

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

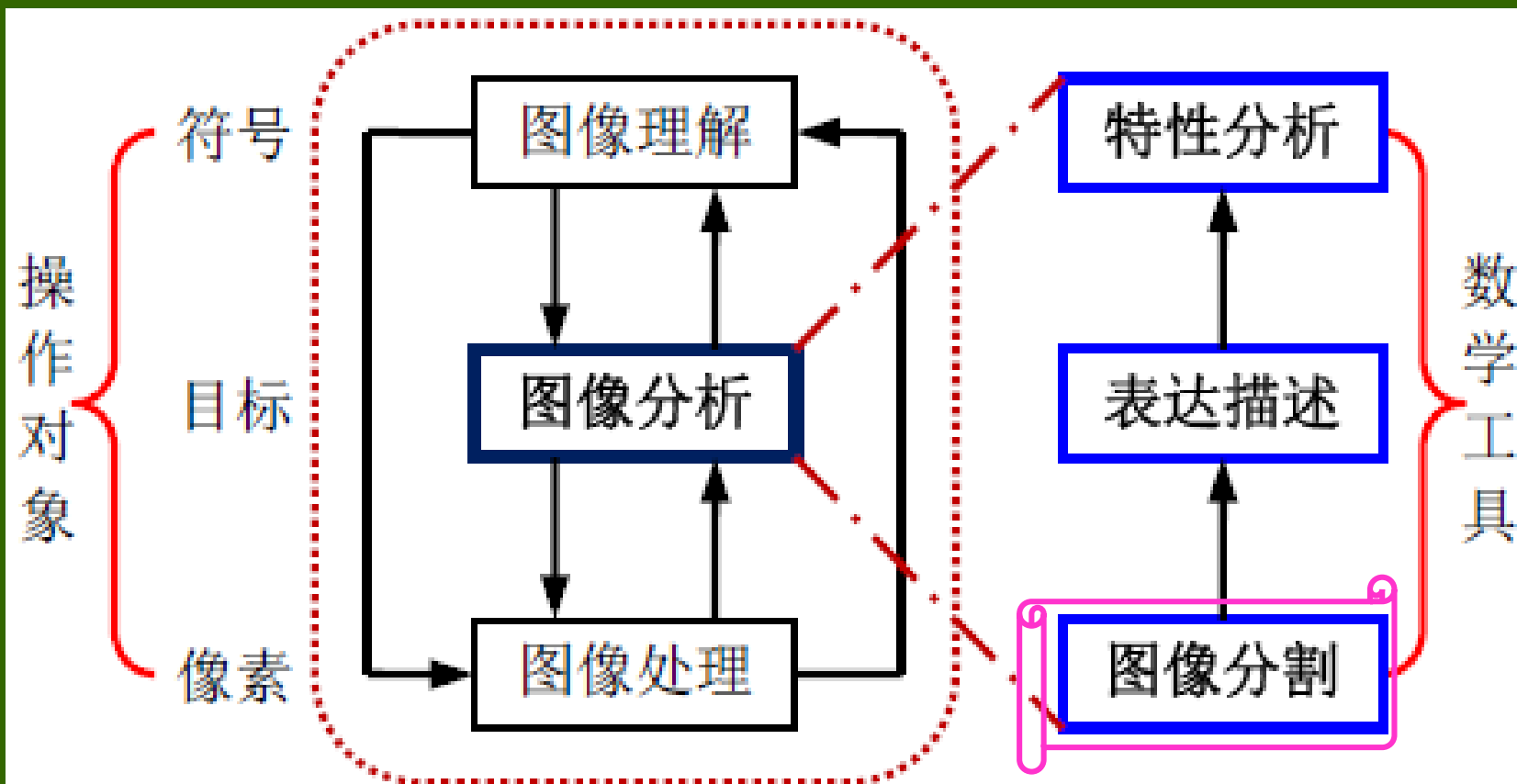
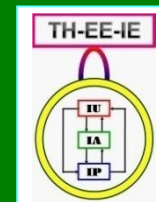


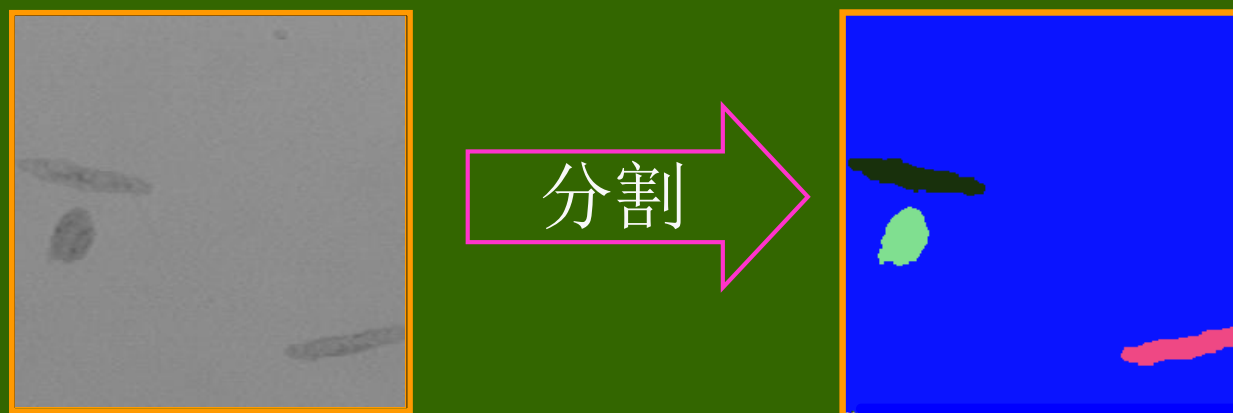
图 1.2.2 图像分析主要功能模块

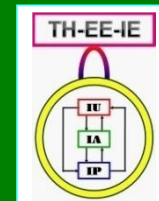


# 第1单元 图象分割

## 图象分割

前景（感兴趣目标），背景  
分离区域，提取目标（语义）  
利用区域特性（灰度、颜色、纹理、……）

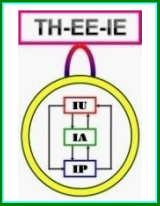




# 第1单元 图象分割

## 20年图象工程中排名前八小类文献数量

排名	层次	研究/技术类别	文献数
1	分析	图象分割和边缘检测	1411
2	处理	图象增强和恢复等	1130
3	分析	目标检测和识别	966
4	处理	图象（视频）压缩编码	945
5	理解	图象匹配和融合等	933
6	分析	人体生物特征提取和验证	773
7	处理	图象数字水印和图象信息隐藏	662
8	处理	图象获取	637

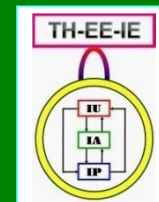


# 第1单元 图象分割

- 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

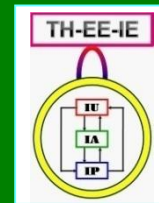
图象分割是由图象处理进到图象分析的关键步骤

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2章 图象分割基础

- 2.1 图象分割定义和技术分类
- 2.2 并行边界技术
- 2.3 串行边界技术
- 2.4 并行区域技术
- 2.5 串行区域技术



## 2.1 图象分割定义和技术分类

### 图象分割严格定义

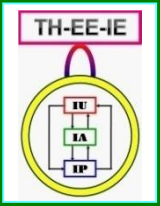
$$(1) \bigcup_{i=1}^n R_i = R$$

(2) 对所有的  $i$  和  $j$ ,  $i \neq j$ , 有  $R_i \cap R_j = \emptyset$

(3) 对  $i = 1, 2, \dots, n$ , 有  $P(R_i) = \text{TRUE}$

(4) 对  $i \neq j$ , 有  $P(R_i \cup R_j) = \text{FALSE}$

(5) 对  $i = 1, 2, \dots, n$ ,  $R_i$  是连通的区域



# 2.1 图象分割定义和技术分类

## 图象分割算法分类

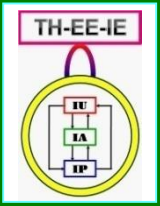
- 不连续性和相似性
- 并行处理策略和串行处理策略

### 分类表

表 2.1.1 分割技术分类表

分类	边界（不连续性）	区域（相似性）
并行处理	PB: 并行边界类	PR: 并行区域类
串行处理	SB: 串行边界类	SR: 串行区域类

- ① 并行边界类
- ② 串行边界类
- ③ 并行区域类
- ④ 串行区域类



## 2.2 并行边界技术

在灰度图中，两个不同的相邻区域之间灰度值会有不连续或局部突变，从而导致出现边缘

2.2.1 边缘及检测原理

2.2.2 正交梯度算子

2.2.3 二阶导数算子

2.2.4 边界闭合

## 2.2.1 边缘及检测原理

### 图象剖面上的几个参数

#### ① 并行边界类

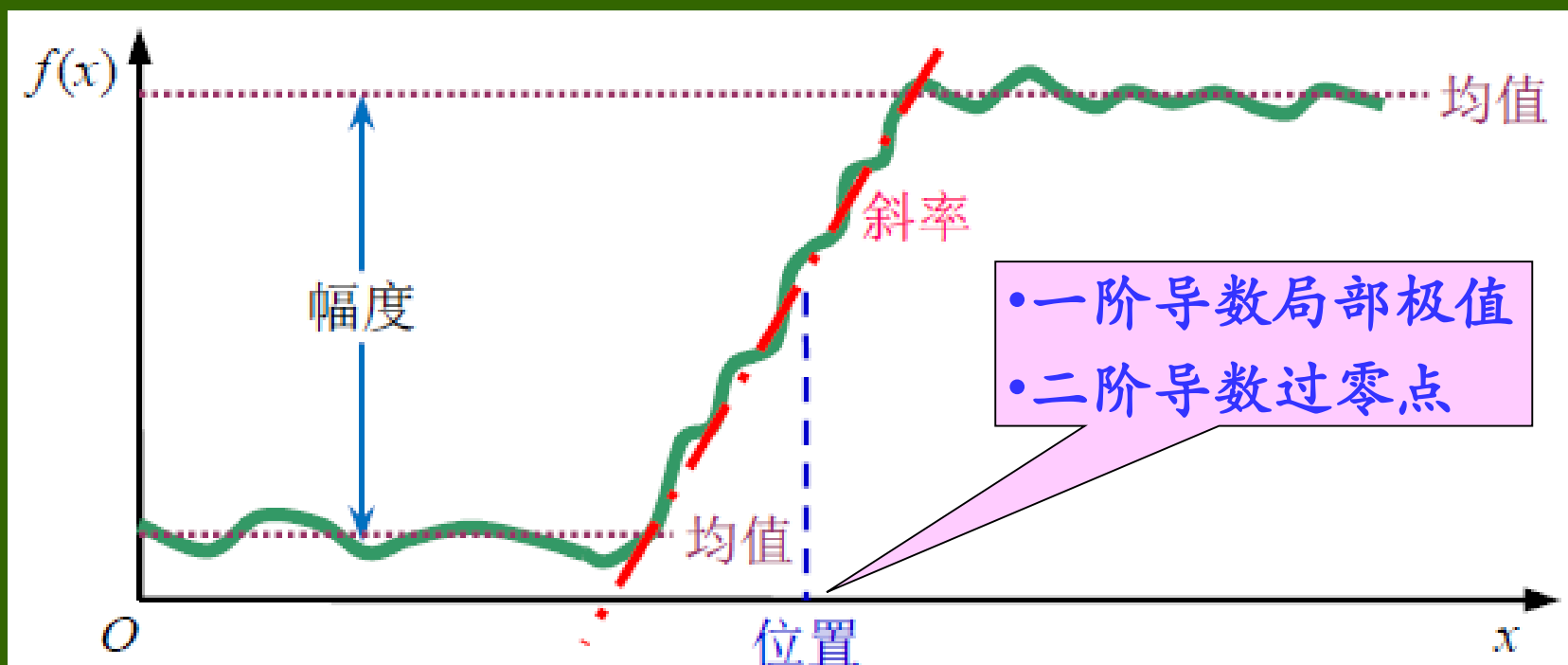
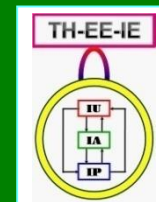


图 2.2.1 描述边缘的几个参数



## 2.2.2 正交梯度算子

### 梯度

#### ① 并行边界类

$$\nabla f(x, y) = [G_x \quad G_y]^T = \left[ \frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right]^T \quad \text{一阶导数}$$

$$\text{mag}(\nabla f) = \|\nabla f_{(2)}\| = [G_x^2 + G_y^2]^{1/2}$$

$$\phi(x, y) = \arctan(G_y / G_x)$$

1	
	-1

	1
-1	

(a) Roberts

-1		1
-1		1
-1		1

1	1	1
-1	-1	-1

(b) Prewitt

-1		1
-2		2
-1		1

1	2	1
-1	-2	-1

(c) Sobel

图 2.2.2 几种常用梯度算子的模板

## 2.2.2 正交梯度算子

组合两个模板的输出

① 并行边界类

(计算简便)

$$\|\nabla f_{(1)}\| = |G_x| + |G_y|$$

$$\|\nabla f_{(\infty)}\| = \max\{|G_x|, |G_y|\}$$

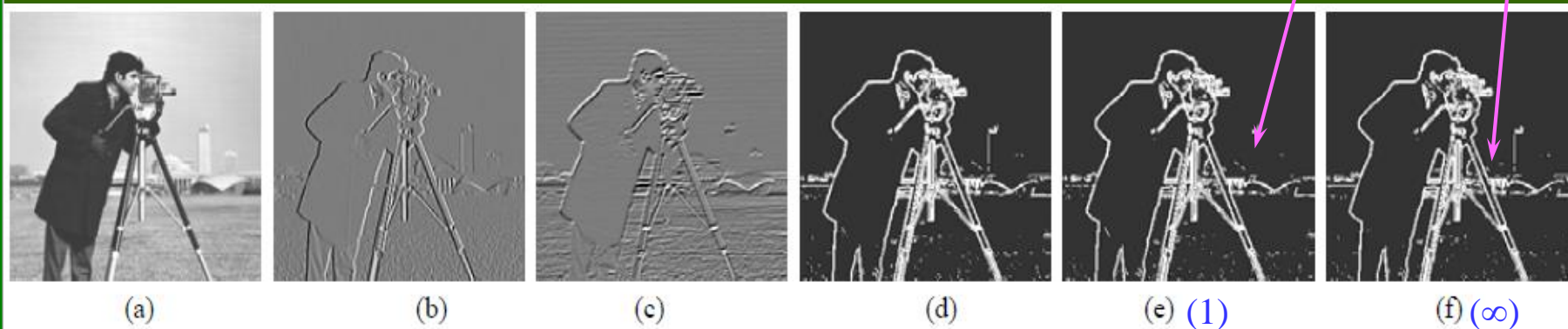
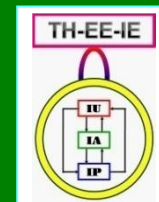


图 2.2.3 梯度图示例



## 2.2.3 方向微分算子

### 基尔希模板 ( $3 \times 3$ )

#### ① 并行边界类

用一组模板与图象卷积，分别计算不同方向上的差分  
值，取其中最大的值作为边缘强度，而将与之对应的方向  
作为边缘方向

各方向之间的  
夹角均为 $45^\circ$

<table><tr><td>-5</td><td>3</td><td>3</td></tr><tr><td>-5</td><td></td><td>3</td></tr><tr><td>-5</td><td>3</td><td>3</td></tr></table>	-5	3	3	-5		3	-5	3	3	<table><tr><td>3</td><td>3</td><td>3</td></tr><tr><td>-5</td><td>0</td><td>3</td></tr><tr><td>-5</td><td>-5</td><td>3</td></tr></table>	3	3	3	-5	0	3	-5	-5	3	<table><tr><td>3</td><td>3</td><td>3</td></tr><tr><td>3</td><td>0</td><td>3</td></tr><tr><td>-5</td><td>-5</td><td>-5</td></tr></table>	3	3	3	3	0	3	-5	-5	-5	<table><tr><td>3</td><td>3</td><td>3</td></tr><tr><td>3</td><td>0</td><td>-5</td></tr><tr><td>3</td><td>-5</td><td>-5</td></tr></table>	3	3	3	3	0	-5	3	-5	-5
-5	3	3																																					
-5		3																																					
-5	3	3																																					
3	3	3																																					
-5	0	3																																					
-5	-5	3																																					
3	3	3																																					
3	0	3																																					
-5	-5	-5																																					
3	3	3																																					
3	0	-5																																					
3	-5	-5																																					
<table><tr><td>3</td><td>3</td><td>-5</td></tr><tr><td>3</td><td></td><td>-5</td></tr><tr><td>3</td><td>3</td><td>-5</td></tr></table>	3	3	-5	3		-5	3	3	-5	<table><tr><td>3</td><td>-5</td><td>-5</td></tr><tr><td>3</td><td>0</td><td>-5</td></tr><tr><td>3</td><td>3</td><td>3</td></tr></table>	3	-5	-5	3	0	-5	3	3	3	<table><tr><td>-5</td><td>-5</td><td>-5</td></tr><tr><td>3</td><td>0</td><td>3</td></tr><tr><td>3</td><td>3</td><td>3</td></tr></table>	-5	-5	-5	3	0	3	3	3	3	<table><tr><td>-5</td><td>-5</td><td>3</td></tr><tr><td>-5</td><td>0</td><td>3</td></tr><tr><td>3</td><td>3</td><td>3</td></tr></table>	-5	-5	3	-5	0	3	3	3	3
3	3	-5																																					
3		-5																																					
3	3	-5																																					
3	-5	-5																																					
3	0	-5																																					
3	3	3																																					
-5	-5	-5																																					
3	0	3																																					
3	3	3																																					
-5	-5	3																																					
-5	0	3																																					
3	3	3																																					

图 2.2.4 8 方向基尔希算子的  $3 \times 3$  模板

## 2.2.3 方向微分算子

### 基尔希模板 ( $3 \times 3$ )

#### ① 并行边界类

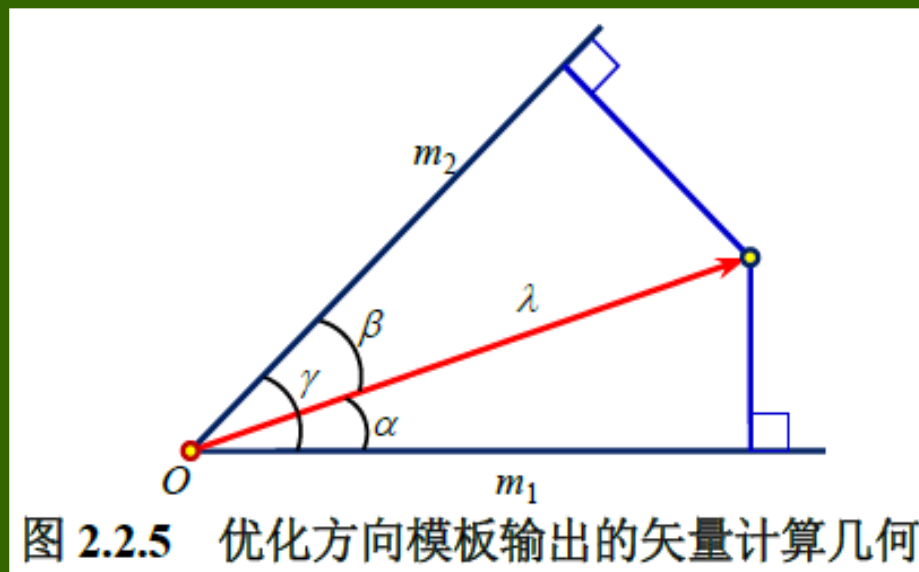
优化方向模板的输出以获得精确的朝向数值和精确的幅度数值

$$m_1 = \lambda \cos \alpha$$

$$m_2 = \lambda \cos \beta$$

$$\alpha = \arctan \left[ \left( \frac{m_2}{m_1} \right) \operatorname{cosec} \gamma - \cot \gamma \right]$$

$$\lambda = \left( m_1^2 + m_2^2 - 2m_1m_2 \cos \gamma \right)^{1/2} \operatorname{cosec} \gamma$$



## 2.2.4 二阶导数算子

### 拉普拉斯算子

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

### ① 并行边界类

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

-2	-3	-2
-3	20	-3
-2	-3	-2

图 2.2.4 拉普拉斯算子的模板

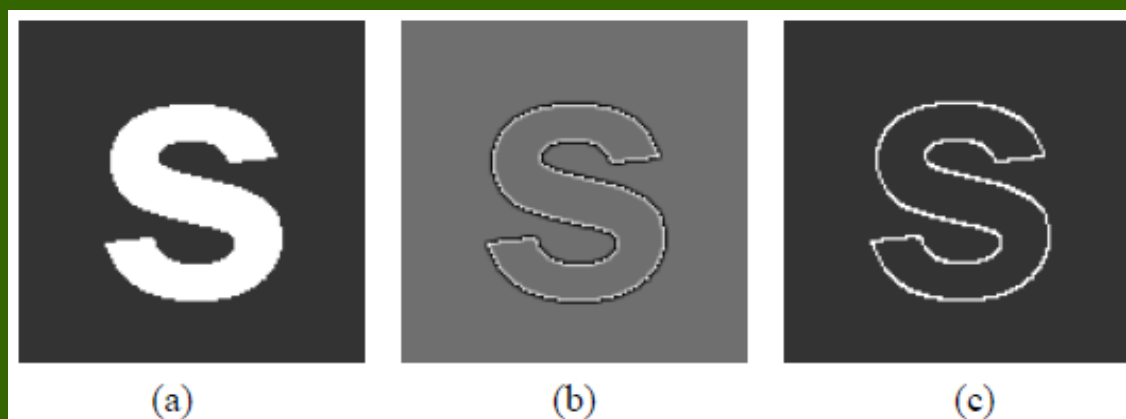


图 2.2.5 拉普拉斯图示例

## 2.2.4 二阶导数算子

### 例2.2.3 借助导数检测2-D边缘

两个一阶（偏）导数和三种二阶（偏）导数

两种导数返回的边缘是不一样的，而且与理想边缘也不一致

二阶导数的结果比一阶导数的结果与理想结果的差距更大

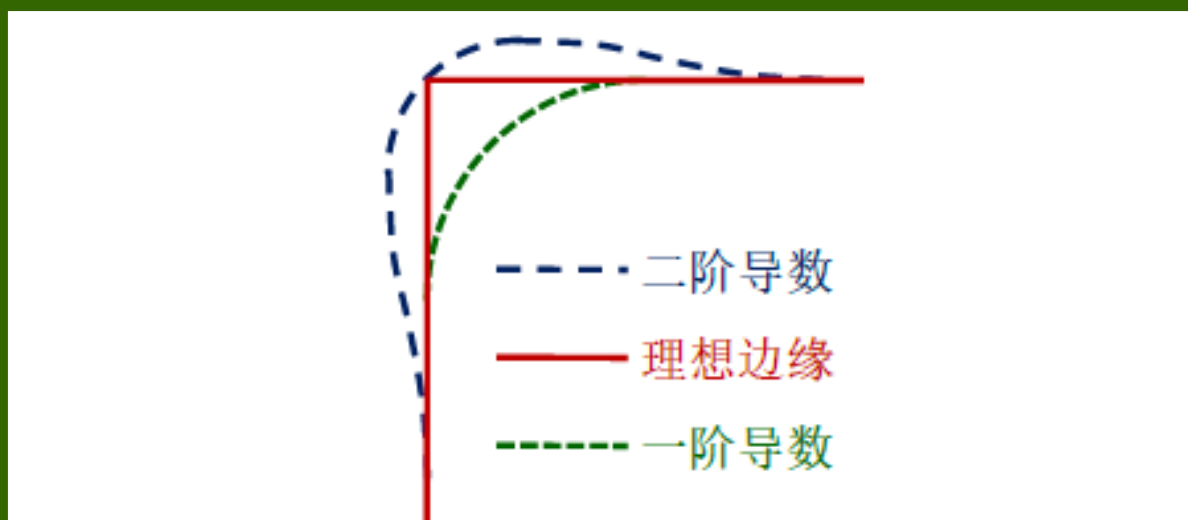
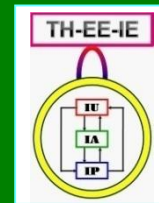


图 2.2.8 一阶导数和二阶导数与理想边缘的不同



## 2.2.4 二阶导数算子

### 马尔算子

#### ① 并行边界类

- 具有一定的生物学和生理学意义
- 先平滑原始图象后再运用拉普拉斯算子
- 对不同分辨率的图象分别处理
  - (1) 用一个2-D的高斯平滑模板与原始图象进行卷积
  - (2) 计算卷积后图象的拉普拉斯值
  - (3) 检测拉普拉斯值图象中的过零点，以此作为边缘点

## 2.2.4 二阶导数算子

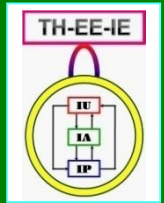
### 坎尼算子

#### ① 并行边界类

- (a) 用高斯滤波器平滑图象以减轻噪声影响
- (b) 检测滤波图象中灰度梯度的大小和方向
- (c) 保留局部最大以细化边缘像素所构成的边界
- (d) 借助滞后阈值化方法最后确定边界像素



图 2.2.6 坎尼算子步骤示例



## 2.2.4 二阶导数算子

### 德瑞切 (Deriche) 算子

① 并行边界类

能用递归方式来实现的理想边缘检测滤波器

$$d'_{1\sigma}(x) = -\sigma^2 x e^{-\sigma|x|}$$

$$d'_{2\sigma}(x) = -2\sigma \sin(\sigma x) e^{-\sigma|x|}$$

平滑滤波器 (计算时间不受平滑参数 $\sigma$ 影响)

$$d_{1\sigma}(x) = \frac{1}{4} \sigma (\sigma |x| + 1) e^{-\sigma|x|}$$

$$d_{2\sigma}(x) = \frac{1}{2} \sigma [\sin(\sigma |x|) + \cos(\sigma |x|)] e^{-\sigma|x|}$$

## 2.2.5 边界闭合

### 连接边缘像素

### ① 并行边界类

同时考虑梯度  
的大小和方向

$$|\nabla f(x, y) - \nabla f(s, t)| \leq T$$

$$|\phi(x, y) - \phi(s, t)| \leq A$$

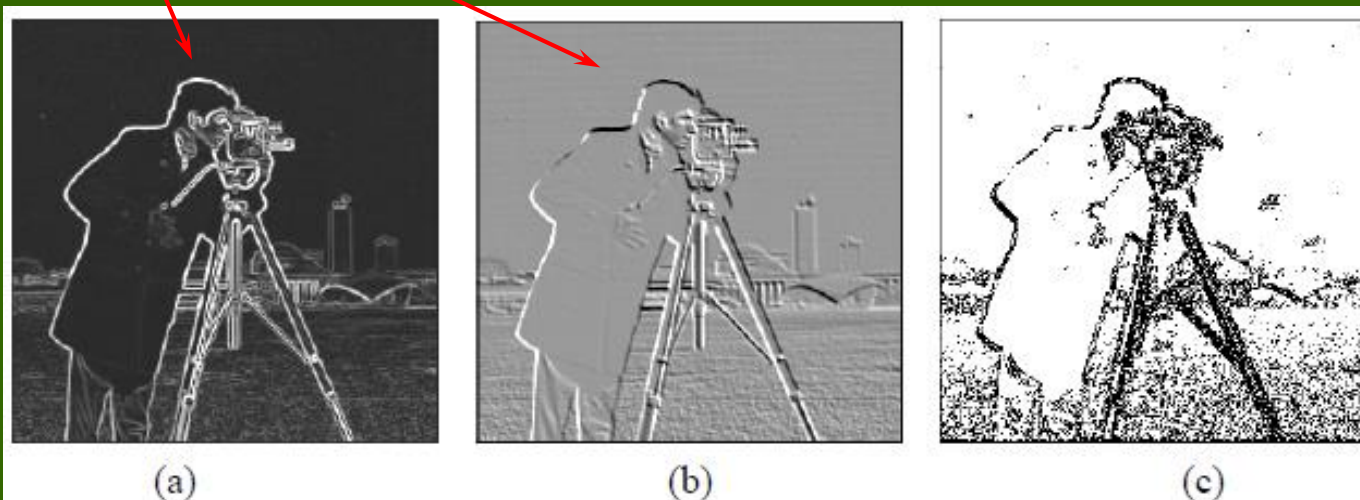
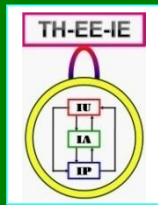


图 2.2.8 根据梯度实现边界闭合



## 2.3 串行边界技术

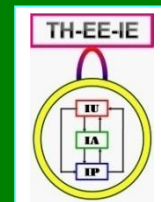
考虑图象中边界的全局信息

早期的典型技术包括基于图搜索和基于动态规划的方法（一边检测边缘点一边连接边缘点）

一个先初始化一个闭合边界，再逐步迭代地调整到真实边缘位置的方法      ⇒      ② 串行边界类

### 2.3.1 主动轮廓模型

### 2.3.2 能量函数



## 2.3.1 主动轮廓模型

主动轮廓:

② 串行边界类

图象上一组排序点的集合

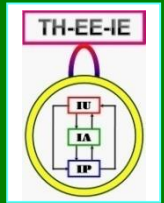
$$V = \{v_1, \dots, v_L\} \quad v_i = (x_i, y_i), \quad i = \{1, \dots, L\}$$

处在轮廓上的点可通过解一个最小能量问题来迭代地逼近目标的边界

$$E_i(v_i') = \alpha E_{\text{int}}(v_i') + \beta E_{\text{ext}}(v_i')$$

$E_{\text{int}}(v_i)$ 是依赖于轮廓形状的（内部）能量函数

$E_{\text{ext}}(v_i)$ 是依赖于图象性质的（外部）能量函数



## 2.3.2 能量函数

### 1. 内部能量

### ② 串行边界类

推动主动轮廓形状的改变  
并保持轮廓上点间的距离

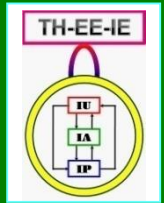
$$\alpha E_{\text{int}}(\mathbf{v}_i) = cE_{\text{con}}(\mathbf{v}_i) + bE_{\text{bal}}(\mathbf{v}_i)$$

#### (1) 连续能量 (2.3.5)

迫使不封闭的曲线变成直线而封闭的曲线变成圆环

#### (2) 膨胀力 (2.3.8)

强制轮廓在没有外来影响的情况下扩展或收缩



## 2.3.2 能量函数

### 2. 外部能量

#### ② 串行边界类

将变形模板向感兴趣的特征（如边缘）吸引  
构建能量函数：考虑目标的尺寸和形状

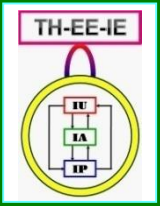
$$\beta E_{\text{ext}}(\mathbf{v}_i) = mE_{\text{mag}}(\mathbf{v}_i) + gE_{\text{grad}}(\mathbf{v}_i)$$

(1) 图象灰度能量 (2.3.11)

将轮廓推向高或低的灰度区域（避免局部极值）

(2) 图象梯度能量 (2.3.12)

将轮廓推向边缘（找寻目标边界）



## 2.4 并行区域技术

使用非常广泛的一类分割技术

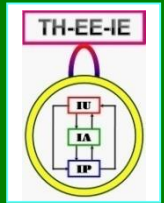
2.4.1 原理和分类

2.4.2 依赖像素的阈值选取

2.4.3 依赖区域的阈值选取

2.4.4 依赖坐标的阈值选取

2.4.5 空间聚类



## 2.4.1 原理和分类

### 取阈值进行分割（阈值化—thresholding）

图象模型：

双峰直方图（对应目标和背景的两个  
单峰直方图混合而成）

大小接近，均值相距足够远，均方差足够小

取阈值分割步骤：  $\Rightarrow$  ③ 并行区域类

(1) 确定阈值（关键/难点）

(2) 根据阈值划分像素

## 2.4.1 原理和分类

### 单阈值分割图象

### ③ 并行区域类

对灰度图（取值在 $g_{\min}$ 和 $g_{\max}$ 之间）确定一个灰度阈值  
 $T$  ( $g_{\min} < T < g_{\max}$ )

$$g(x, y) = \begin{cases} 1 & \text{如 } f(x, y) > T \\ 0 & \text{如 } f(x, y) \leq T \end{cases}$$

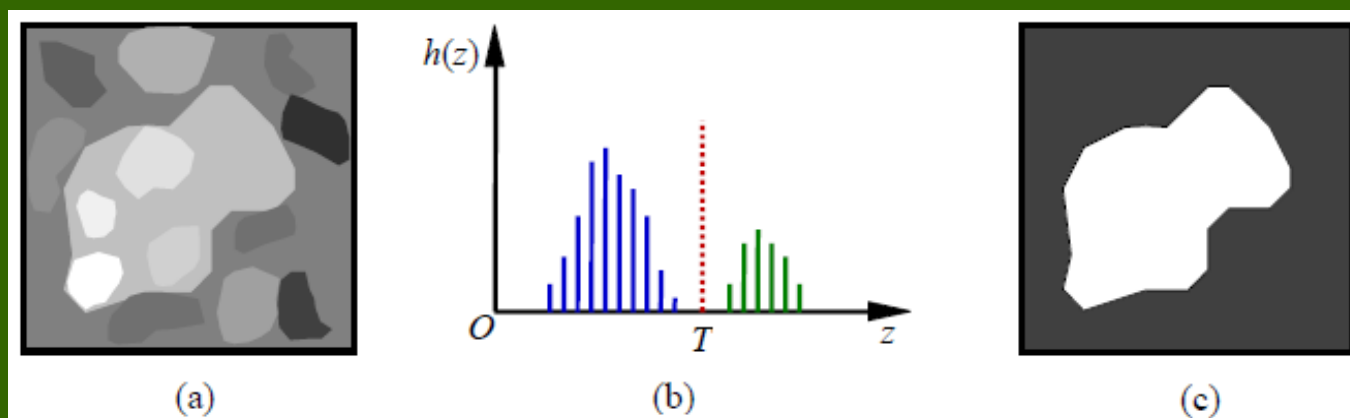
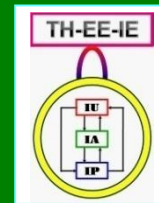


图 2.4.1 单阈值分割图示



## 2.4.1 原理和分类

### 阈值分割方法分类

### ③ 并行区域类

$$T = T[x, y, f(x, y), q(x, y)]$$

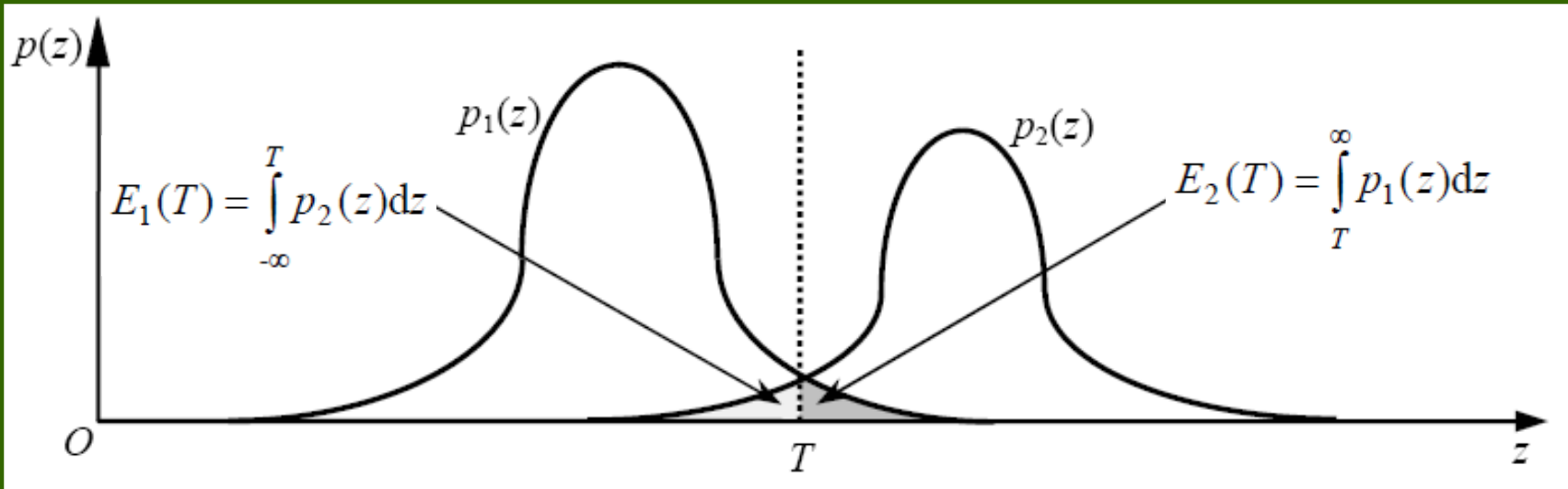
- (1) 依赖象素的（全局）阈值方法  
仅根据  $f(x, y)$  来选取阈值
  - (2) 依赖区域的（局部）阈值方法  
根据  $f(x, y)$  和  $q(x, y)$  来选取阈值
  - (3) 依赖坐标的（动态）阈值方法  
除根据  $f(x, y)$  和  $q(x, y)$  来选取，还与  $x, y$  有关
- 将前两种阈值也称为固定阈值

## 2.4.2 依赖象素的阈值选取

### 1. 最优阈值（最小化误差）

$$p(z) = P_1 p_1(z) + P_2 p_2(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(z-\mu_1)^2}{2\sigma_1^2}\right] + \frac{P_2}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(z-\mu_2)^2}{2\sigma_2^2}\right]$$

$$E(T) = P_2 \times E_1(T) + P_1 \times E_2(T)$$



## 2.4.2 依赖象素的阈值选取

### 2. 由直方图凹凸性确定的阈值

直方图的一个峰淹没在另一个峰旁的缓坡里

直方图的包络  
→ 区域凸包  
→ 最大凸残差  
→ 分割阈值

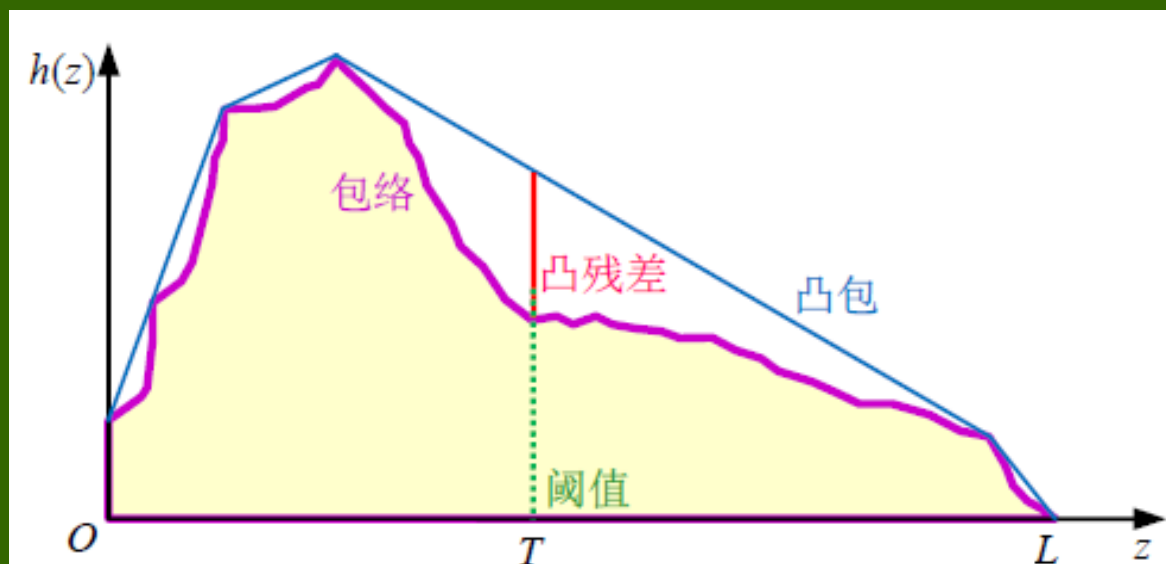
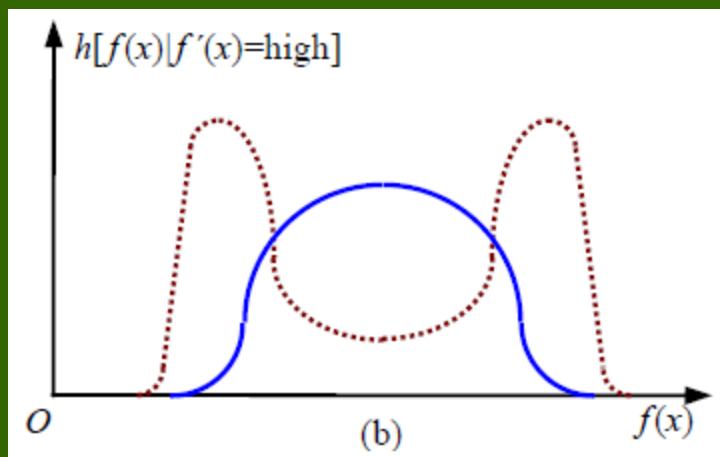


图 2.4.4 分析直方图凹凸性来确定分割阈值

## 2.4.3 依赖区域的阈值选取

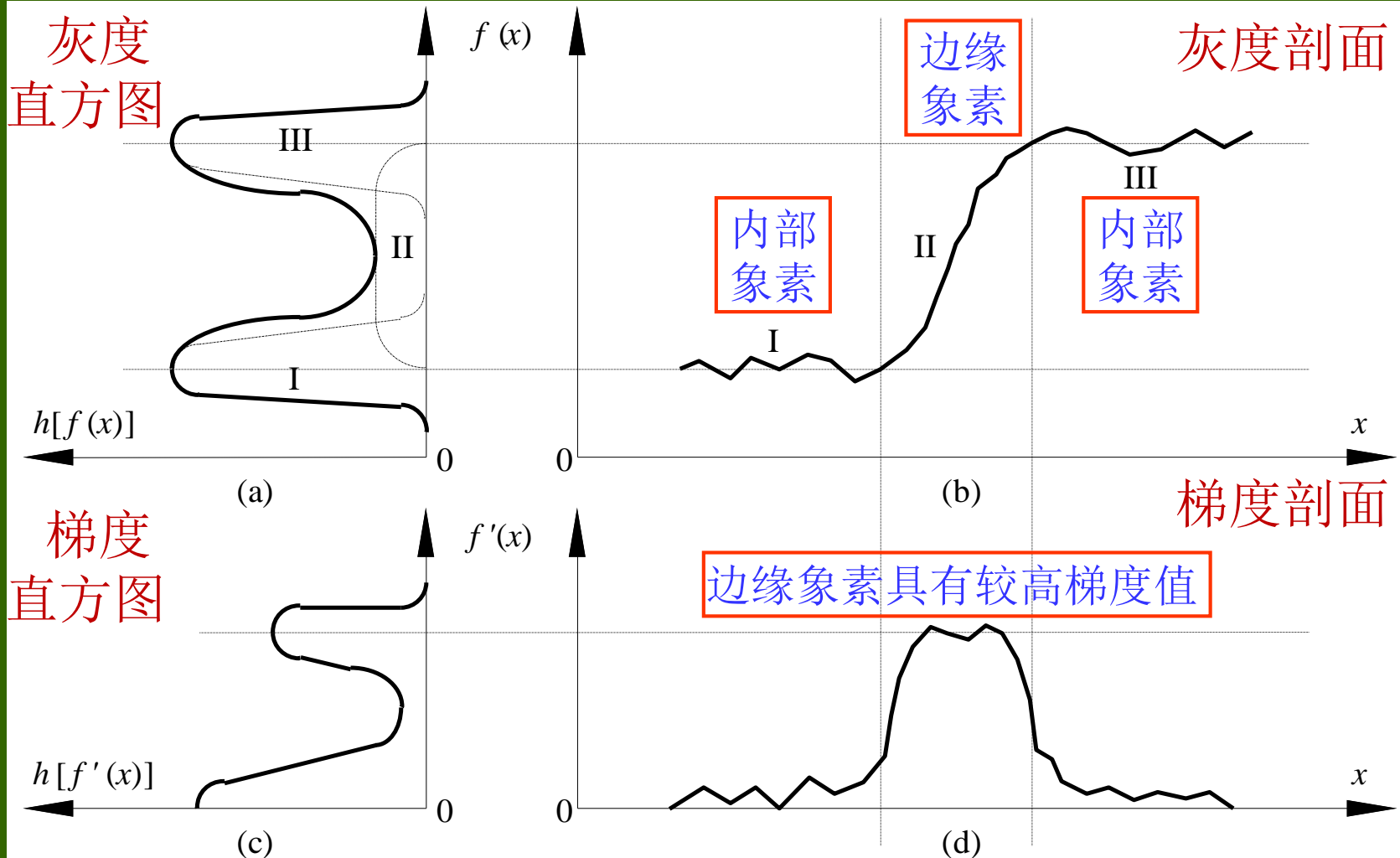
### 1. 直方图变换

- 仅利用像素灰度可能出现的问题：  
灰度直方图的谷（局部极小）被填充
- 借助邻域性质变换原来的直方图



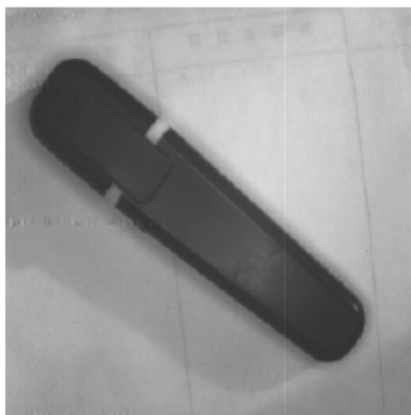
- ① 获得低梯度值像素的直方图  
峰之间的谷比原直方图深
- ② 获得高梯度值像素的直方图  
峰由原直方图的谷转化而来

## 2.4.3 依赖区域的阈值选取

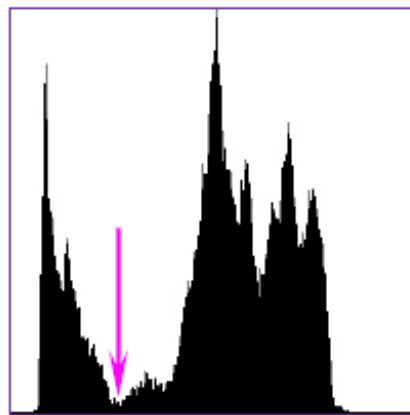


## 2.4.3 依赖区域的阈值选取

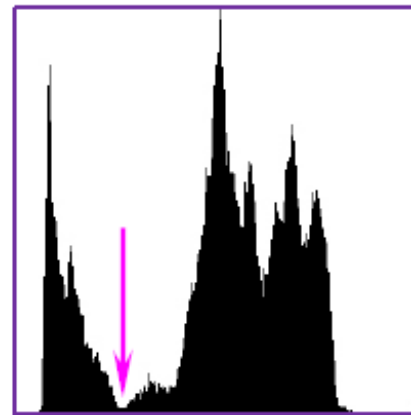
- ① 比较图(b)和图(c)可见在低梯度值直方图中谷更深了（注意箭头所指位置）
- ② 对比图(b)和图(d)可见在高梯度值直方图中其单峰基本对应原来的谷



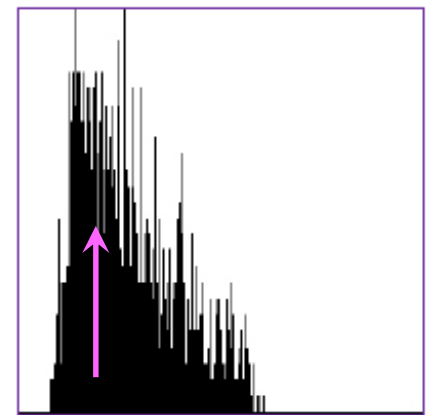
(a)



(b)



(c)



(d)

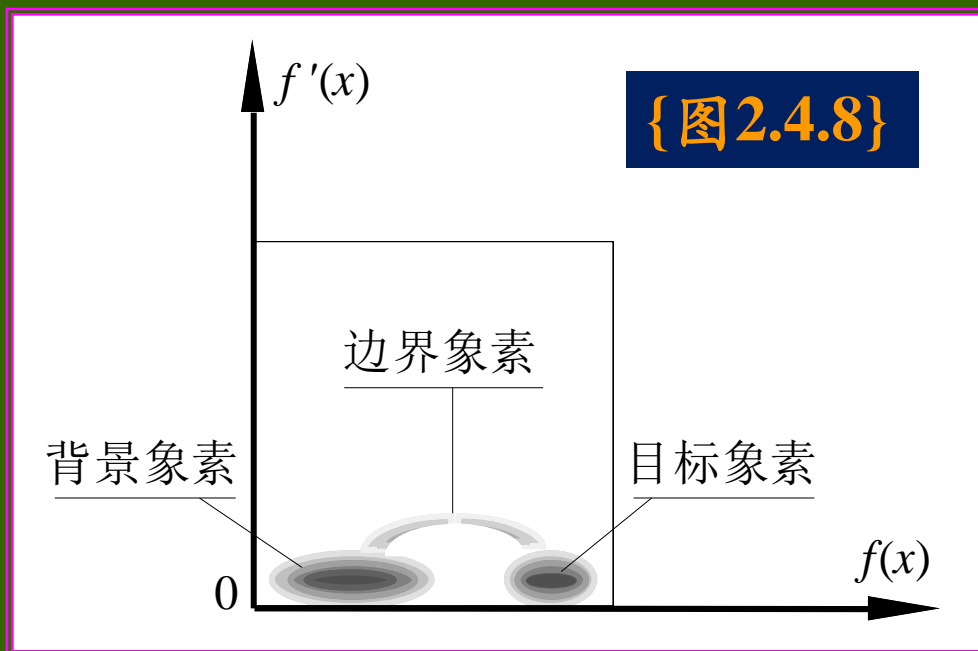
图 2.4.7 变换直方图实例

## 2.4.3 依赖区域的阈值选取

### 2. 灰度-梯度散射图

2-D直方图

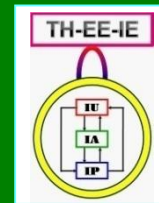
2-D图象



1个轴是灰度值轴  
1个轴是梯度值轴

示例

目标，背景聚类



## 2.4.4 依赖坐标的阈值选取

- 全局阈值不能兼顾图象各处的情况
- 用与坐标相关的一系列阈值来对图象分割

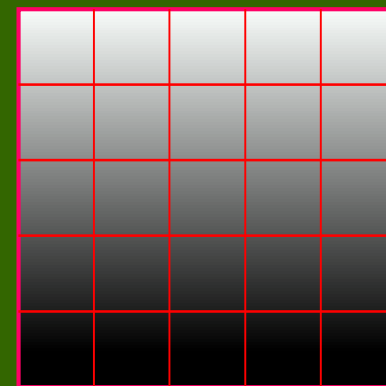
### 基本思路/步骤:

将图象分解成一系列子图象

对每个子图象计算一个阈值

对这些子图象阈值进行插值

用插值结果（阈值曲面）进行分割



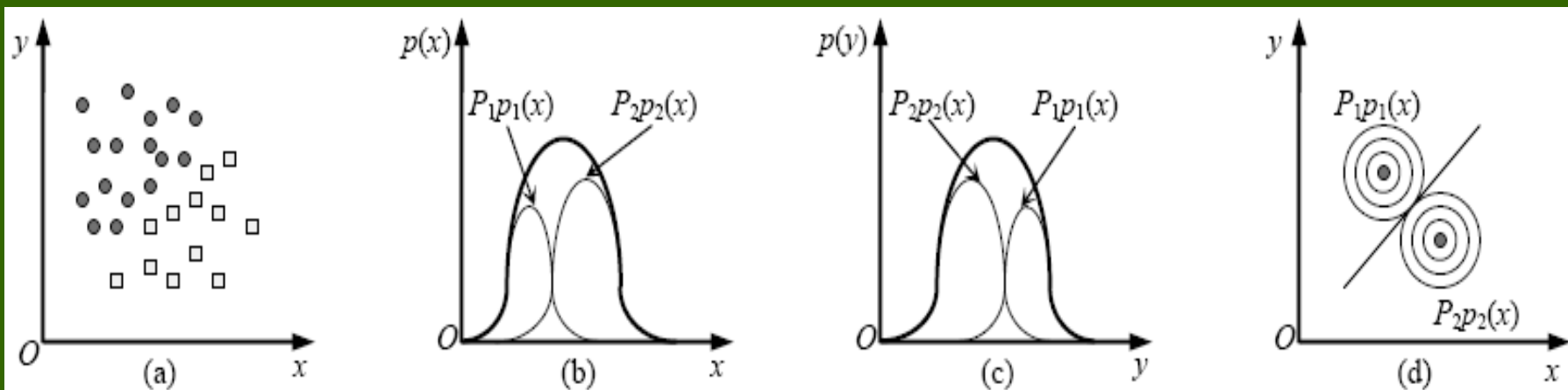
## 2.4.5 空间聚类

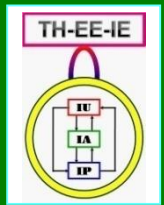
分割： $\Leftrightarrow$  像素分类

③ 并行区域类

特征空间聚类（取阈值可看作1-D聚类，  
灰度-梯度散射图可看作2-D聚类）

高维空间聚类（分类能力较强）





## 2.4.5 空间聚类

### $K$ -均值聚类

$$E = \sum_{i=1}^K \sum_{x \in Q_j^{(i)}} \|g(x) - \mu_j^{(i+1)}\|^2$$

- (1) (在特征空间) 任意选 $K$ 个初始类均值
- (2) 特征点赋类

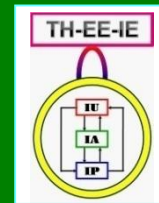
$$x \in Q_l^{(i)} \quad \text{如果} \quad \|g(x) - \mu_l^{(i)}\| < \|g(x) - \mu_j^{(i)}\|$$

- (3) 更新类均值

$$\mu_l^{(i+1)} = \frac{1}{N_l} \sum_{x \in Q_l^{(i)}} g(x)$$

- (4) 判断算法收敛

{ISODATA}



## 2.5 串行区域技术

从区域着手顺序进行  $\Rightarrow$  ④ 串行区域类

串行策略特点：

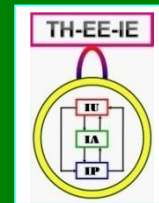
当前处理借助早期结果

优点：抗噪声，抗干扰

缺点：较复杂，费时间

### 2.5.1 区域生长

### 2.5.2 分裂合并



## 2.5.1 区域生长

基本思想:

④ 串行区域类

将相似像素结合起来构成区域

基本步骤:

- (1) 选择区域的种子像素
- (2) 确定将相邻像素包括进来的准则
- (3) 制定生长停止的规则

讨 论:

- (1) 种子像素的选取
- (2) 生长准则依赖应用

## 2.5.2 分裂合并

### 主要步骤

### ④ 串行区域类

- ◆ 首先把图象分成任意大小且不重叠的区域
- ◆ 然后再（根据准则）合并或分裂这些区域  
（迭代进行直到实现分割）

示例（四叉树）：分裂

分裂

合并

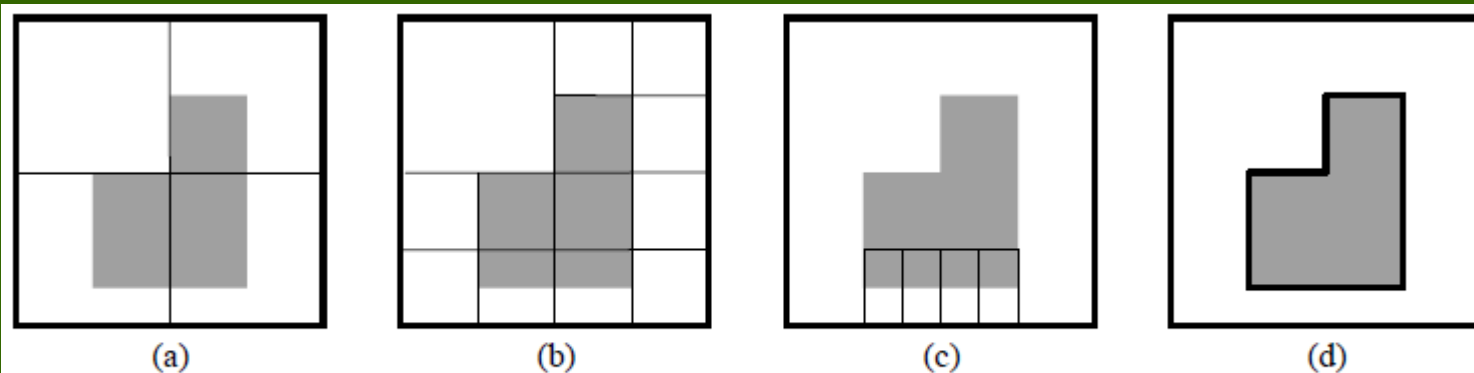
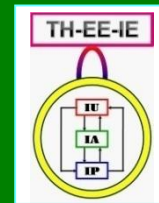


图 2.5.3 分裂合并法分割图像图解



## 2.5.2 分裂合并

- ◆ 令 $R$ 代表整个图象区域， $P$ 代表逻辑谓词
  - ◆ 把 $R$ 连续地分裂成越来越小的 $1/4$ 的正方形子区域 $R_i$ ，并且始终使 $P(R_i) = \text{TRUE}$
- (1) 对任一个区域 $R_i$ ，如果 $P(R_i) = \text{FALSE}$ ，就将其分裂成不重叠的四等分
  - (2) 对相邻的两个区域 $R_i$ 和 $R_j$ ，如果 $P(R_i \cup R_j) = \text{TRUE}$ ，就将它们合并起来
  - (3) 如果进一步的分裂或合并都不可能了，则结束



# 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/)