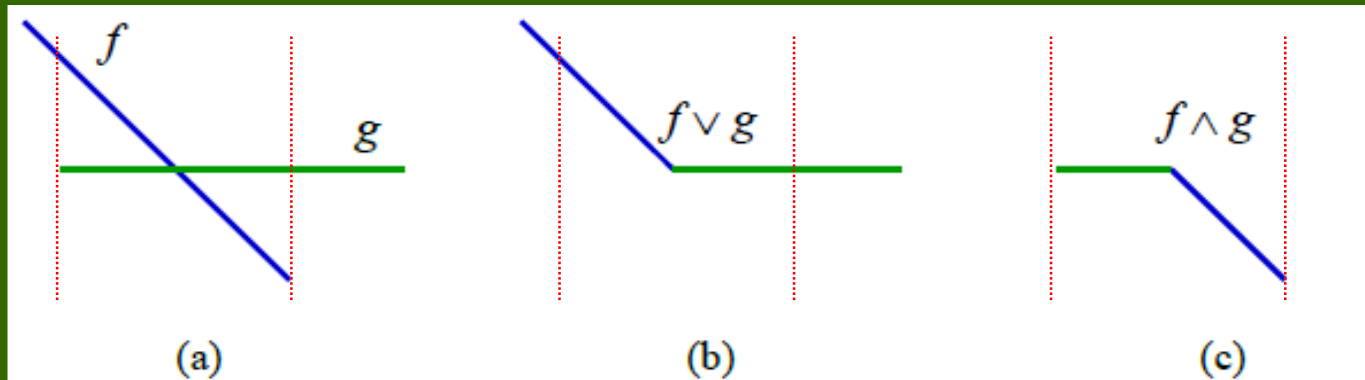
- 两个信号 $f(x)$ 和 $g(x)$ 的最大值记为 $(f \vee g)(x)$

$$(f \vee g)(x) = \max \{f(x), g(x)\}$$

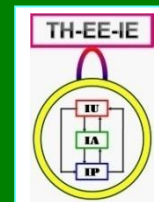
14.1 灰度图象的排序

- 如果 $x \in D[f] \cap D[g]$, 那么 $(f \vee g)(x)$ 是 $f(x)$ 和 $g(x)$ 的最大值, 否则 $(f \vee g)(x) = -\infty$
 - 如果 $x \in D[f] - D[g]$, 那么 $(f \vee g)(x) = f(x)$
 - 如果 $x \in D[g] - D[f]$, 那么 $(f \vee g)(x) = g(x)$
 - 如果 $x \notin D[f] \cup D[g]$, 那么 $(f \vee g)(x) = -\infty$
- } 对偶

例



➤ 反射 (reflection) / 映像: $\hat{f}(x, y) = f(-x, -y)$



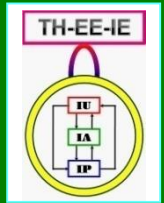
14.2 灰度形态学基本运算

这里运算的操作对象不再看作集合而看作图象
函数: $f(x, y)$ 是输入图象, $b(x, y)$ 是结构元素 (图象)

14.2.1 灰度膨胀和腐蚀

14.2.2 灰度开启和闭合

14.2.3 灰度基本运算性质



14.2.1 灰度膨胀和腐蚀

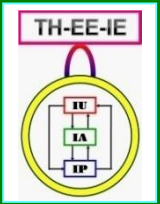
1. 灰度膨胀

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

D_f 和 D_b 分别是 f 和 b 的定义域

与2-D卷积的形式很类似，区别是这里用最大（运算）替换了卷积中的求和/积分（运算），用加法替换了卷积中的相乘

膨胀灰度图象的结果是：比背景亮的部分得到扩张，而比背景暗的部分受到收缩



14.2.1 灰度膨胀和腐蚀

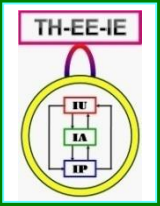
2. 灰度腐蚀

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

D_f 和 D_b 分别是 f 和 b 的定义域

与2-D相关的形式很类似，区别是这里用**最小**（运算）替换了相关中的求和/积分（运算），用**减法**替换了相关中的相乘

腐蚀灰度图象的结果是：比背景暗的部分得到扩张，而比背景亮的部分受到收缩



14.2.1 灰度膨胀和腐蚀

3. 灰度膨胀和腐蚀的对偶性

膨胀和腐蚀相对于函数的补（补函数）和映像也是对偶的

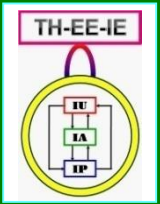
$$(f \oplus b)^c = f^c \ominus \hat{b}$$

$$(f \ominus b)^c = f^c \oplus \hat{b}$$

函数的补定义为 $f^c(x, y) = -f(x, y)$

函数的映像定义为 $\hat{b}(x, y) = b(-x, -y)$

{P. 346, 例14.2.6: 另一种推广方法}



14.2.2 灰度开启和闭合

- 用 b (灰度) 开启 f : $f \circ b = (f \ominus b) \oplus b$
- 用 b (灰度) 闭合 f : $f \bullet b = (f \oplus b) \ominus b$

灰度开启和闭合的对偶性

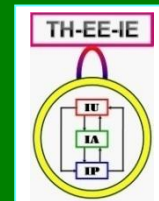
$$(f \circ b)^c = (f^c \bullet \hat{b})$$

$$(f \bullet b)^c = (f^c \circ \hat{b})$$

利用 $f^c(x, y) = -f(x, y)$

$$-(f \circ b) = (-f \bullet \hat{b})$$

$$-(f \bullet b) = (-f \circ \hat{b})$$



14.2.3 灰度基本运算性质

表 14.2.1 灰度数学形态学 4 种基本运算的性质 {对比表13.2.2}

性质 \ 运算	膨胀	腐蚀	开启	闭合
互换性	$f \oplus b = b \oplus f$			
组合性	$(f \oplus b) \oplus c = f \oplus (b \oplus c)$	$(f \ominus b) \ominus c = f \ominus (b \oplus c)$		
增长性	$f_1 \perp f_2 \Rightarrow f_1 \oplus b \perp f_2 \oplus b$	$f_1 \perp f_2 \Rightarrow f_1 \ominus b \perp f_2 \ominus b$	$f_1 \perp f_2 \Rightarrow f_1 \circ b \perp f_2 \circ b$	$f_1 \perp f_2 \Rightarrow f_1 \bullet b \perp f_2 \bullet b$
同前性			$(f \circ b) \circ b = f \circ b$	$(f \bullet b) \bullet b = f \bullet b$
外延性	$f \perp (f \oplus b)$			$f \perp (f \bullet b)$
非外延性		$(f \ominus b) \perp f$	$(f \circ b) \perp f$	

14.3 灰度形态学组合运算

1. 形态梯度

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

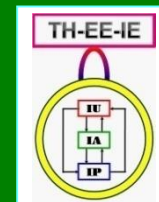
形态梯度能加强图象中比较尖锐的灰度过渡区

形态梯度



索贝尔梯度





14.3 灰度形态学组合运算

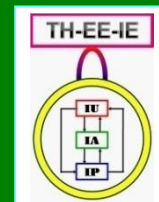
2. 形态平滑

先对图象开启然后再闭合（对照(13.4.1)）

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

这2种操作的综合效果
是去除或减弱亮区和
暗区的各类噪声
（对照图13.4.1）





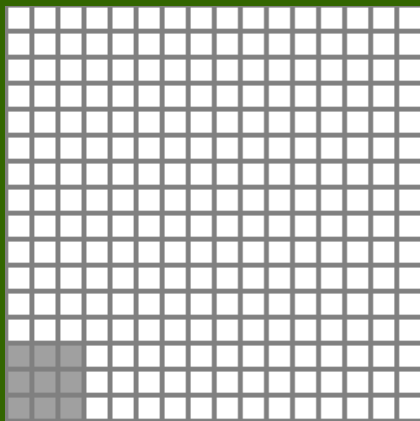
14.3 灰度形态学组合运算

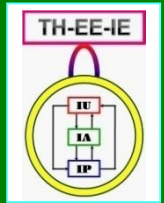
3. 高帽变换和低帽变换

高帽变换使用上部平坦的柱体或平行六面体（像1顶高帽）作为结构元素

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

可增强图象中亮区的细节（亮目标在暗背景上）





14.3 灰度形态学组合运算

3. 高帽变换和低帽变换

低帽变换使用下部平坦的柱体或平行六面体（类似将高帽的帽顶冲下放置）作为结构元素

实际中，仍可使用上部平坦的柱体或平行六面体（与高帽变换相同）作为结构元素，但将操作改为先用结构元素对原图象进行闭合，再从结果中减去原图象

$$T_b = (f \bullet b) - f$$

可增强图象中暗区的细节（暗目标在亮背景上）

14.3 灰度形态学组合运算

4. 形态滤波器

非线性信号滤波器，它通过形态变换来局部地修改信号的几何特征

开启有些像非线性低通滤波器，但只允许大的结构通过而能去除小的结构

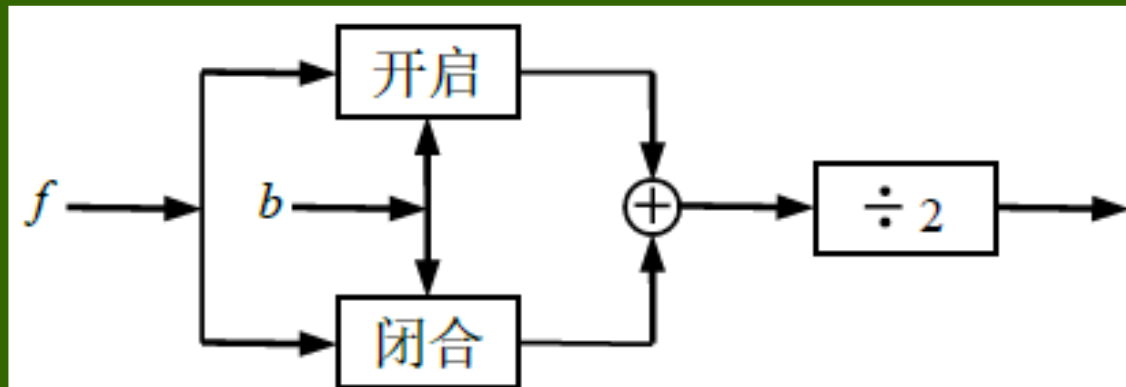
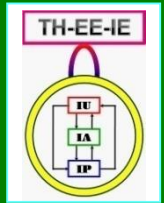


图 14.3.4 开启和闭合结合的混合滤波器



14.3 灰度形态学组合运算

4. 形态滤波器

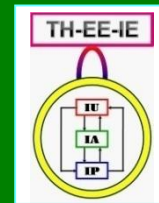
筛滤波器：一种允许尺寸在某个窄范围内的结构通过的形态滤波器

例：提取尺寸为 $n \times n$ 个像素的亮点状缺陷

$$(f \circ b^{n \times n}) - (f \circ b^{(n-2) \times (n-2)})$$

第1项将尺寸小于 $n \times n$ 的所有亮结构除去，第2项将尺寸小于 $(n-2) \times (n-2)$ 的所有亮结构除去

将上两项相减，得到的结果中留下了其尺寸在 $n \times n$ 和 $(n-2) \times (n-2)$ 之间的亮结构



14.3 灰度形态学组合运算

5. 软形态滤波器

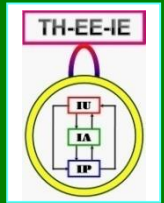
对加性噪声比较不敏感，对需滤波目标的形状变化也比较不敏感

可以在加权排序统计的基础上定义

用结构系统替换结构元素

集合 B 中
元素个数

结构系统 $[B, C, r]$ 包括3个参数：有限平面集合 B 和 C ， $C \subset B$ ，及满足 $1 \leq r \leq |B|$ 的自然数 r 。集合 B 称为结构集合， C 是它的（硬）中心， $B - C$ 给出它的（软）轮廓，而 r 是它的中心的阶数



14.3 灰度形态学组合运算

5. 软形态滤波器

软膨胀

$$f \oplus [B, C, r](x) =$$

复合集 $\{r \diamond f(c) : c \in C_x\} \cup \{f(b) : b \in (B - C)_x\}$ 的第 r 个最大值

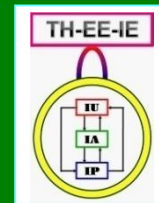


重复操作: $\{1, 1, 1, 2, 3, 3\} = \{3 \diamond 1, 2, 2 \diamond 3\}$ 复合集

软腐蚀

$$f \ominus [B, C, r](x) =$$

复合集 $\{r \diamond f(c) : c \in C_x\} \cup \{f(b) : b \in (B - C)_x\}$ 的第 r 个最小值



14.4 灰度形态学实用算法

1. 背景估计和消除

- 开启操作可将比背景亮且比结构元素尺寸小的区域除去
- 选取合适的结构元素进行开启可使图象中仅剩下对背景的估计
- 从原始图中减去对背景的估计就可将目标提取出来

$$\text{背景估计} = f \circ b$$

$$\text{背景消除} = f - (f \circ b)$$

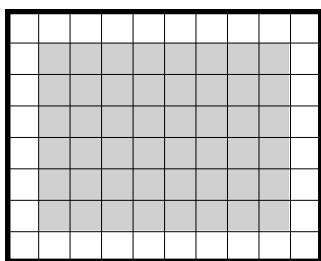
14.4 灰度形态学实用算法

2. 边缘的形态检测

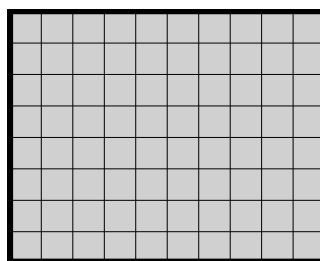
形态边缘检测器不会加强或放大噪声

- 基本的形态梯度

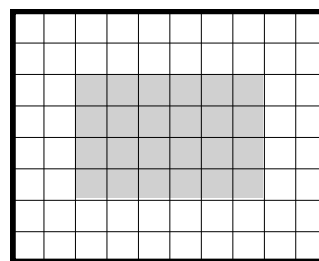
$$\text{grad}_1 = (f \oplus b) - (f \ominus b)$$



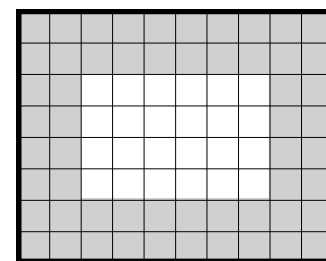
(a)



(b)



(c)



(d)

! 给出的边界有两个像素宽

14.4 灰度形态学实用算法

2. 边缘的形态检测

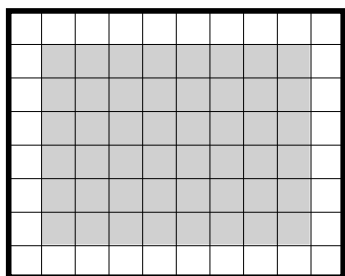
- 获得较细边界的形态梯度

$$\text{grad}_2 = (f \oplus b) - f$$

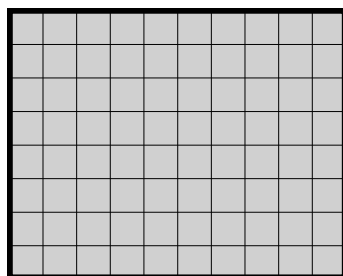
$$\text{grad}_2 = f - (f \ominus b)$$

连续等价

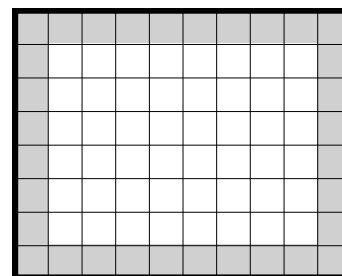
离散不等价



(a)

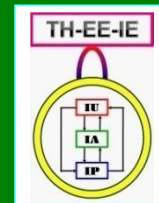


(b)



(c)

第1式/第2式? 另一式的图什么样?



14.4 灰度形态学实用算法

2. 边缘的形态检测

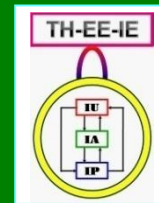
- 对孤立噪声点不敏感的形态梯度

$$\text{grad}_3 = \min\{[(f \oplus b) - f], [f - (f \ominus b)]\}$$

- 适用于理想斜面边缘
- 检测不出理想阶梯边缘
- 基于模糊图象的形态梯度（4-邻域圆柱模板）

$$h(i, j) = [f(i, j) + f(i+1, j) + f(i, j+1) + f(i-1, j) + f(i, j-1)]/5$$

$$\text{grad}_4 = \min\{[(h \oplus b) - h], [h - (h \ominus b)]\}$$



14.4 灰度形态学实用算法

3. 聚类快速分割

- 条件膨胀

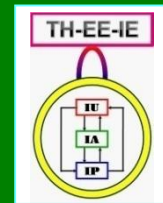
在条件 X 的情况下用 b 膨胀 f

$$f \oplus b; X = (f \oplus b) \cap X$$

- 反复条件膨胀

$$f \oplus \{b\}; X = [[(f \oplus b) \cap X] \oplus b] \cap X] \oplus b \dots$$

$\{b\}$ 代表迭代地用 b 膨胀 f 直到不再有变化



14.4 灰度形态学实用算法

3. 聚类快速分割

- 最终腐蚀

重复腐蚀一个目标直到它消失，此时保留这之前最后一步的结果（也称为目标的种子）

令 $f_k = f \ominus kb$ ， b 是单位圆， kb 是半径为 k 的圆

- 三个步骤：

$$U_k = (f_{k+1} \oplus \{b\}); f_k$$

$$g_k = f_k - U_k$$

$$g = \bigcup_{k=1,m} g_k$$

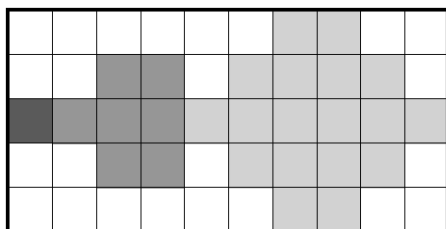
（最终目标集合）

14.4 灰度形态学实用算法

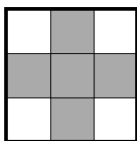
3. 聚类快速分割

(1) 用单位圆形结构元素 b 迭代地腐蚀原始图象 f

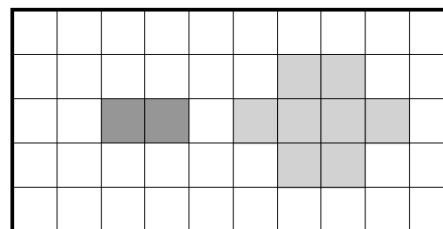
$$f_k = f \ominus kb \quad k=1, \dots, m \quad \text{其中} \quad \{m: f_m \neq \emptyset\}$$



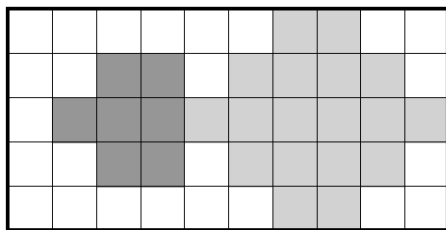
(a)



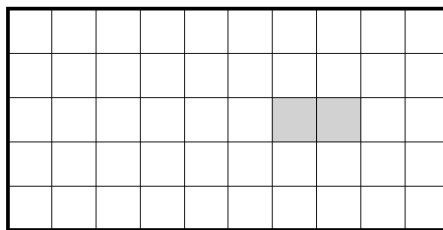
(b)



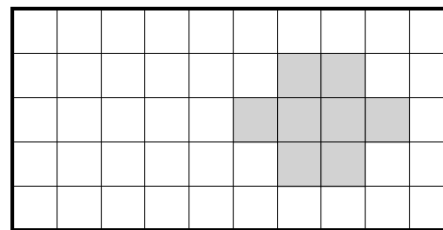
(c)



(d)



(e)

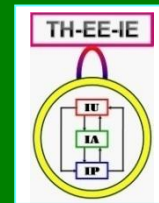


(f)

$$S_1 = (a) - (b)$$

$$S_2 = (c) - (b)$$

$$S_3 = (e)$$



14.4 灰度形态学实用算法

3. 聚类快速分割

(2) 最终腐蚀各个 f_k 并从 f_k 中减去腐蚀的结果

$$g_k = f_k - (f_k \ominus \{b\}; f_k)$$

用图(a)限定并用 b 迭代地膨胀图(c)就可得到图(d)
将图(d)从图(a)中减去, 得到第一个种子,

(3) 从各个种子出发借助下式可恢复回原来各区域的完整尺寸

$$U = \bigcup g_k \oplus (k-1)b \quad k=1 \text{ 到 } m$$

14.4 灰度形态学实用算法

4. 水线分割

(1) 产生距离图

将较大的值赋予较远离背景的目标象素（与距离成正比）就可得到一幅距离图

为用形态学方法产生距离图，可迭代地腐蚀二值图，在每次腐蚀后将所有剩下象素的值加1

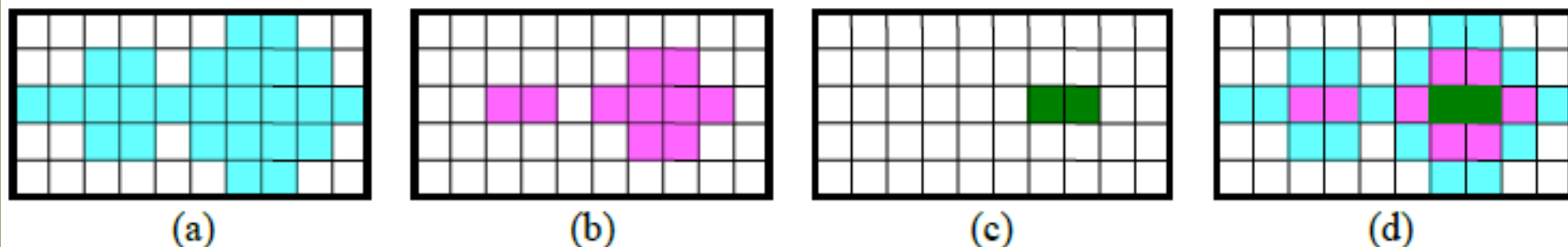


图 14.4.5 距离图计算示例

14.4 灰度形态学实用算法

4. 水线分割

- (2) 计算最终腐蚀集合 Y_k (与聚类快速分割相同)
- (3) 从种子开始生长并恢复原尺寸但不使各区域相连 (使用条件粗化, 12个结构元素)

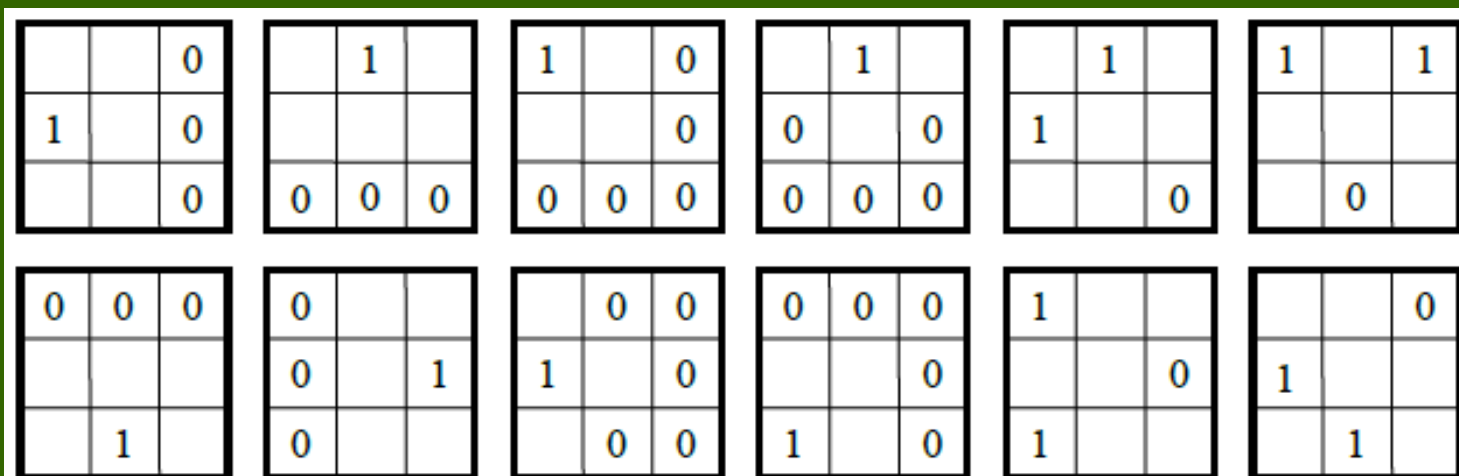


图 14.4.6 用于水线算法的结构元素

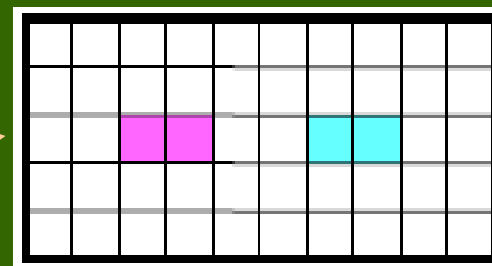
14.4 灰度形态学实用算法

4. 水线分割

种子生长示例

① 将最后两个种子结合

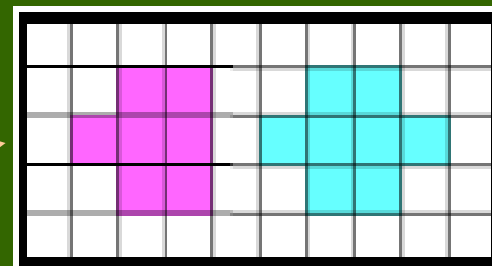
(使用击中击不中变换)



② 将结构元素的零象素与

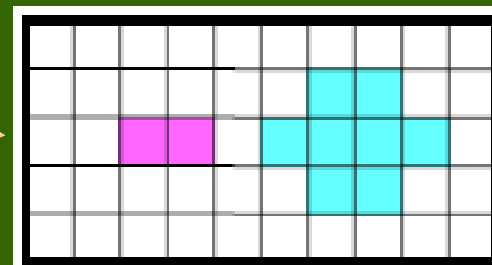
水线运算的结果匹配

以避免生长的区域相连



③ 求水线运算的结果和

一幅条件图象的交集



14.4 灰度形态学实用算法

5. 纹理分割

(b)先用一系列逐步增大的圆形结构元素依次去闭合原始图，(c)再选用一个比大圆之间的间隙要大的结构元素进行一次开启 $\Rightarrow \Rightarrow \Rightarrow$ 灰度阈值化

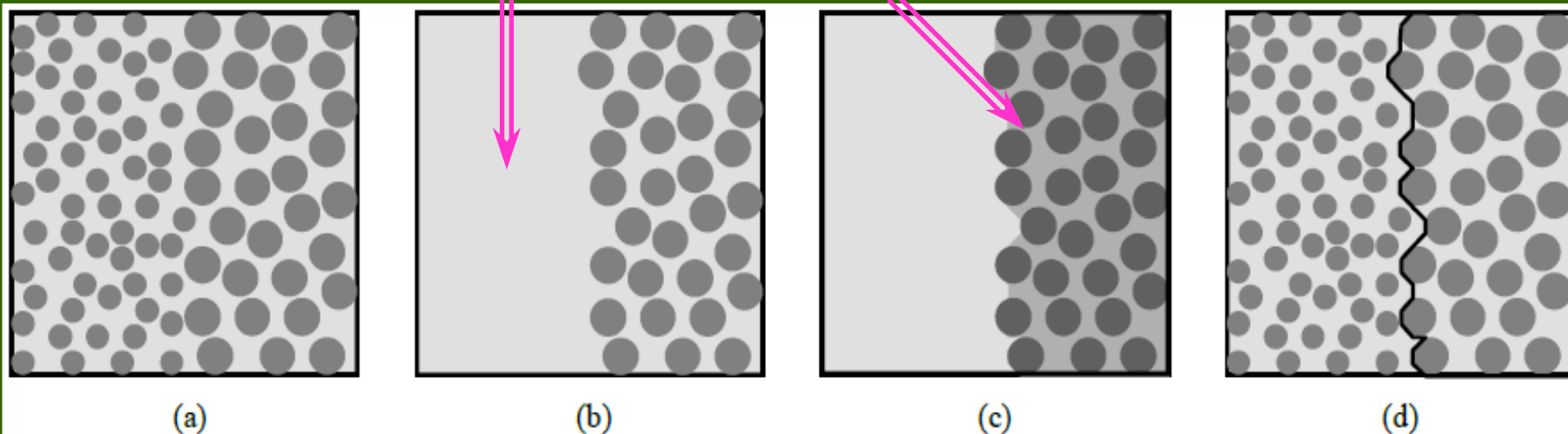
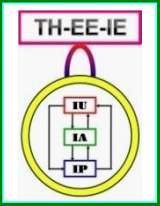
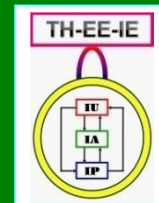


图 14.4.8 形态学纹理分割示例



14.5 图象代数

- ✓ 图象代数是一个比较大的概念
= 数学形态学 + 卷积一类的线性处理技术
- ✓ 图象代数可看作是一种为表示图象处理算法而设计的高级语言，提供通用的数学环境
- ✓ 已经证明用图象代数编写任何图象处理程序在理论上都是可能的
- ✓ 图象代数本质上是并行的
- ✓ 图象代数的一个不足之处是，它的表达方法对人们直观理解算法的帮助不大？



14.5 图象代数

1. 图象运算

- 记 \mathbf{R} 为实数集合, \mathbf{X} 为图象的定义域
- 一幅实值图象 a 定义为:

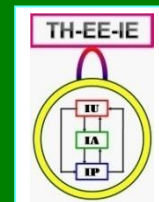
$$a = \{[x, a(x)]: x \in \mathbf{X}\}$$

- a 和 b 是具有相同定义域 \mathbf{X} 的两幅实图象

$$c = a + b = \{[x, c(x)]: c(x) = a(x) + b(x), x \in \mathbf{X}\}$$

$$c = a * b = \{[x, c(x)]: c(x) = a(x) * b(x), x \in \mathbf{X}\}$$

$$c = a \vee b = \{[x, c(x)]: c(x) = a(x) \vee b(x), x \in \mathbf{X}\}$$



14.5 图象代数

1. 图象运算

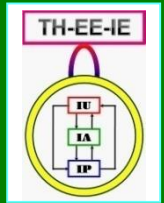
- 任何定义在 \mathbf{R} 上的函数都可用实图象作为其变量导出，导出函数可用对图象值进行点操作实现

$$c = f(a) = \{[x, c(x)]: c(x) = f[a(x)], x \in X\}$$

✓ 特性函数

- 将特性函数用于一幅实图象产生一幅二值图象（这是一个取阈值操作）

$$T^S(a) = \{[x, c(x)]: c(x) = 1, \text{ 如果 } a(x) \in S, \text{ 否则 } c(x) = 0\}$$



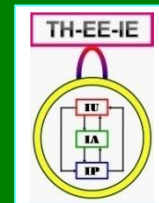
14.5 图象代数

2. 模板与形态结构元素

- 模板结合了一般常说的模板、掩模、窗以及邻域操作，所以它的概念与数学形态学中的结构元素有所不同
- 在数学形态学中原始图象和结果图象的定义域是相同的，而在图象代数中并不一定相同
- 模板是一组图象

$$t_y = \{[x, t_y(x)]: x \in X\}$$

{图14.5.1}



14.5 图象代数

3. 图象和模板间的操作

实模板的支撑

$$Z(t_y) = \{x \in X : t_y(x) \neq 0\}$$

- 广义卷积

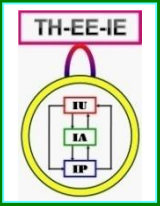
$$a \otimes t = \{[y, b(y)] : b(y) = \sum_{x \in Z(t_y)} a(x) \times t_y(x), \quad y \in Y\}$$

- 加性极大

$$a \uparrow t = \{[y, b(y)] : b(y) = \max_{x \in Z(t_y)} [(a(x) + t_y(x)) : x \in X], \quad y \in Y\}$$

- 乘性极大

$$a \uparrow\uparrow t = \{[y, b(y)] : b(y) = \max_{x \in Z(t_y)} [(a(x) \times t_y(x)) : x \in X], \quad y \in Y\}$$



14.5 图象代数

4. 图象代数算法

- 局部平均

$$a \otimes t$$

$$c(y) = \frac{1}{n \times n} \sum_{x \in Z(t_y)} a(x)$$

- 膨胀和腐蚀

$$a \uparrow t$$

$$a \downarrow t$$

- 开启和闭合

$$(a \downarrow t^*) \uparrow t$$

$$(a \uparrow t) \downarrow t^*$$

对偶

- 击中击不中变换

$$(a \downarrow t) (a^c \downarrow t^*)$$



联系信息

- ☞ 通信地址：北京清华大学电子工程系
- ☞ 邮政编码：100084
- ☞ 办公地址：清华大学，罗姆楼，6层305室
- ☞ 办公电话：(010) 62798540
- ☞ 传真号码：(010) 62770317
- ☞ 电子邮件：zhang-yj@tsinghua.edu.cn
- ☞ 个人主页：oa.ee.tsinghua.edu.cn/~zhangyujin/