

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1单元 图象分割

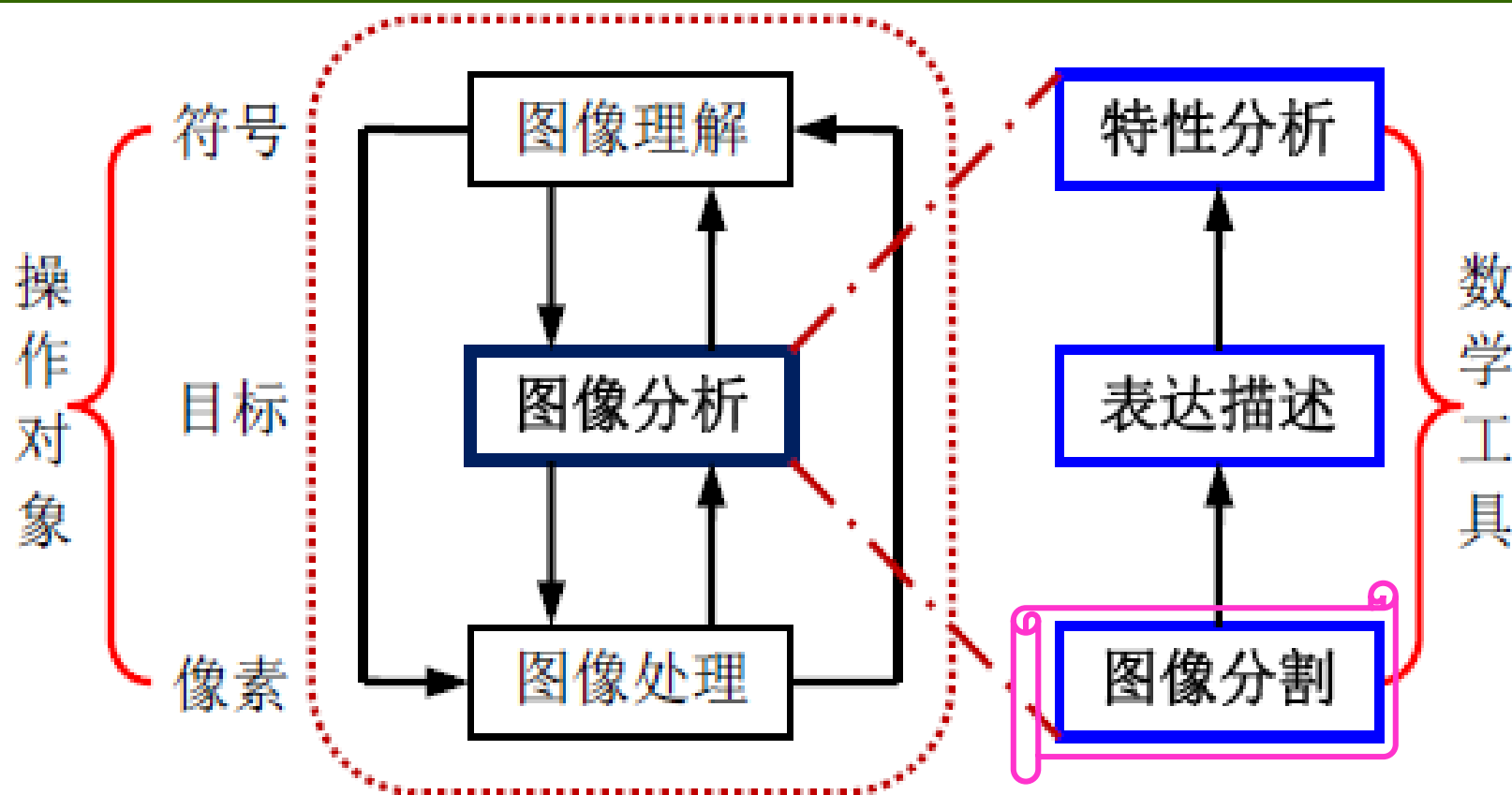
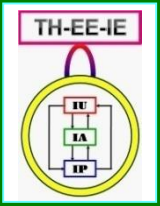


图 1.2.2 图像分析主要功能模块

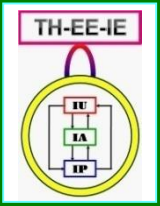


# 第1单元 图象分割

- 第2章 图象分割基础
- 第3章 典型分割算法
- 第4章 分割技术扩展
- 第5章 分割评价比较

图象分割是由图象处理进到图象分析的关键步骤

很多时候关注的仅是图象中的目标或前景（其他部分称为背景），它们一般对应图象中特定的、具有独特性质和含义的区域



# 第3章 典型分割算法

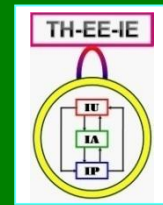
分别对应四类技术

3.1 兴趣点检测

3.2 图割方法

3.3 特色的阈值化和聚类技术

3.4 分水岭分割算法



## 3.1 兴趣点检测

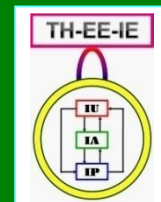
泛指图象中或目标上具有特定几何性质或属性性质的点

① 并行边界类

3.1.1 二阶导数检测角点

3.1.2 最小核同值区算子

3.1.3 哈里斯兴趣点算子



## 3.1.1 二阶导数检测角点

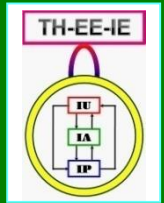
### 角点

- 在其小邻域中具有两个明显不同的边缘的点
  - 局部（离散）曲率较大的边缘点
- 由各个二阶导数组成的对称矩阵

$$\mathbf{I}_{(2)} = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}, \quad I_{xy} = I_{yx}$$

旋转坐标系

$$\tilde{\mathbf{I}}_{(2)} = \begin{bmatrix} I_{\tilde{x}\tilde{x}} & 0 \\ 0 & I_{\tilde{y}\tilde{y}} \end{bmatrix} = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix} \quad \text{主曲率}$$



## 3.1.1 二阶导数检测角点

二阶导数矩阵 $I_{(2)}$ 的秩和行列式都旋转不变

拉普拉斯值：对边缘和直线都给出较强响应

$$\text{Laplacian} = I_{xx} + I_{yy} = K_1 + K_2$$

海森值：对边缘和直线没有响应，但是在角点的邻域有较强的响应

$$\text{Hessian} = \det(I_{(2)}) = I_{xx}I_{yy} - I_{xy}^2 = K_1K_2$$

$$C = Kg = K\sqrt{I_x^2 + I_y^2} = \frac{I_{xx}I_y^2 - 2I_{xy}I_xI_y + I_{yy}I_x^2}{I_x^2 + I_y^2}$$

## 3.1.2 最小核同值区算子

### 1、核同值区

**USAN:** Univalue Segment Assimilating Nucleus  
核同值区：相对于模板的核，模板中有一定的区域与它有相同的灰度

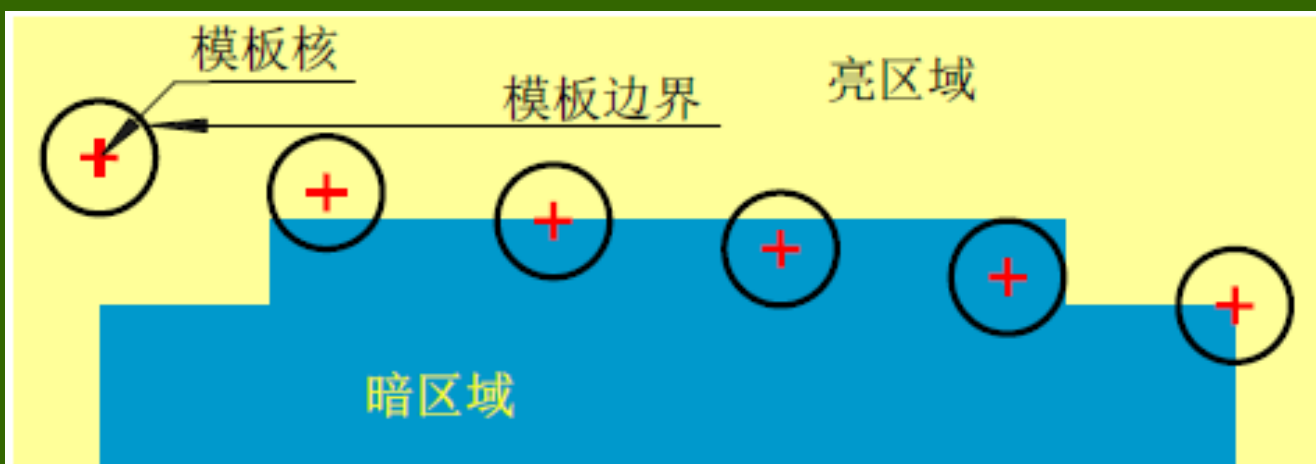
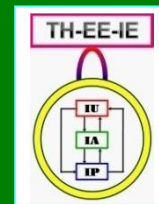


图 3.1.1 圆形模板在图像中的不同位置



## 3.1.2 最小核同值区算子

### 1、核同值区

- USAN的面积携带了关于图象中核像素处结构的主要信息
- 当核像素处在图象中的灰度一致区域中时，USAN的面积会达到最大。该面积当核处在直线边缘处约为最大值的一半，而当核处在角点处则为最大值的1/4
- 使用USAN面积作为特征起到了增强边缘和角点的效果

## 3.1.2 最小核同值区算子

### 2、最小核同值区算子角点检测

**SUSAN:** 最小 (Smallest) 核同值区

检测模板:

输出响应

$$C(x_0, y_0; x, y) = \begin{cases} 1 & \text{如果 } |f(x_0, y_0) - f(x, y)| \leq T \\ 0 & \text{如果 } |f(x_0, y_0) - f(x, y)| > T \end{cases}$$

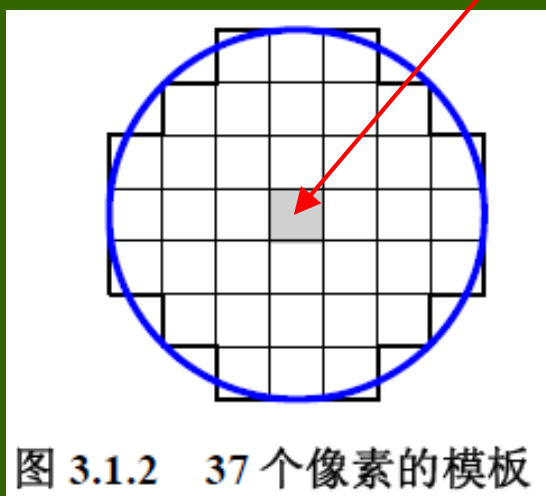


图 3.1.2 37 个像素的模板

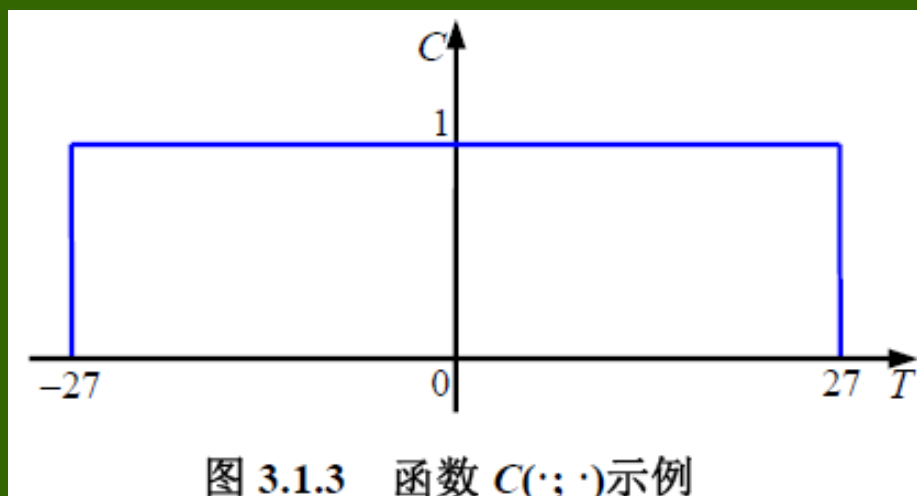
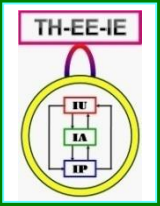


图 3.1.3 函数  $C(\cdot; \cdot)$  示例



## 3.1.2 最小核同值区算子

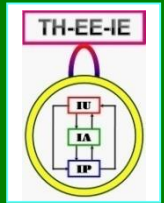
- 检测对模板中的每个像素  $f(x, y)$  进行
- 得到输出的游程和 (running total)

$$S(x_0, y_0) = \sum_{(x, y) \in N(x_0, y_0)} C(x_0, y_0; x, y)$$

- 边缘响应

$$R(x_0, y_0) = \begin{cases} G - S(x_0, y_0) & \text{如果 } S(x_0, y_0) < G \\ 0 & \text{否则} \end{cases}$$

几何阈值  $G = 3S_{\max}/4$ , 其中  $S_{\max}$  是  $S$  所能取的最大值



## 3.1.3 哈里斯兴趣点算子

### 哈里斯矩阵

$$\mathbf{H} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

#### 1、角点检测

计算像素邻域中灰度值平方差的和来检测角点

角点强度

$$C = \frac{\det(\mathbf{H})}{\text{trace}(\mathbf{H})} \quad (\text{不受旋转影响})$$

锐角角点

$$\mathbf{H} = \begin{bmatrix} l_2 g^2 \sin^2 \theta & l_2 g^2 \sin \theta \cos \theta \\ l_2 g^2 \sin \theta \cos \theta & l_2 g^2 \cos^2 \theta + l_1 g^2 \end{bmatrix}$$

# 3.1.3 哈里斯兴趣点算子

## 1、角点检测

角点强度

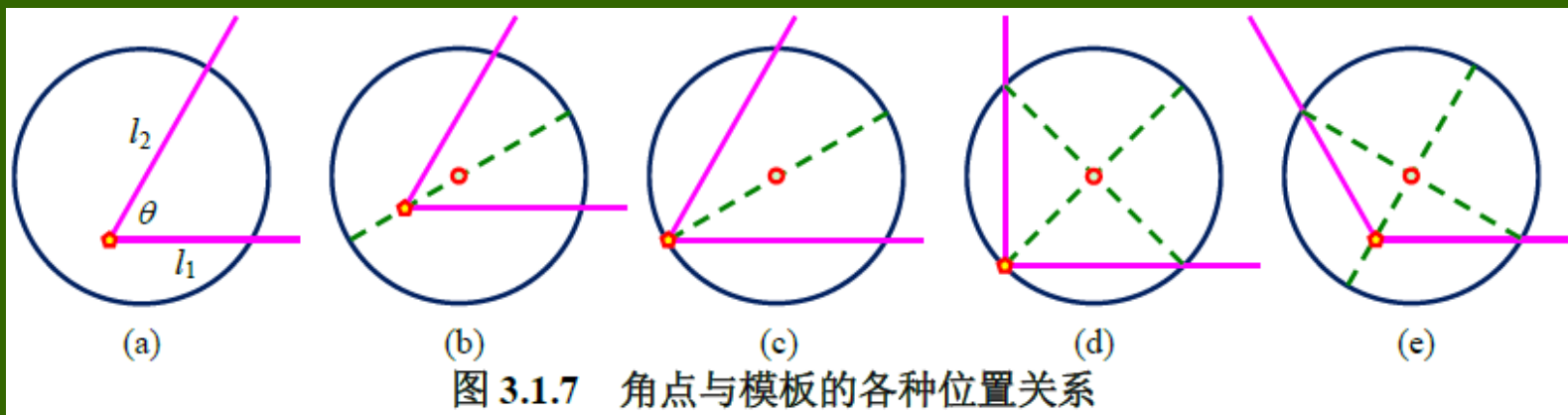
$$C = \frac{l_1 l_2}{l_1 + l_2} g^2 \sin^2 \theta$$

强度因子

对比度因子

形状因子

角点落在角的中分线（也是直径）上



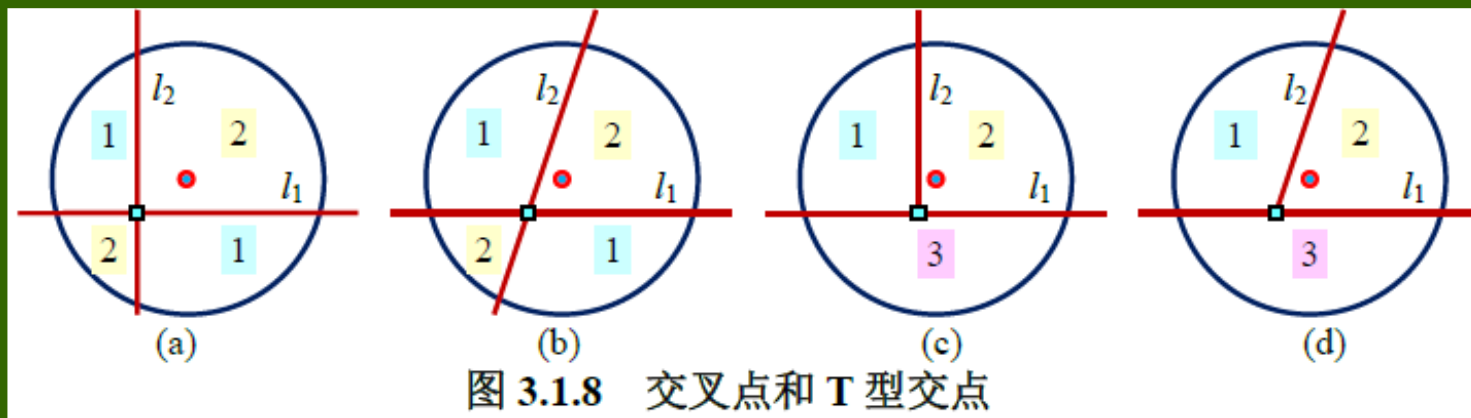
# 3.1.3 哈里斯兴趣点算子

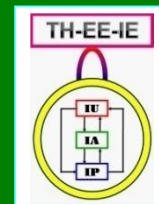
## 2、交叉点检测

互相垂直或不垂直的直线的交点

$$C = \frac{l_1 l_2 g_1^2 g_2^2}{l_1 g_1^2 + l_2 g_2^2} \sin^2 \theta$$

T型交点可以看作是比角点和交叉点更一般的兴趣点，因为它涉及3个具有不同灰度的区域





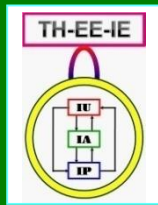
## 3.2 图割方法

### 基于图论的图象分割技术

### ② 串行边界类

本质上采用了基于边缘的串行分割思路

- (1) 将待分割图象 $I$ 映射为一个对弧加权的有向图 $G$ ， $G$ 在尺寸上和维数上都与 $I$ 对应
- (2) 确定目标和背景的种子像素（常交互进行），并针对它们构建两个特殊的图结点，即源结点 $s$ 和汇结点 $t$ ；然后将所有结点根据它们的目标或背景标号分别与源结点或汇结点相连接



## 3.2 图割方法

### 基于图论的图象分割技术

(3) 计算弧代价函数，并对图 $G$ 中的各个弧赋予一定的弧代价（与边缘强度有关）

(4) 使用最大流图优化算法来确定对图 $G$ 的图割，从而区分对应目标和背景象素的结点

图割方法的一个重要特性是提供了一种借助交互方法以有效改进先前获得的（串行）分割结果的能力

## 3.2 图割方法

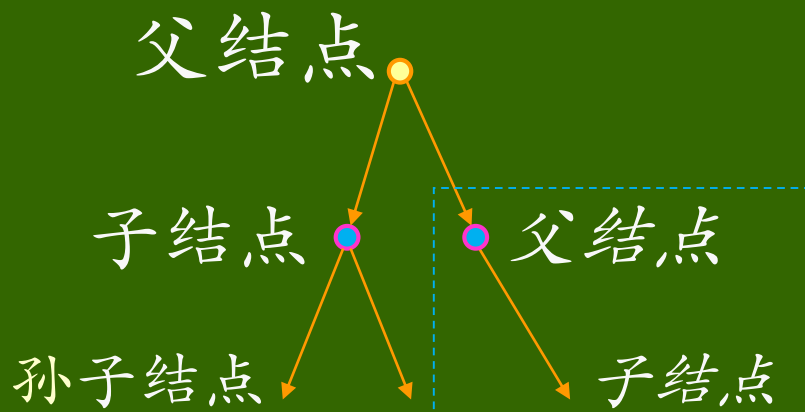
### 1. 构建有向图 $G$

将边缘像素和边界段（弧）用图表示

图:  $G = \{N, A\}$

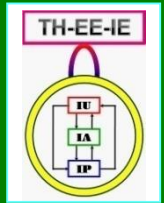
结点集  $\{n_1, \dots\}$

结点对集  $\{(n_i, n_j)\}$



通路代价

$$C = \sum_{i=2}^K c(n_{i-1}, n_i)$$



## 3.2 图割方法

### 1. 构建有向图 $G$

对任一段弧 $(n_i, n_j)$ 都可定义一个代价（或费用），记为 $C(n_i, n_j)$ ，它可看作是对弧的加权

对给定的待分割图象 $I$ ，要将其转化表示为一个对弧加权的图 $G$ 。其中，将图象 $I$ 中每个像素看成图 $G$ 中的一个结点，即结点集合 $N$ 由所有像素构成；而将像素间的邻接关系用图 $G$ 中的弧来表示，即结点对集合 $A$ 表示像素间的（加权）联系

## 3.2 图割方法

### 2. 图割分割

目标种子  $o$   
背景种子  $b$

源  $S \Leftrightarrow$  汇  $T$   
图  $G_{st}$

图  $G_{st}$  中的  $s-t$  割

分割结果

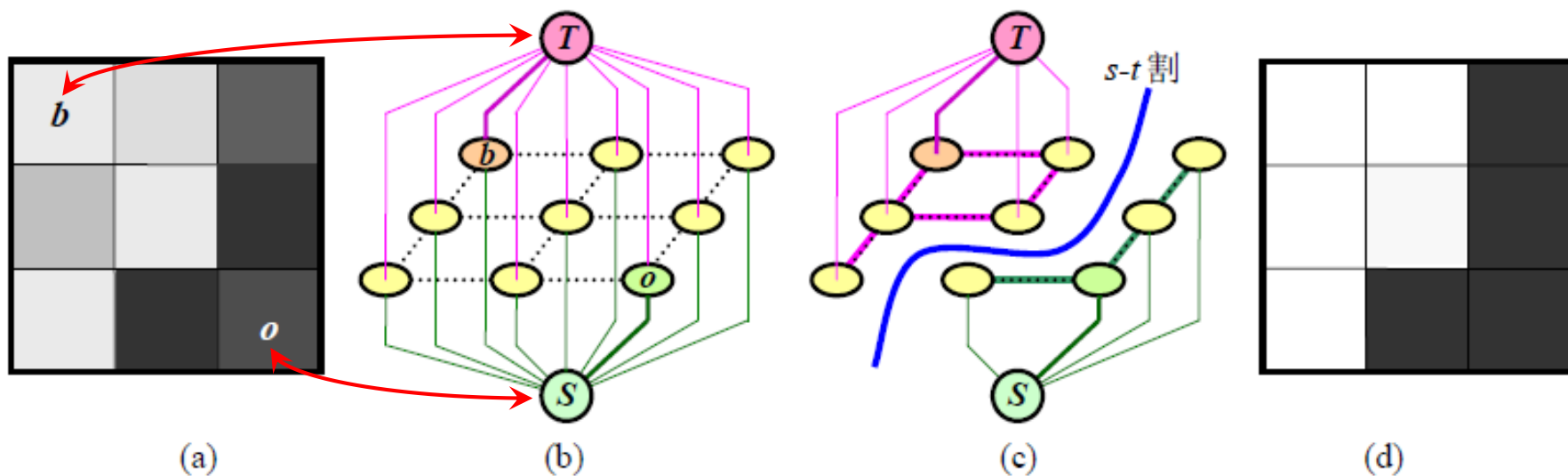
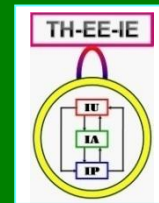


图 3.2.1 图割分割示意



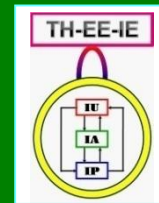
## 3.2 图割方法

### 2. 图割分割

一个弧加权图  $G_{st} = [N \cup \{s, t\}, A]$ , 结点集  $N$  对应图象  $I$  中的像素,  $s$  和  $t$  是两个特殊的终端结点

弧集合  $A$  中的元素可分为两类: 连接一对相邻像素的弧与将像素和终端结点连接起来的弧

在  $G_{st}$  中的一个割将图中结点分成两组, 它的代价是这个割所对应的弧 (割所穿过/跨越的弧) 的代价之和。代价最小的割称为最小  $s$ - $t$  割, 它将结点分成两组不重叠的子集  $S$  ( $s \in S$ ) 和  $T$  ( $t \in T$ ), 且从  $s$  到  $t$  没有有向的通路



## 3.2 图割方法

### 3. 弧的代价

$$C(L) = \lambda R(L) + F(L)$$

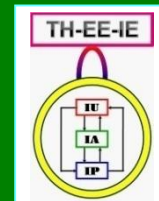
弧两个端结点所  
对应像素的灰度

$$R(L) = \sum_{p \in I} R_p(L_p)$$

弧两个端结点所对  
应像素间的灰度差

$$F(L) = \sum_{(p,q) \in A} F_{(p,q)} \delta(L_p, L_q)$$

$$\delta(L_p, L_q) = \begin{cases} 1 & L_p \neq L_q \\ 0 & \text{其他} \end{cases}$$



## 3.2 图割方法

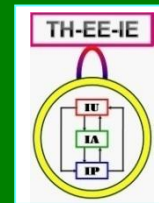
### 3. 弧的代价

先考虑连接一对相邻像素的弧

再考虑将像素和终端结点连接起来的弧

表 3.2.1 各种弧的代价函数

弧	$(p, q)$	$(s, p)$	$(p, t)$
代价	$C_{(p, q)} \quad (p, q) \in N$	$\lambda R_p(b) \quad p \in I, p \notin (O \cup B)$ $K \quad p \in O$ $0 \quad p \in B$	$\lambda R_p(o) \quad p \in I, p \notin (O \cup B)$ $0 \quad p \in O$ $K \quad p \in B$



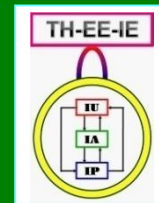
## 3.2 图割方法

### 4. 最小 $s-t$ 割的计算

对偶：计算最小 $s-t$ 割  $\Leftrightarrow$  计算最大流  
最大流的值等于最小割的代价

#### ➤ 增强通路算法

考虑推动从源 $s$ 到汇 $t$ 的流直至达到最大流  
开始时将流的状态初始化为0（保留在 $G_r$ 中）  
沿未饱和的弧来确定最短的 $s \rightarrow t$  通路  
借助推动最大可能的流而使得弧达到饱和



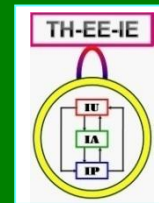
## 3.3 特色的阈值化和聚类技术

### ③ 并行区域类

3.3.1 多分辨率阈值选取

3.3.2 借助过渡区选择阈值

3.3.3 借助均移确定聚类



## 3.3.1 多分辨率阈值选取

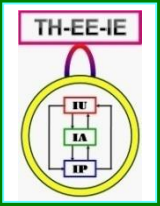
### 1. 确定分割区域的类数

- 利用在粗分辨率下的直方图细节信息
- 小波分解：尺度函数  $\phi(x) \leftrightarrow$  低通滤波器
- 图象直方图  $H(x)$  的低通分量为

$$S_{2^i} [H(x)] = H(x) \otimes \phi_{2^i}(x)$$

- 直方图的多分辨率小波分解表示

$$\{S_{2^i}[H(x)], W_{2^i}[H(x)], \quad 1 \leq i \leq I\}$$



## 3.3.1 多分辨率阈值选取

### 1. 确定分割区域的类数

- 在低分辨率时确定初始的区域分割类数
- 判断直方图中独立峰的个数
- 独立峰应满足三个条件：
  - (1) 具有一定的灰度范围
  - (2) 具有一定的峰下面积
  - (3) 具有一定的峰谷差

## 3.3.1 多分辨率阈值选取

### 2. 确定最优阈值

- (1) 用从负值变化到正值的零交叉点确定峰的起点
- (2) 用从正值变化到负值的零交叉点确定峰的终点
- (3) 用起点和终点间的最大值点确定峰的位置
- (4) 用前一个峰的终点和后一个峰的起点间的最小值点确定这两个峰之间谷点的位置

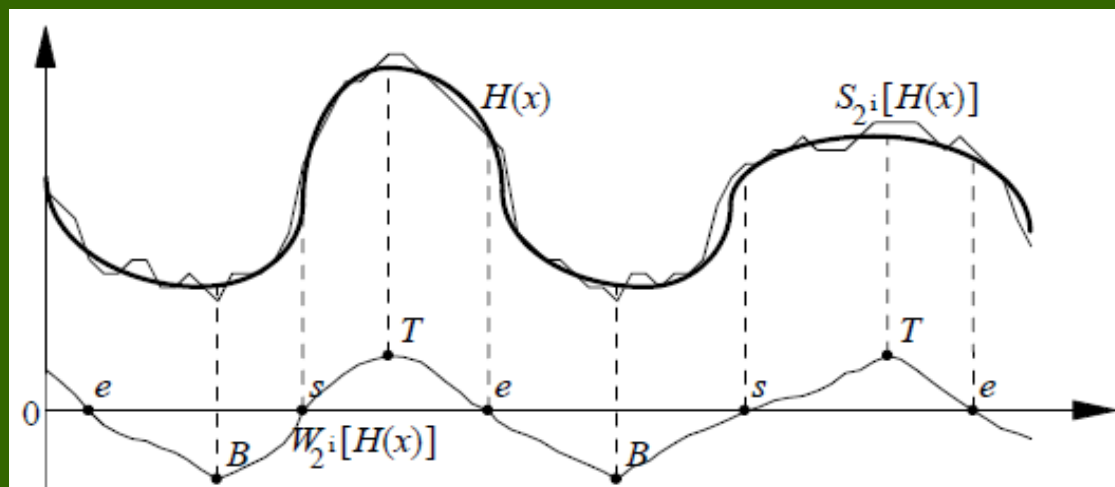
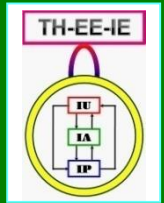


图 3.3.1 直方图的峰点和谷点的确定



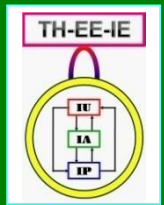
## 3.3.1 多分辨率阈值选取

### 2. 确定最优阈值

- 分辨率增加  $\rightarrow$  阈值数目也增加?
- 设在两相邻尺度  $2^{i+1}$  和  $2^i$  所对应的阈值分别为  $T_j^{i+1}$  和  $T_k^i$

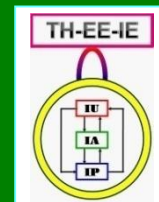
$$\text{dis}(T_j^{i+1}, T_k^i) = \min \left\{ \text{dis}(T_j^{i+1}, T_l^i), \quad l = 0, 1, \dots, N^i \right\}$$

- 在尺度  $2^{i+1}$  的阈值  $T_j^{i+1}$  对应于在尺度  $2^i$  的阈值  $T_k^i$
- 跟踪在最低分辨率一层选取的所有阈值
- 选取相应的最高分辨率一层的对应阈值



## 3.3.2 借助过渡区选择阈值

- 一般讨论基于区域或边界的算法时认为区域的并集覆盖了整个图象而边界本身没有宽度
- 实际数字图象中的边界是有宽度的
- 本身也是图象中的一个区域，一个特殊的区域
- 该区域一方面将不同的区域分隔开来，具有边界的特点；另一方面，其面积不为零，具有区域的特点
- 将这类特殊区域称为过渡区



## 3.3.2 借助过渡区选择阈值

### 1. 过渡区和有效平均梯度

有效平均梯度

$$EAG = \frac{TG}{TP}$$

$$TG = \sum_{i,j \in Z} g(i,j)$$

$$TP = \sum_{i,j \in Z} p(i,j)$$

$$p(i,j) = \begin{cases} 1 & \text{当 } g(i,j) > 0 \\ 0 & \text{当 } g(i,j) = 0 \end{cases}$$

剪切变换

$$f_{\text{low}}(i,j) = \begin{cases} f(i,j) & \text{当 } f(i,j) > L \\ L & \text{当 } f(i,j) \leq L \end{cases}$$

$$f_{\text{high}}(i,j) = \begin{cases} L & \text{当 } f(i,j) \geq L \\ f(i,j) & \text{当 } f(i,j) < L \end{cases}$$

**特点：** 将被剪切了的部分设为剪切值

## 3.3.2 借助过渡区选择阈值

### 2. 有效平均梯度的极值点和过渡区边界

$$L_{\text{low}} = \arg \left\{ \max_L [EAG_{\text{low}}(L)] \right\}$$

$$L_{\text{high}} = \arg \left\{ \max_L [EAG_{\text{high}}(L)] \right\}$$

$L_{\text{high}}$ 和 $L_{\text{low}}$ 在灰度值上限定了过渡区的灰度范围

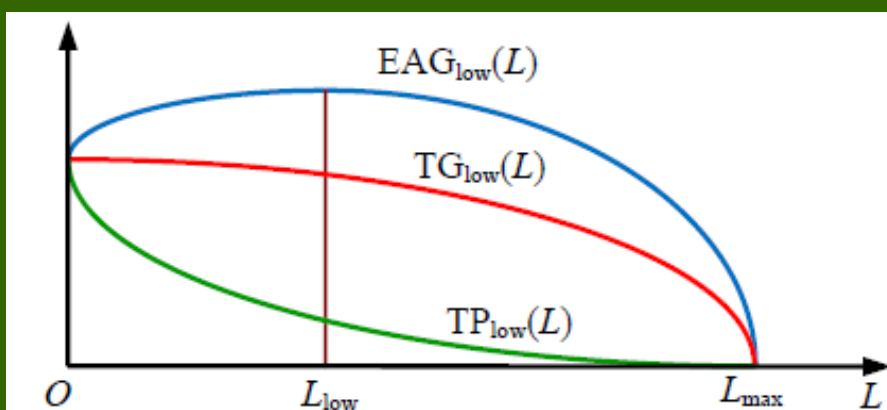


图 3.3.2 对  $EAG_{\text{low}}(L)$  曲线是单峰曲线的解释

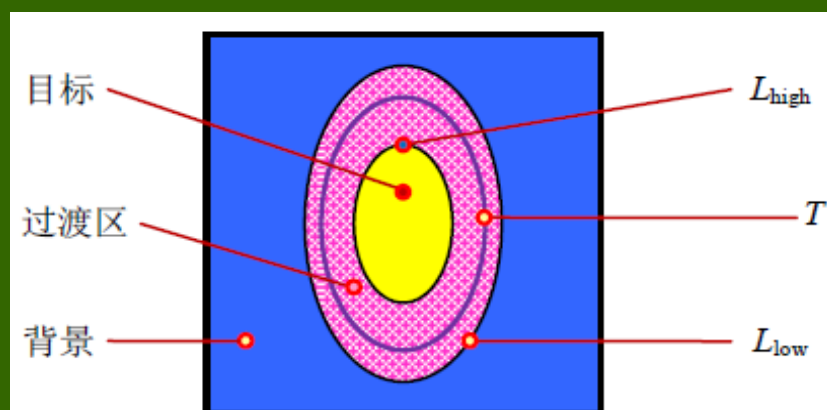


图 3.3.3 过渡区示例

## 3.3.2 借助过渡区选择阈值

### 推广到多阈值分割

两个极值点有三个重要的性质 (P.67)

两个极值点的三个重要性质在图象中有不止一个过渡区时也成立

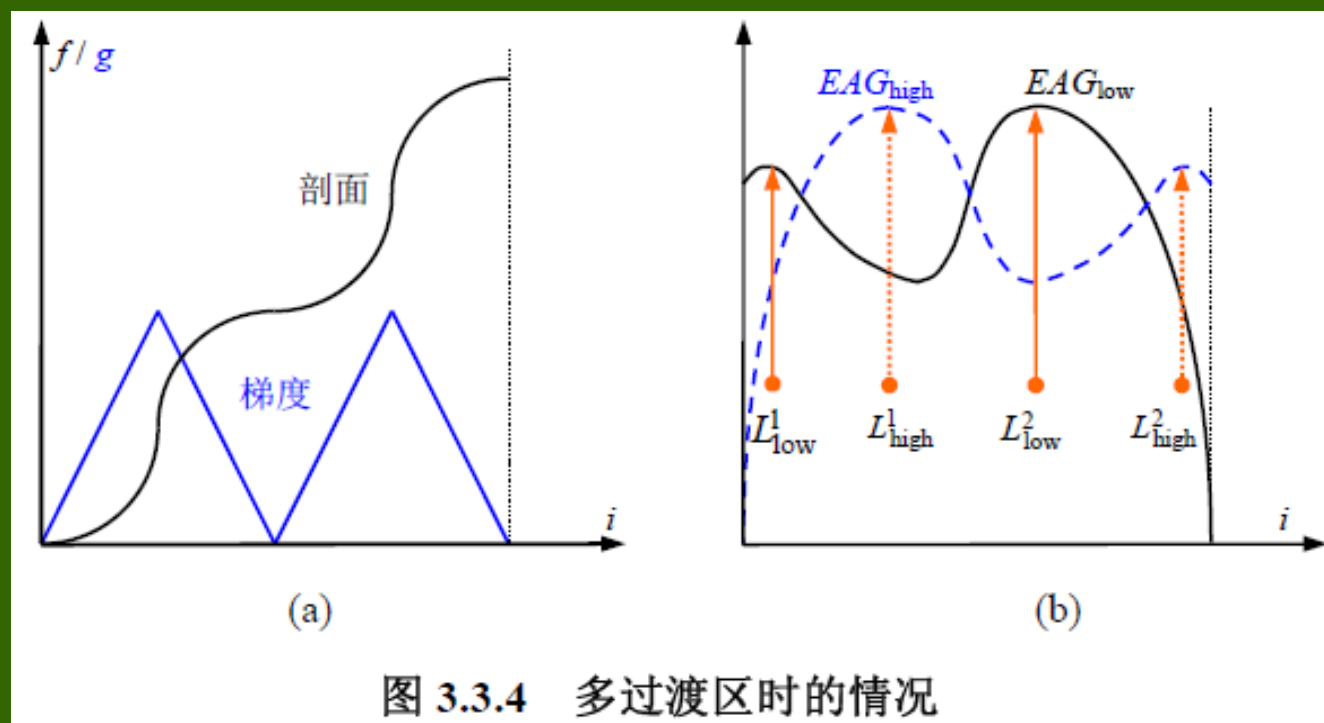


图 3.3.4 多过渡区时的情况

## 3.3.3 借助均移确定聚类

一种非参数技术，**均移**指偏移的均值向量

首先随机选择一个初始的感兴趣区域（初始窗）并确定其重心；接下来搜索周围特征点密度更大的感兴趣区域并确定其重心；重复上面的过程不断将均值移动直到收敛

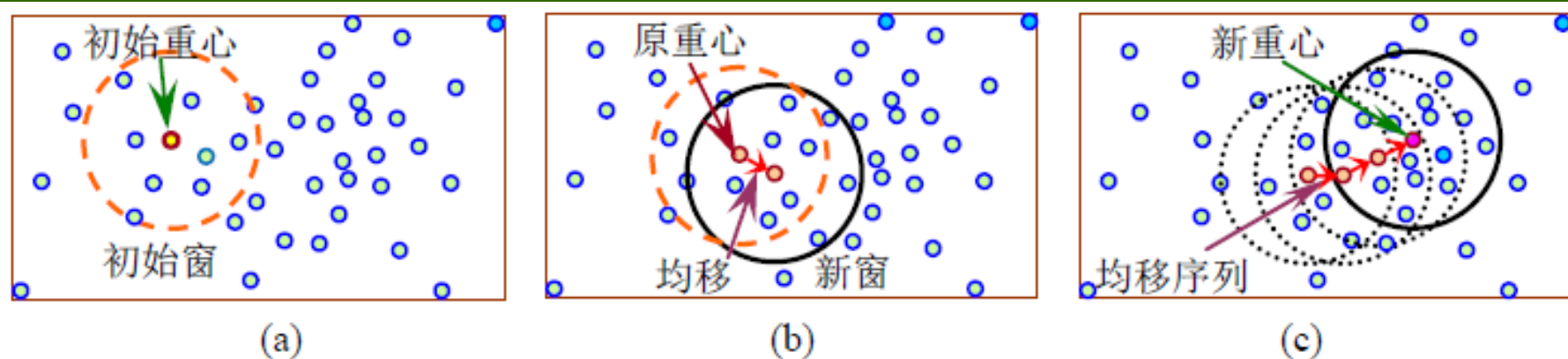
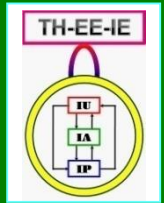


图 3.3.5 均移方法的原理示意



## 3.4 分水岭分割算法

- 分水岭 (watershed, 也称分水线/水线)
- 把图象看成3-D地形的表示, 即2-D的地基 (对应图象空间坐标) 加上第3维的高度 (对应图象灰度值)
- 计算过程是串行的, 得到的是目标的边界 (但利用了区域的一致性) ④ 串行区域类

3.4.1 基本原理和步骤

3.4.2 算法改进和扩展

## 3.4.1 基本原理和步骤

### 1. 分水岭

建立不同目标间的分水岭（涨水法）

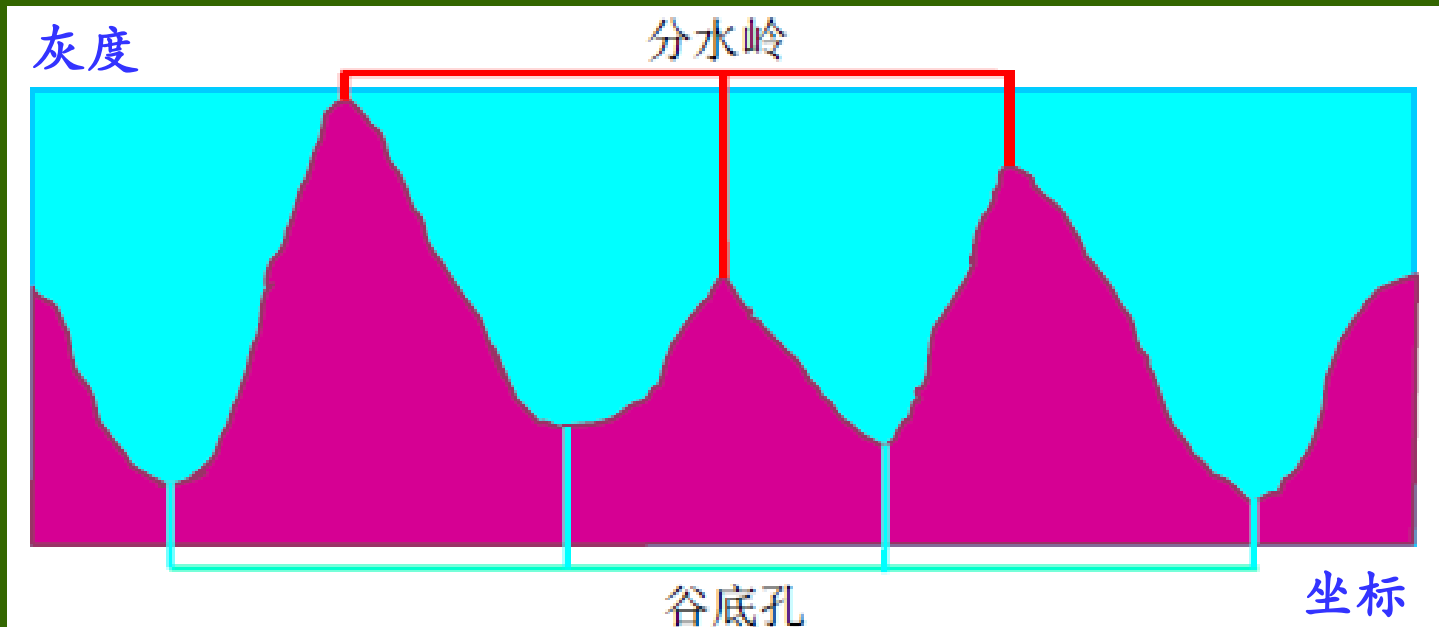
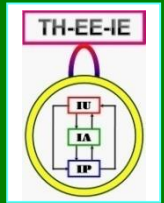


图 3.4.2 涨水法分水岭示意



## 3.4.1 基本原理和步骤

### 2. 分水岭计算步骤

设待分割图象为  $f(x, y)$ ，其梯度图象为  $g(x, y)$

用  $M_1, M_2, \dots, M_R$  表示  $g(x, y)$  中各局部极小值的象素位置， $C(M_i)$  为与  $M_i$  对应的区域中的象素坐标集合

用  $n$  表示当前灰度阈值， $T[n]$  代表记为  $(u, v)$  的象素集合， $g(u, v) < n$ ， $T[n] = \{(u, v) \mid g(u, v) < n\}$

对  $M_i$  所在的区域，其中满足条件的坐标集合  $C_n(M_i)$  可看作一幅二值图象

$$C_n(M_i) = C(M_i) \cap T[n]$$

## 3.4.1 基本原理和步骤

### 2. 分水岭计算步骤

$$C_n(M_i) = C(M_i) \cap T[n]$$

梯度

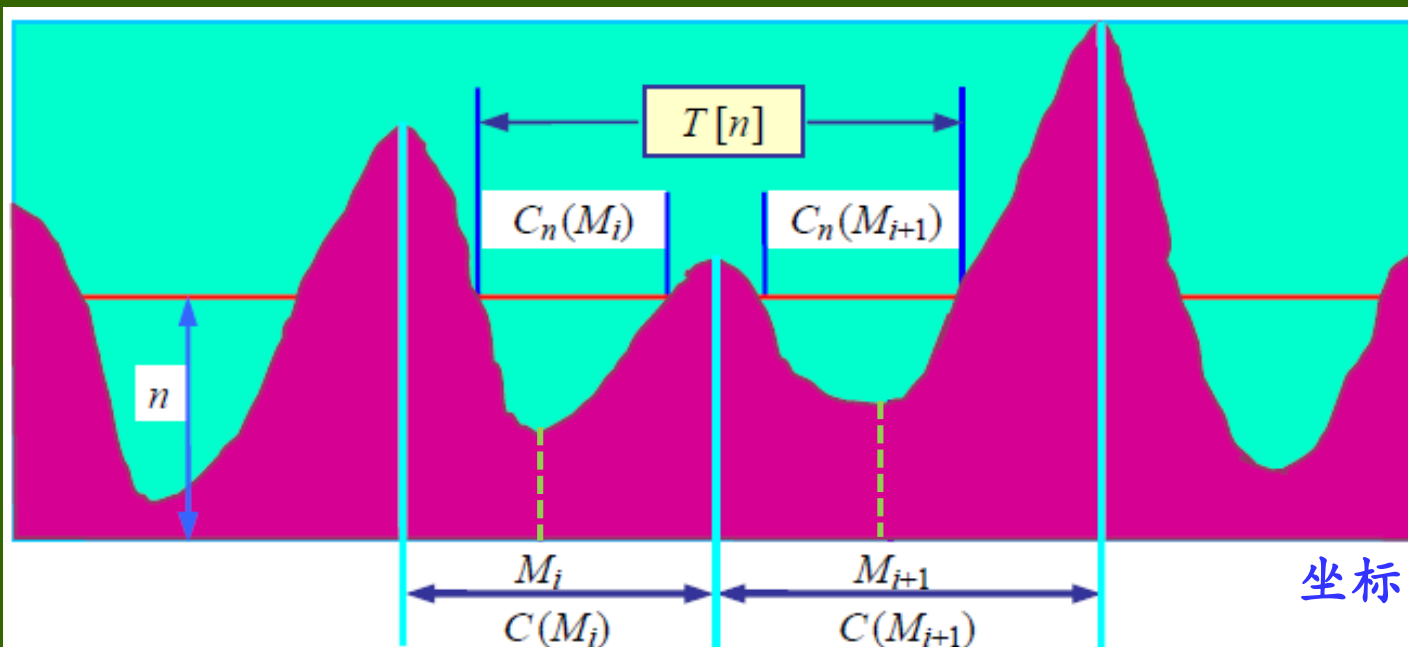
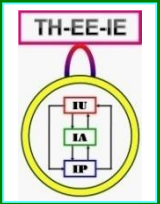


图 3.4.4 计算  $C_n(M_i)$  的示意图



## 3.4.1 基本原理和步骤

### 2. 分水岭计算步骤

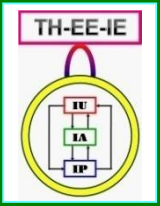
用 $C[n]$ 代表在灰度阈值为 $n$ 时图象中所有满足条件（被淹没）的像素

$$C[n] = \bigcup_{i=1}^R C_n(M_i)$$

$C[\max + 1]$ 将是所有区域的并集

$$C[\max + 1] = \bigcup_{i=1}^R C_{\max+1}(M_i)$$

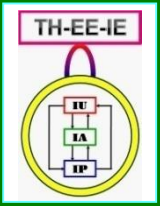
$C[n-1]$ 是 $C[n]$ 的子集， $C[n]$ 是 $T[n]$ 的子集，所以  
 $C[n-1]$ 又是 $T[n]$ 的子集



## 3.4.1 基本原理和步骤

### 2. 分水岭计算步骤

- 令 $S$ 代表 $T[n]$ 中的连通组元集合，对每个（新的）连通组元 $s \in S[n]$ ，有3种可能性：
  - (1)  $s \cap C[n-1]$ 是1个空集
  - (2)  $s \cap C[n-1]$ 里包含 $C[n-1]$ 中的一个连通组元
  - (3)  $s \cap C[n-1]$ 里包含 $C[n-1]$ 中一个以上的连通组元
- 分别处理：
  - (1)  $C[n]$ 可由把连通组元 $s$ 加到 $C[n-1]$ 中得到
  - (2)  $C[n]$ 可由把连通组元 $s$ 加到 $C[n-1]$ 中得到
  - (3) 需要在 $s$ 中建分水岭（否则不同区域将被连通）



## 3.4.2 算法改进和扩展

### 1. 利用标号控制分割

- 问题：过分割（over-segmentation）  
受图象中噪声和其它不规则结构影响
- 方案：利用标号（marker）  
图象中的一个连通组元  
内部标号：对应目标  
外部标号：对应背景（分水岭）

## 3.4.2 算法改进和扩展

### 1. 利用标号控制分割 流程框图

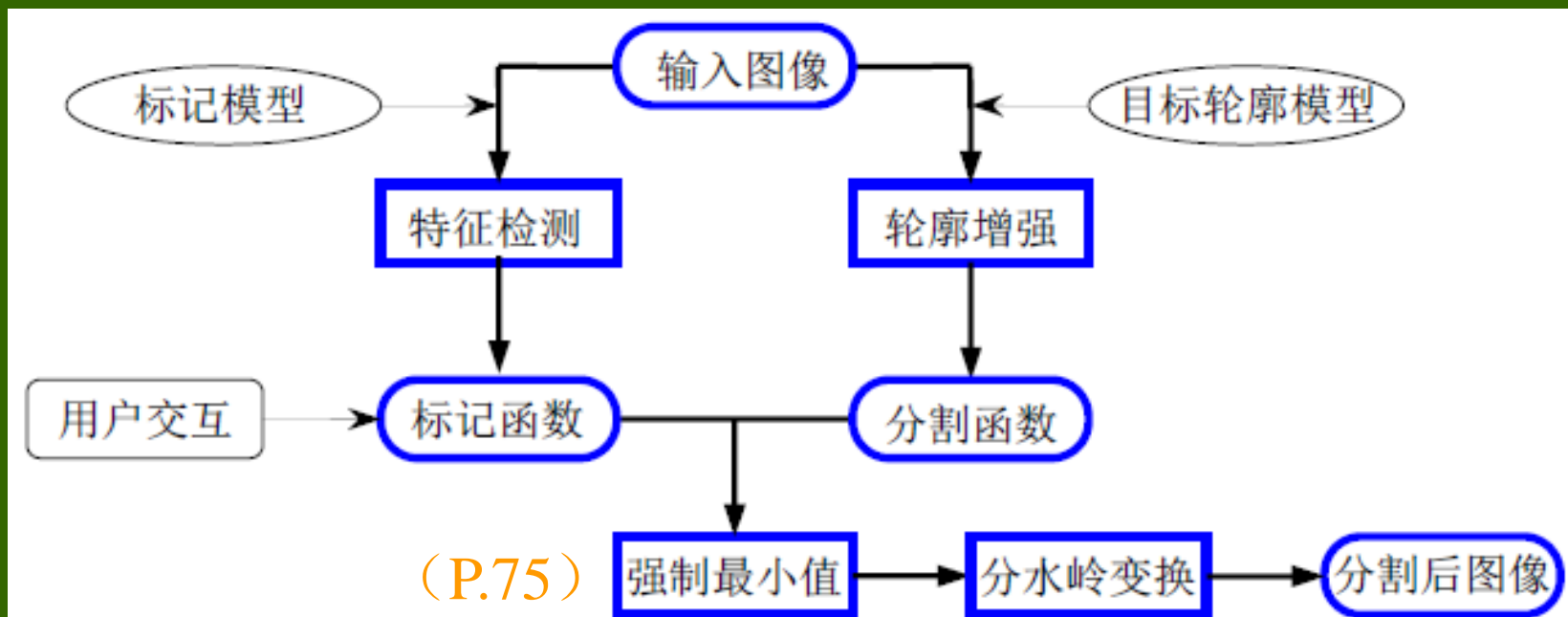


图 3.4.7 标号控制分割方法的流程框图



# 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](mailto:zhang-yj@tsinghua.edu.cn)
- ☞ 个人主页：[oa.ee.tsinghua.edu.cn/~zhangyujin/](http://oa.ee.tsinghua.edu.cn/~zhangyujin/)