

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2单元 表达描述

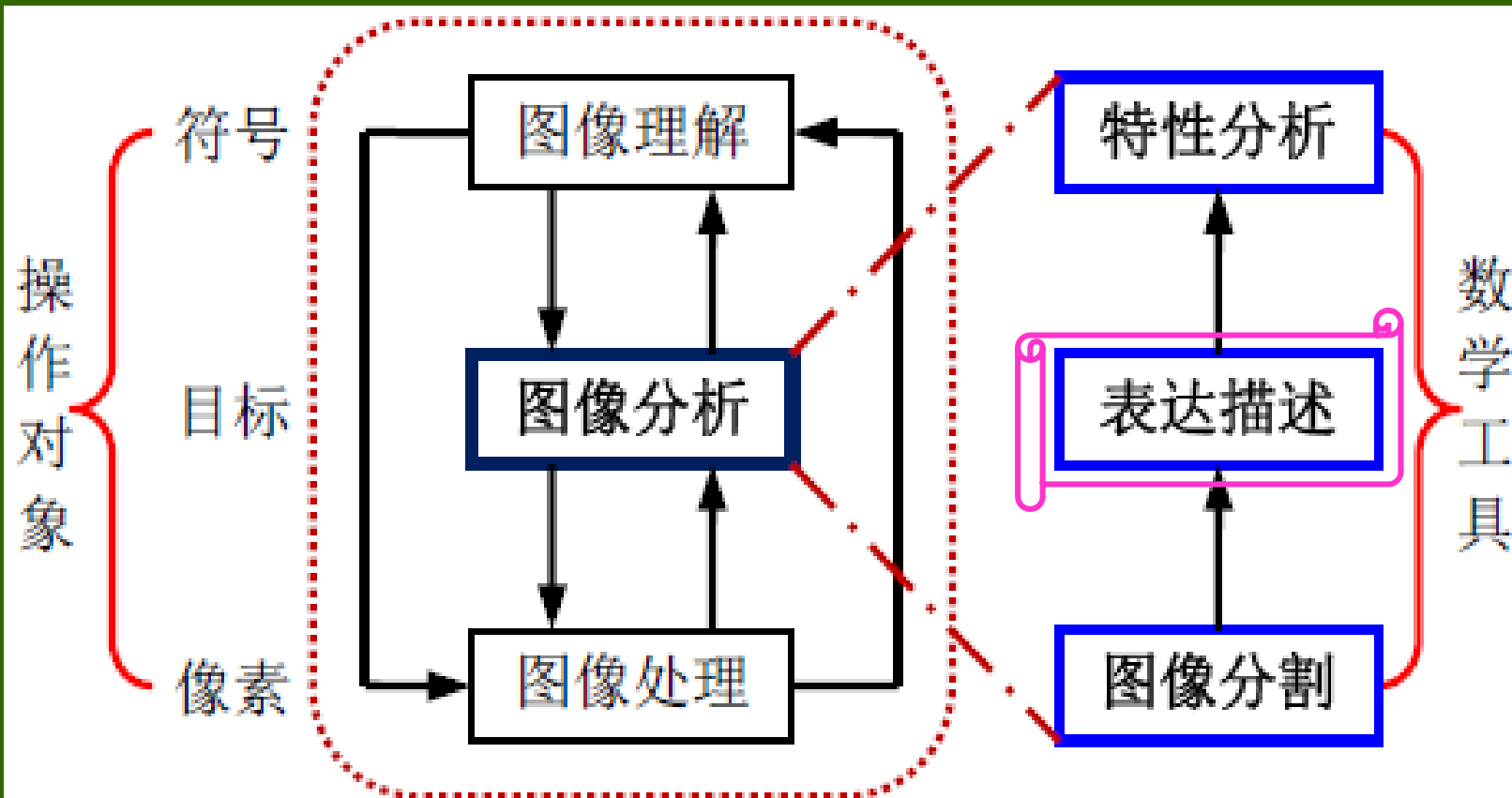
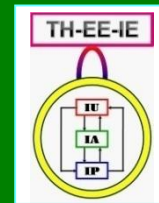


图 1.2.2 图像分析主要功能模块

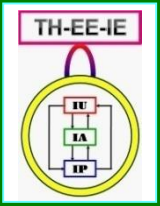


第2单元 表达描述

- 第6章 目标表达
- 第7章 目标描述
- 第8章 测量和误差分析

通过图象分割得到了图象中感兴趣的区域，即目标

为有效地刻画目标，需要对它们采取合适的数据结构进行**表达**，采用恰当的形式**描述**它们的特性，并在此基础上进行特征**测量**，从目标获得一些定量的数值以进行分析

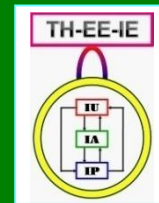


第2单元 表达描述

- 第6章 目标表达
- 第7章 目标描述
- 第8章 测量和误差分析

通过图象分割得到了图象中感兴趣的区域，即目标

为有效地刻画目标，需要对它们采取合适的数据结构进行**表达**，采用恰当的形式描述它们的特性，并在此基础上进行特征测量，从目标获得一些定量的数值以进行分析



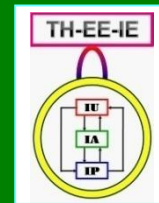
第6章 目标表达

目标表达是要对图象中的目标采取合适的数据结构来进行表示

6.1 基于边界的表达

6.2 基于区域的表达

6.3 基于变换的表达



6.1 基于边界的表达

6.1.1 技术分类

6.1.2 链码

6.1.3 边界段

6.1.4 边界标志

6.1.5 多边形

6.1.6 地标点

6.1.1 技术分类

- (1) 边界点集合：逐点，各点之间可没有顺序
- (2) 参数边界：将目标的轮廓线表示为参数曲线
- (3) 曲线逼近：用几何基元来近似

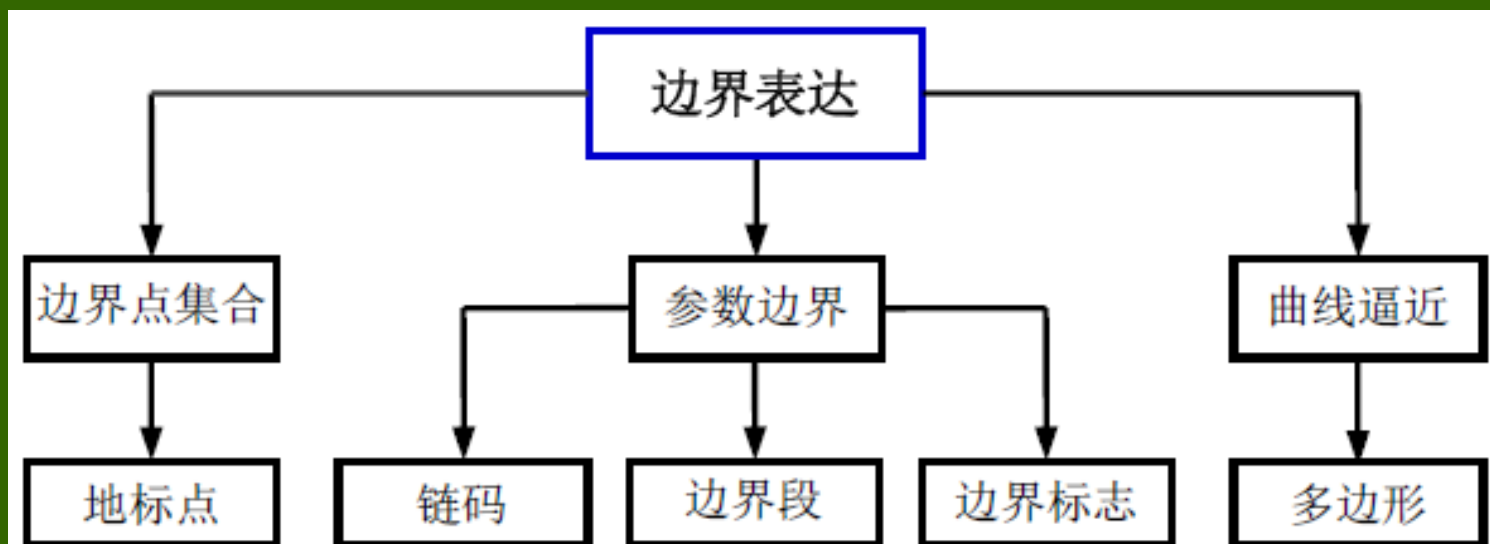


图 6.1.1 基于边界表达技术的分类

6.1.2 链码

1. 链码表达

用线段表示边界上相邻两个象素之间的联系
每个线段的长度固定而方向数目取为有限
起点用坐标表示，其余点只考虑接续方向

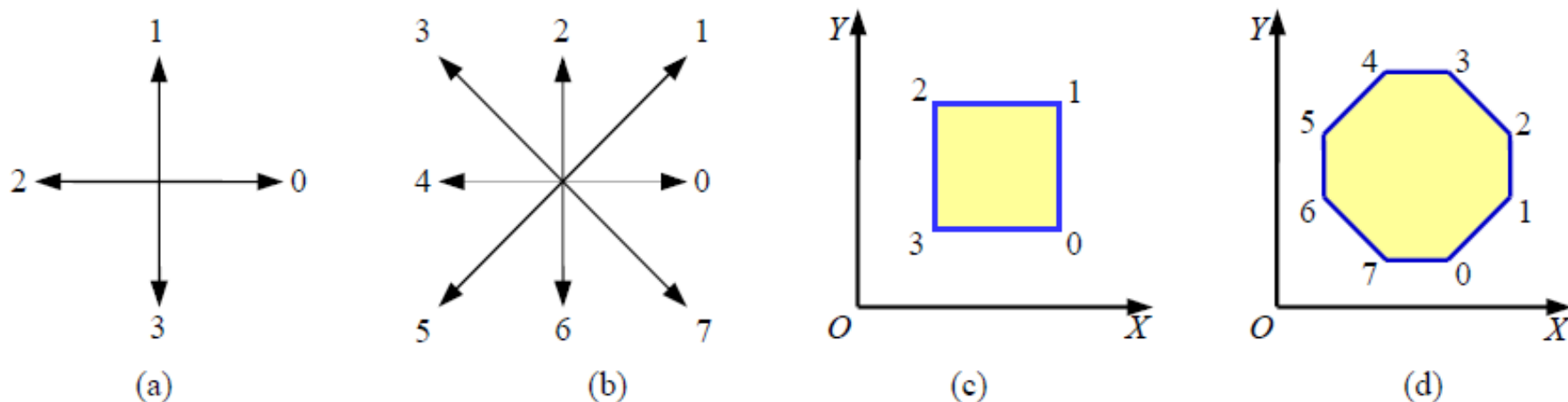


图 6.1.2 4-方向和 8-方向链码

6.1.2 链码

2. 链码归一化

① 起点归一化

将链码看作由方向数构成的自然数
选取值最小的自然数顺序

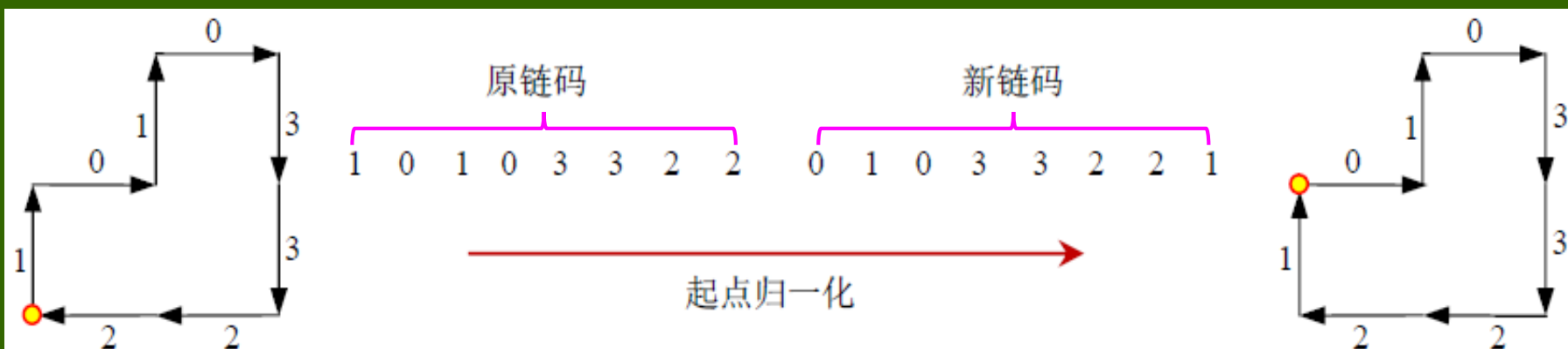


图 6.1.3 链码的起点归一化

6.1.2 链码

2. 链码归一化

② 旋转归一化

利用链码的一阶差分
差分码不随轮廓旋转而变化

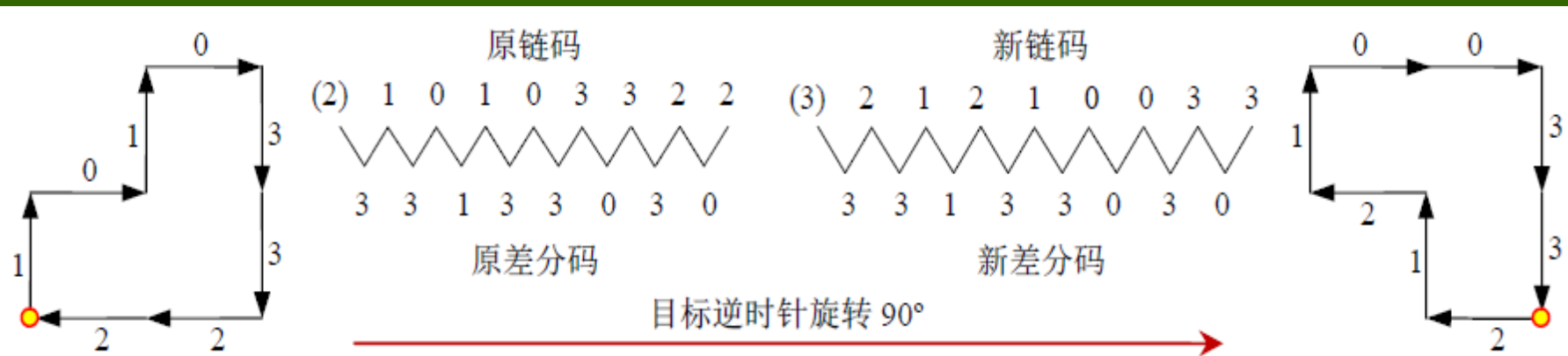


图 6.1.4 链码的旋转归一化（利用一阶差分）

6.1.2 链码

3. 平滑链码表达轮廓

将原始的链码序列用较简单的序列代替

虚线箭头：在象素 p 和 q 之间的原始 8-连通链码

实线箭头：用来替换原始序列的新序列

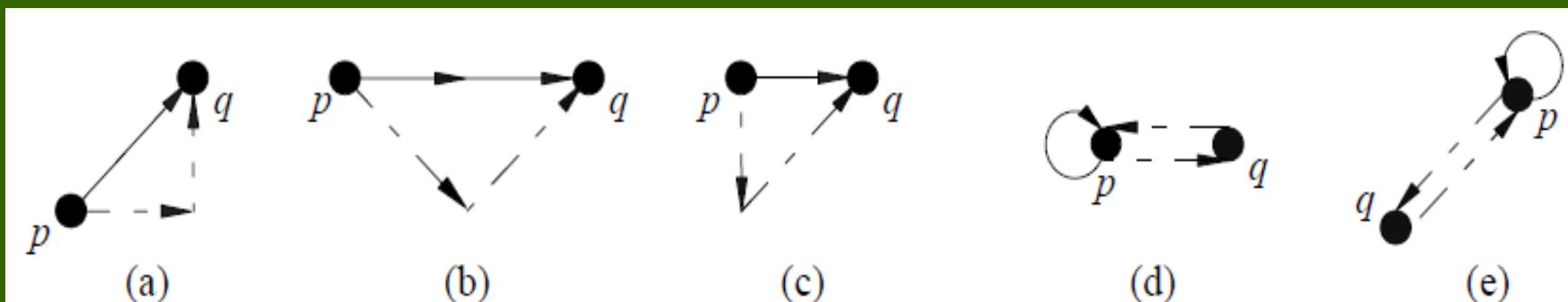


图 6.1.5 基于链码的轮廓平滑模板

6.1.2 链码

4. 缝隙码

连接相邻轮廓像素边缘的中点

红色: 外链码
(16个码)

紫色: 内链码
(14个码)

蓝色: 缝隙码
(18个码)

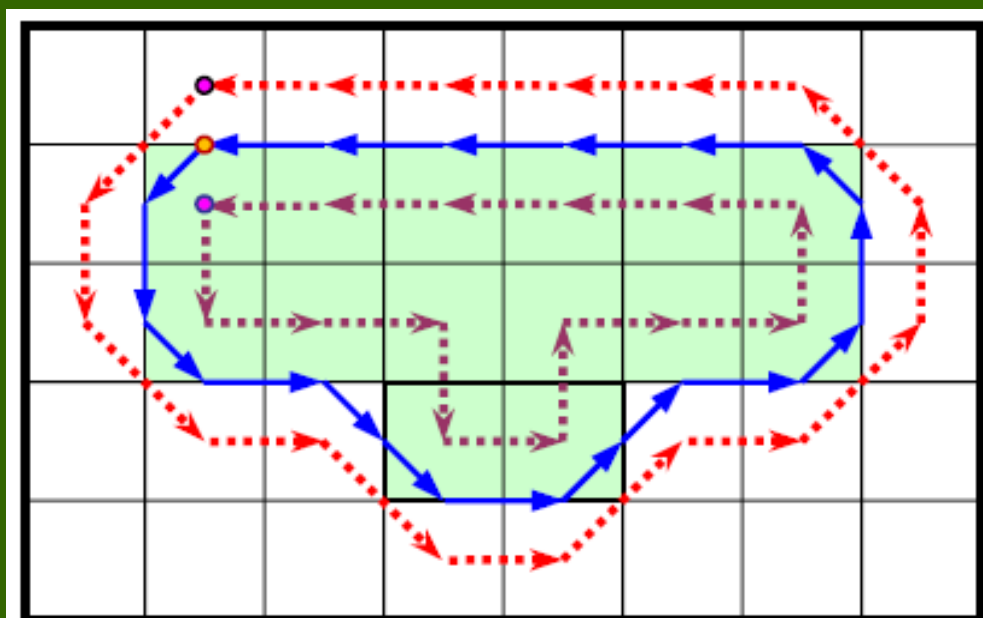


图 6.1.7 轮廓链码和中点缝隙码

6.1.3 边界段

简化表达

把边界分解成若干个（边界）段分别表示
可节省表达数据量（每段表示多个点）
借助**凸包**（包含目标的最小凸形） H

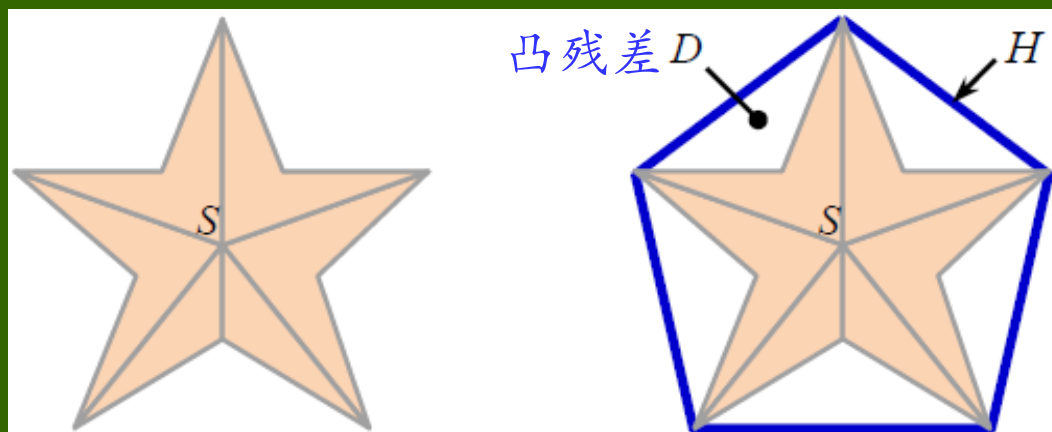


图 6.1.8 区域的凸包

6.1.3 边界段

➤ 根据凸包把边界分解

分解 { 目标: 象素集合 S
凸包: 包含 S 的最小凸形 H
凸残差: $D = H - S$

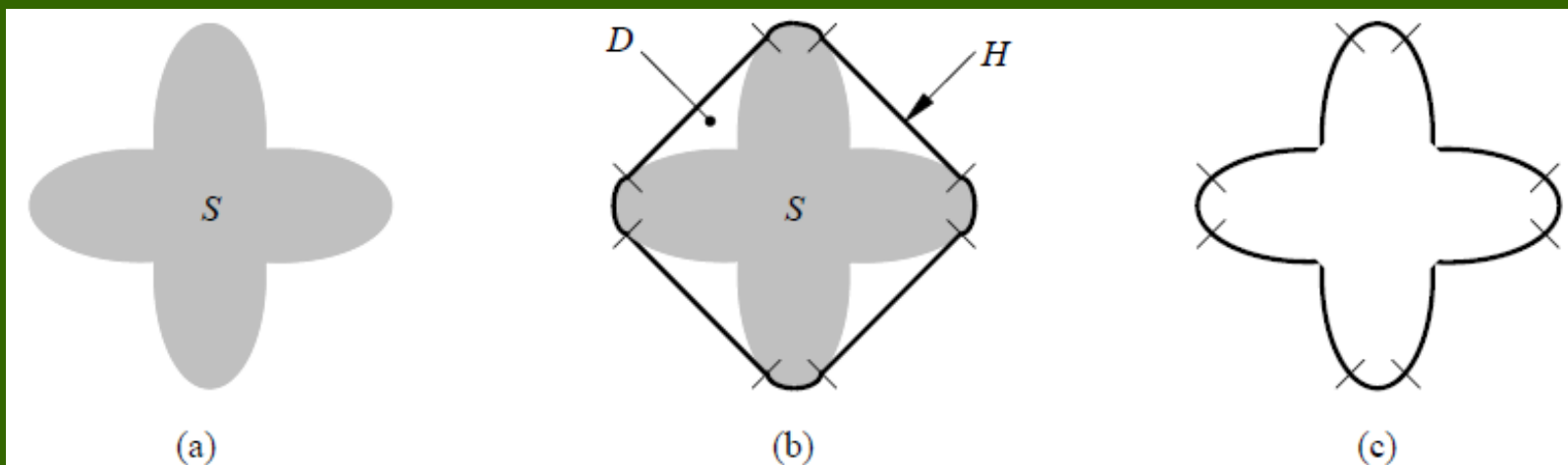
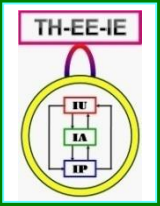


图 6.1.9 利用区域凸包将区域边界分段



6.1.4 边界标志

- 边界标志是（2-D）边界的1-D泛函表达
- 把2-D边界用较易描述的1-D函数形式来表达
- 标志可由广义的投影产生
 - 水平的、垂直的、对角线的、放射的、旋转的、等等
- 注意：投影并不是一种能保持信息的变换，将2-D平面上的区域边界变换为1-D的曲线有可能丢失信息

6.1.4 边界标志

1. 距离为角度的函数

先对给定的物体求出重心，然后把边界点与重心的距离作为（与X轴夹角）角度的函数

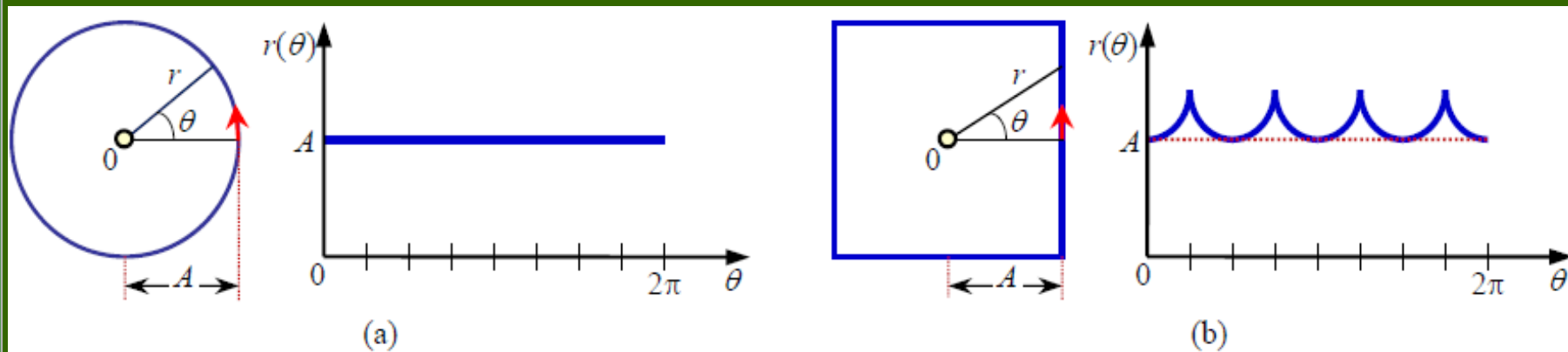


图 6.1.10 两个距离为角度函数的标志

不受目标平移影响，但随旋转或放缩而变化

6.1.4 边界标志

2. ψ - s 曲线（切线角为弧长的函数）

沿边界围绕目标一周，在每个位置作出该点切线与一个参考方向（如横轴）的角度值 {~链码}

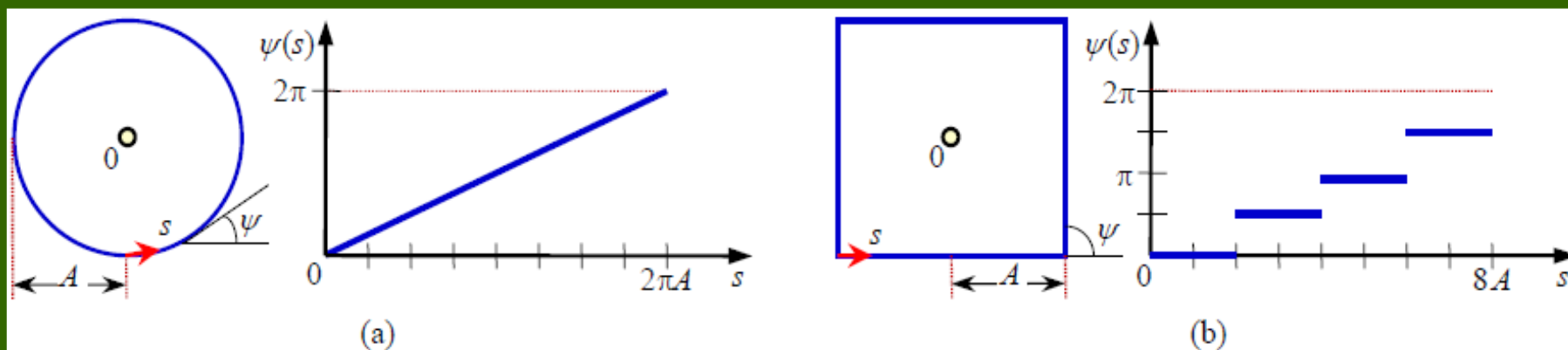


图 6.1.11 两个切线为弧长函数的标志

水平直线段对应边界上的直线段（ ψ 不变）

6.1.4 边界标志

3. 斜率密度函数

将 ψ - s 曲线沿 ψ 轴投影（累积相同的 ψ ）
切线角的直方图 $h(\theta)$

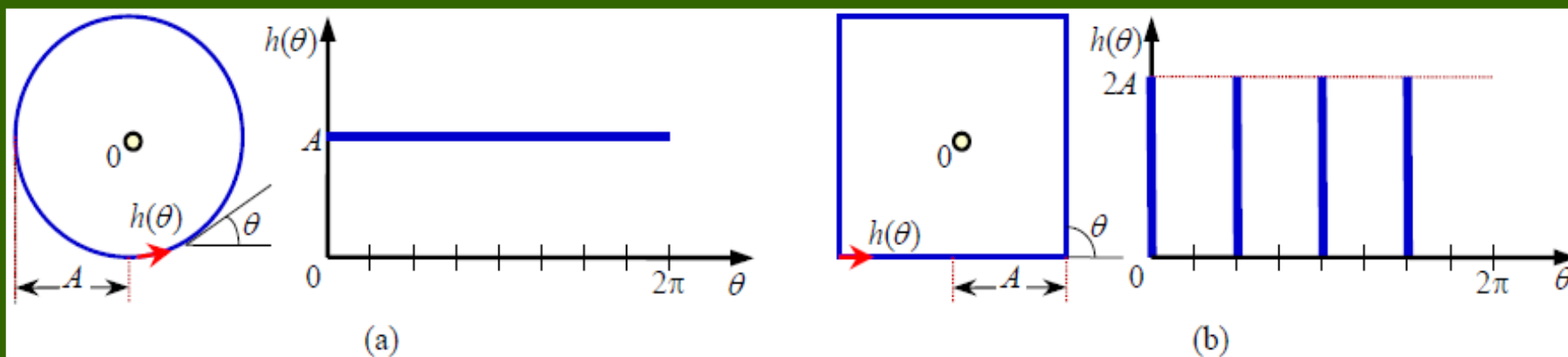


图 6.1.12 两个斜率密度函数的标志

对具有常数切线角的边界段有比较强的响应

6.1.4 边界标志

4. 距离为弧长的函数

将各个边界点与目标重心的距离作为边界点序列（围绕目标得到）的函数

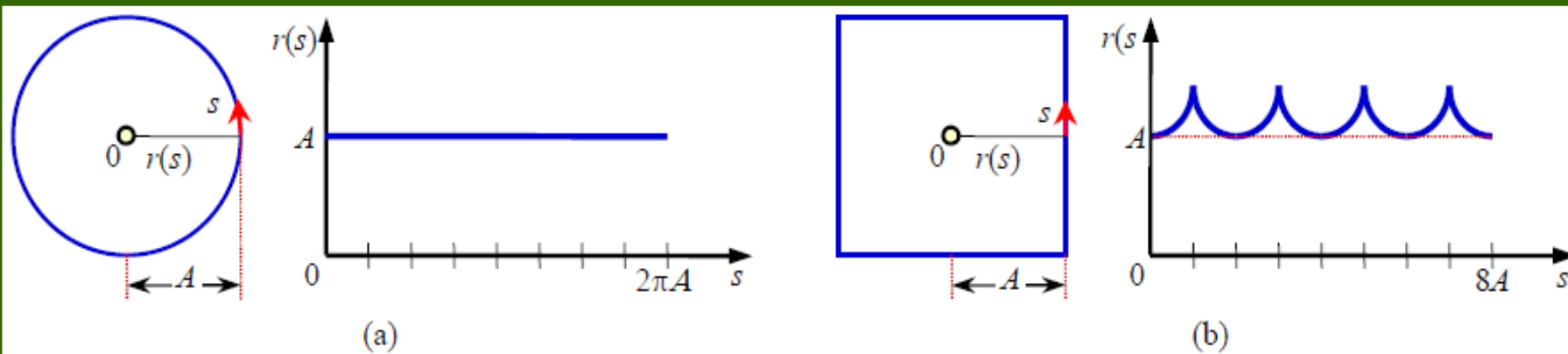
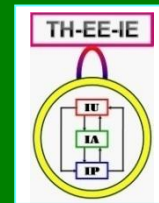


图 6.1.13 两个距离为弧长函数的标志

与距离为角度的函数相比？



6.1.5 多边形

用多边形去近似逼近边界

常见的三种方法：

- (1) 基于收缩的最小周长多边形法
- (2) 基于聚合 (merge) 的最小均方误差线段逼近法
- (3) 基于分裂 (split) 的最小均方误差线段逼近法

6.1.5 多边形

(1) 基于收缩的最小周长多边形法 {P.138}

将原边界看成是有弹性的线，将组成边界的像素序列的内外边各看成一堵墙，将线拉紧

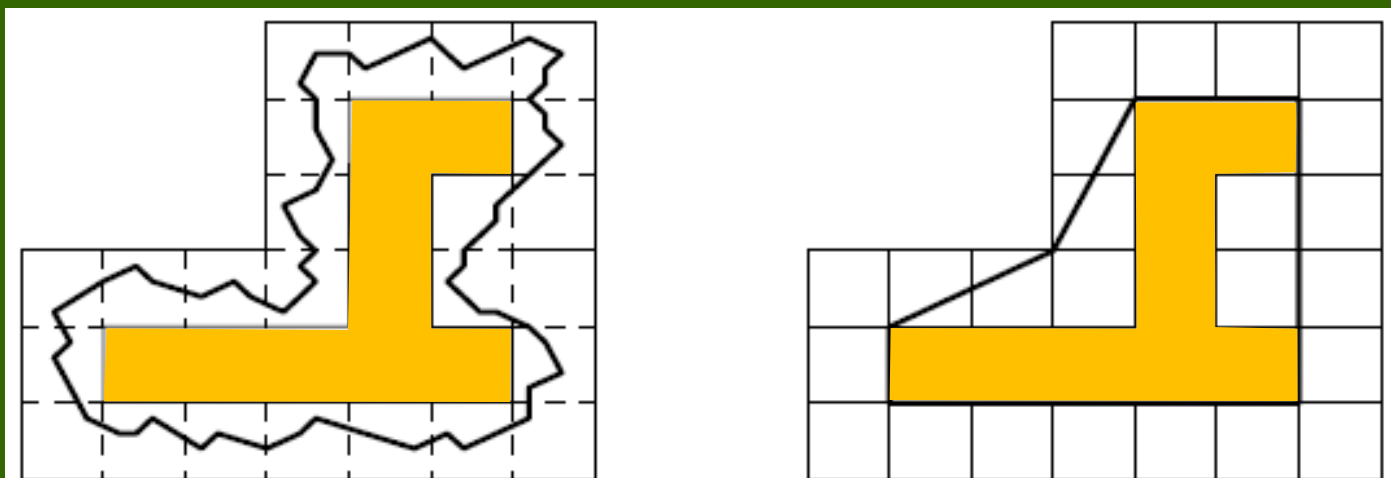


图 6.1.14 最小周长多边形法

6.1.5 多边形

(2) 基于聚合的最小均方误差线段逼近法

{P.138}

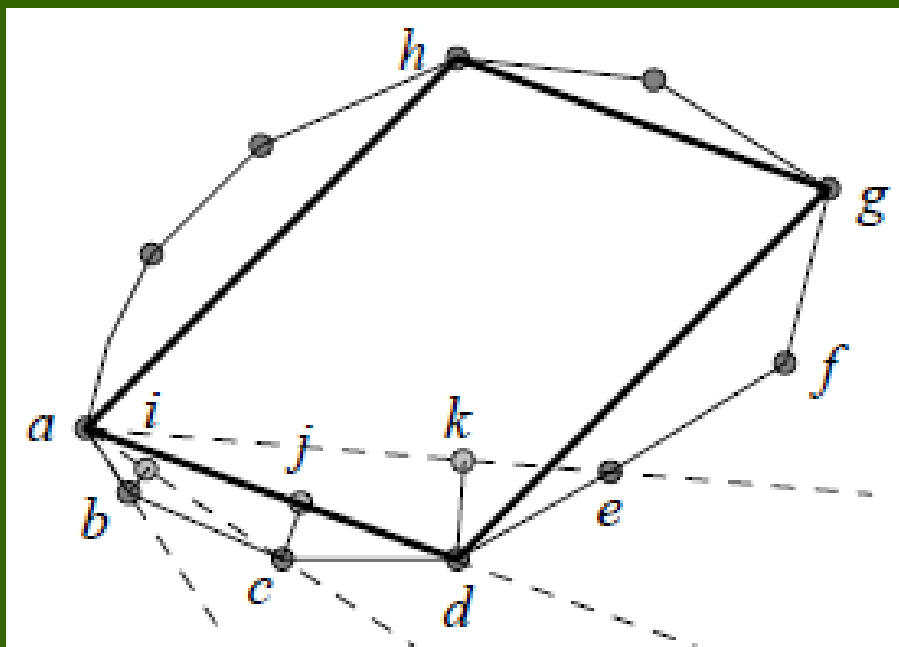


图 6.1.15 聚合逼近多边形

6.1.5 多边形

(3) 基于分裂的最小均方误差线段逼近法

{P.138}

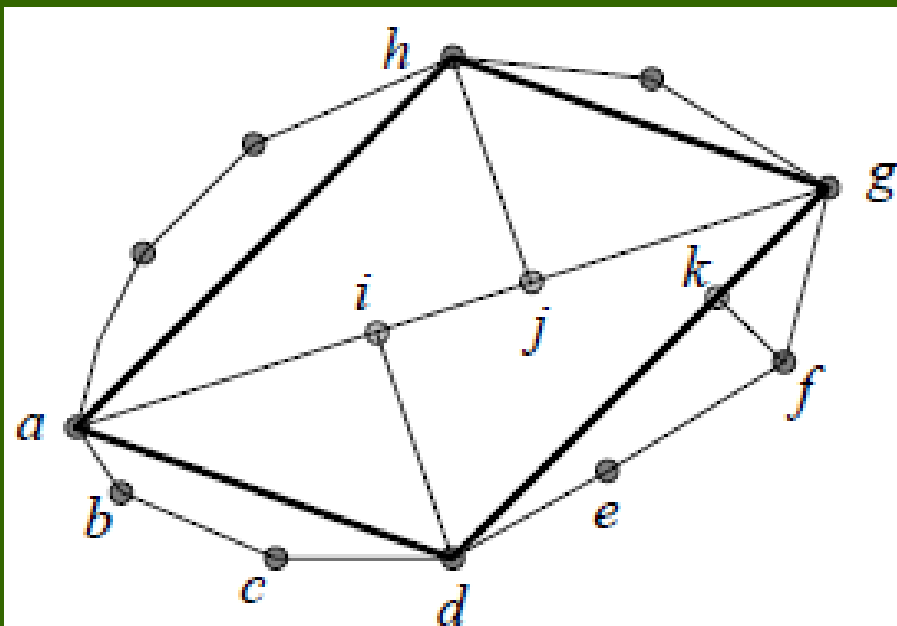
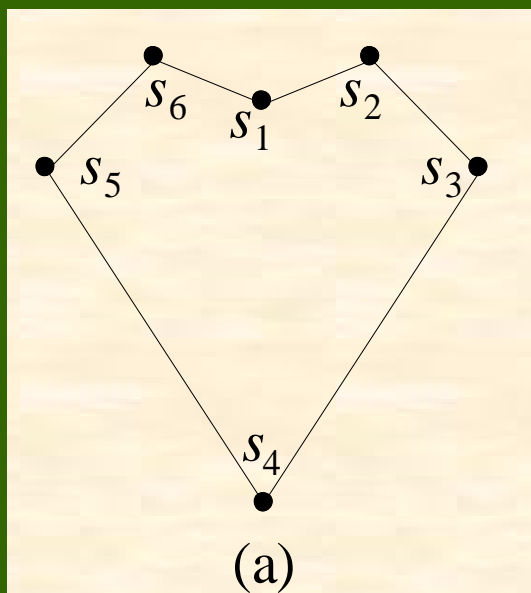


图 6.1.16 分裂逼近多边形

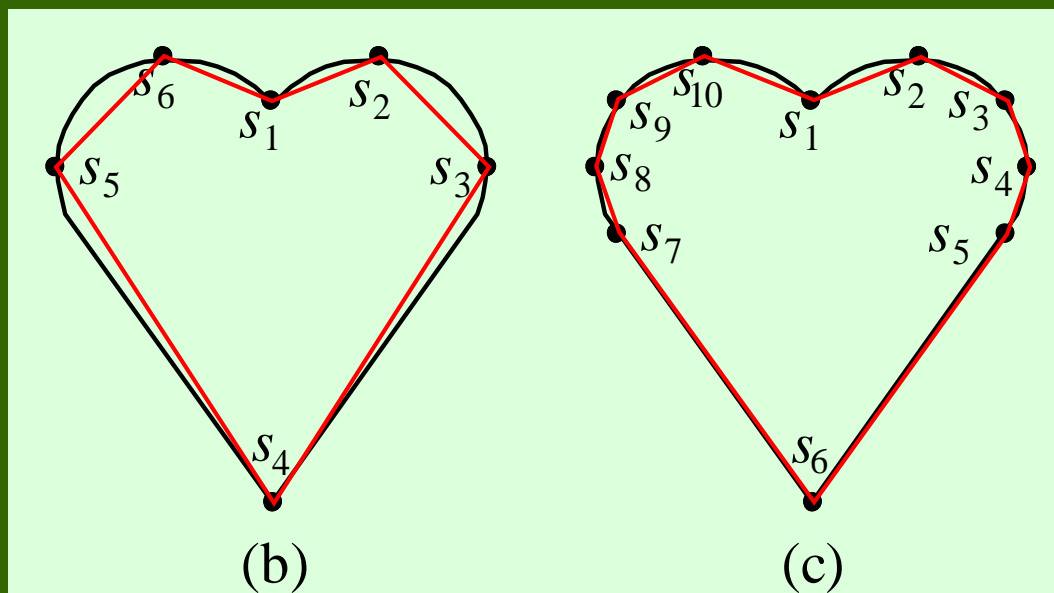
6.1.6 地标点

地标点或标志点 (Landmark Points)

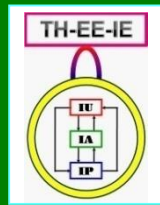
一种近似表达方法，点越多近似得越好



准确表达



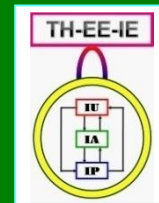
近似表达



6.1.6 地标点

具有顶点 $S_1 = (1, 1)$, $S_2 = (1, 2)$, $S_3 = (2, 1)$ 的三角形

方式	表达	解释
$2n$ -矢量	$S_o = [1, 1, 1, 2, 2, 1]$	S_o 是一个 $2n \times 1$ 的实坐标矢量
$2n$ -集合	$S_f = \{1, 1, 1, 2, 2, 1\}$	S_f 是一个包含 $2n$ 个实坐标的集合
矢量-平面	$S_v = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \end{bmatrix}$	S_v 是一个 $n \times 2$ 的矩阵, 每行包含一个地标点的 x -和 y -实坐标
复数-平面	$S_c = \begin{bmatrix} 1+j \\ 1+2j \\ 2+j \end{bmatrix}$	S_c 是一个 $n \times 1$ 的复数矢量, 每个复数表示一个地标点的 x -和 y -坐标



6.2 基于区域的表达

6.2.1 技术分类

6.2.2 空间占有数组

6.2.3 四叉树

6.2.4 金字塔

6.2.5 围绕区域

6.2.6 骨架

6.2.1 技术分类

- (1) 区域分解：简单的单元形式
- (2) 围绕区域：外接圆，外包围矩形，……
- (3) 内部特征：内部像素的特定抽象集合

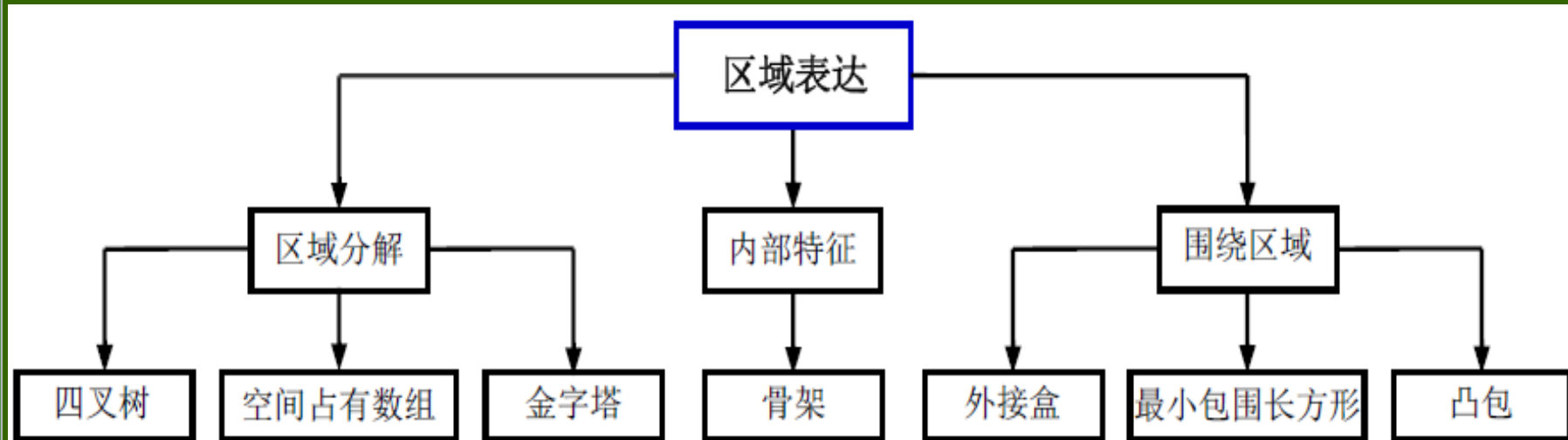
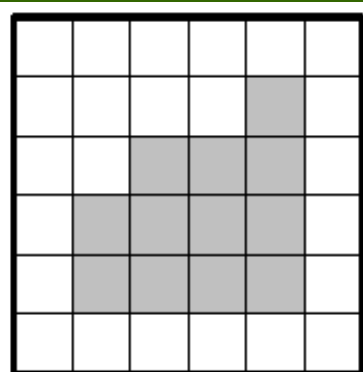


图 6.2.1 基于区域表达技术的分类

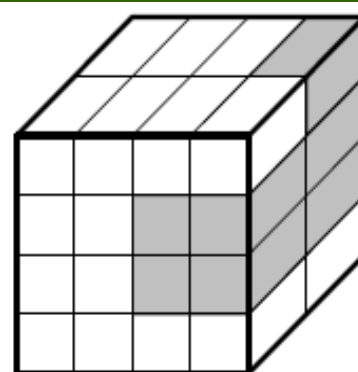
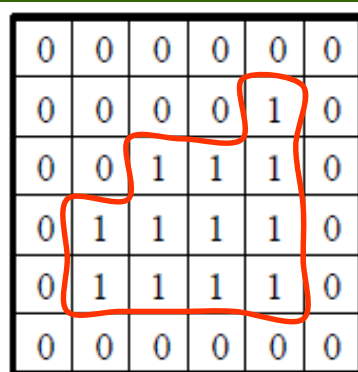
6.2.2 空间占有数组

对图象 $f(x, y)$ 中任一点 (x, y) :

如果它在给定的区域内, 就取 $f(x, y)$ 为1
否则就取 $f(x, y)$ 为0



(a)



(b)

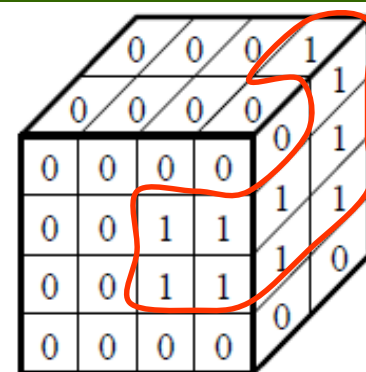
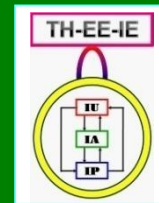


图 6.2.2 空间占有数组表达示例

所有 $f(x, y)$ 为1的点组成的集合就代表了所要表示的区域



6.2.3 四叉树

基本思路

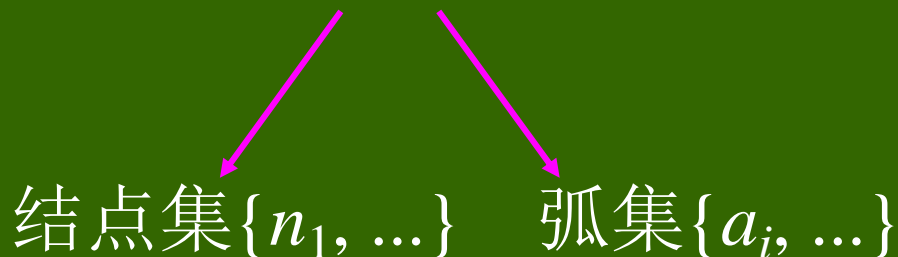
分层分解图象

利用金字塔式的数据结构

四叉树表达法

每次将图象一分为四

树结构: $T = \{N, A\}$



6.2.3 四叉树

四叉树表达图示

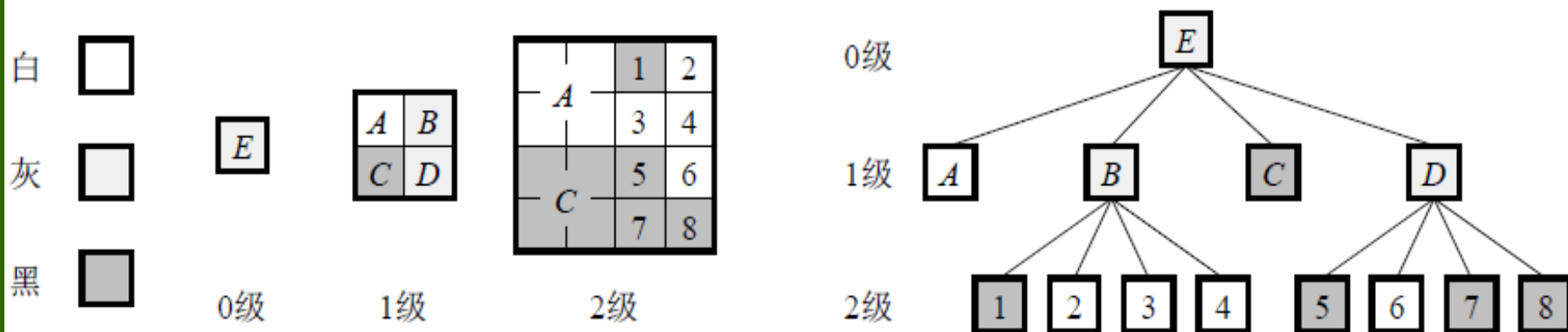


图 6.2.3 四叉树表达图示

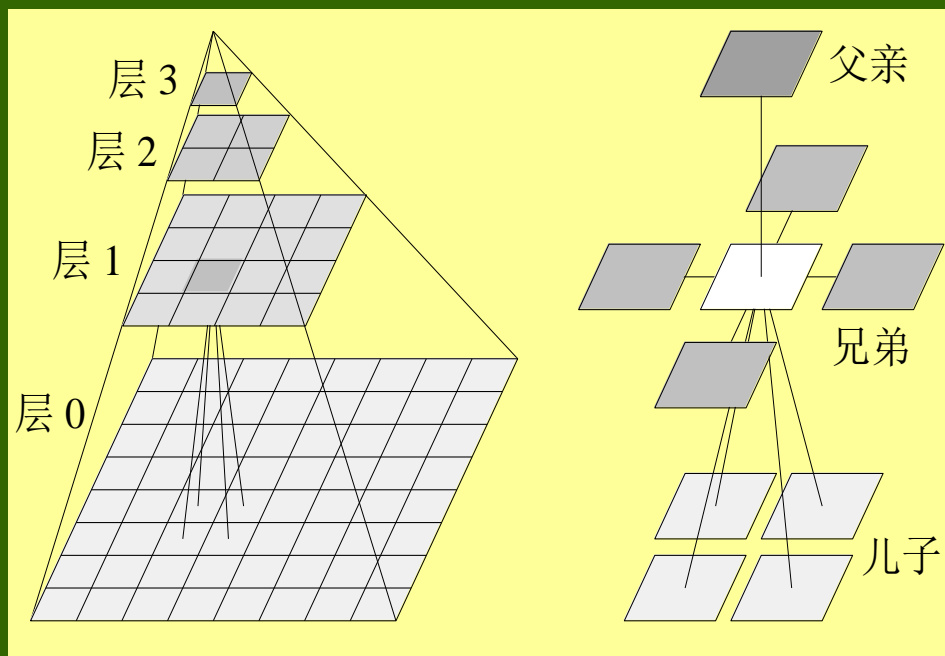
结点数目

$$N = \sum_{k=0}^n 4^k = \frac{4^{n+1} - 1}{3} \approx \frac{4}{3} 4^n$$

6.2.4 金字塔

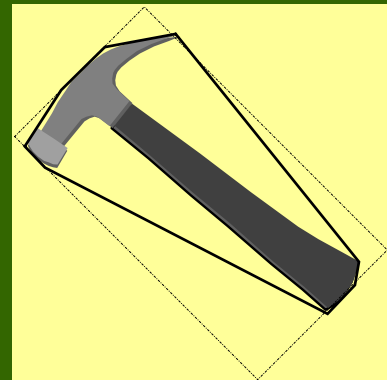
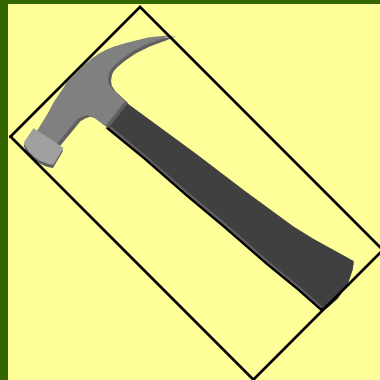
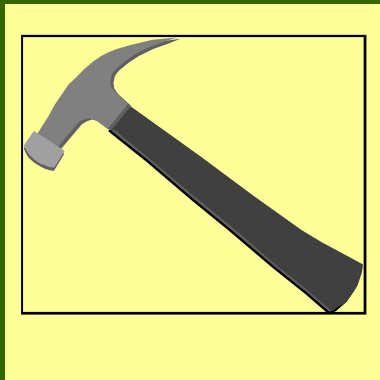
- 金字塔结构可借助图（graph）来解释
- 金字塔结构由各层内的“兄弟/邻居”关系和各层间的“父子”关系所确定

- 用 $(n \times n)/r$ 描述金字塔结构
- 缩减率 r
- 缩减窗（ $n \times n$ 方窗）



6.2.5 围绕区域

- (1) 外接盒 (Feret box) : 包含目标区域的最小长方形 (朝向特定的参考方向)
- (2) 围盒 (minimum enclosing rectangle, MER) : 包含目标区域的 (可朝向任何方向) 最小长方形
- (3) 凸包: 包含目标区域的最小凸多边形



6.2.5 围绕区域

计算凸包

目标用 n 个顶点序列确定的多边形表示

(a) 先从顶点序列获得前3个顶点，构成一个三角形；(b) 下一个点 D 在三角形内；(c) D 在三角形外但没有原来的顶点需要移去；(d) 有一个原来的顶点需要从当前凸包中移去

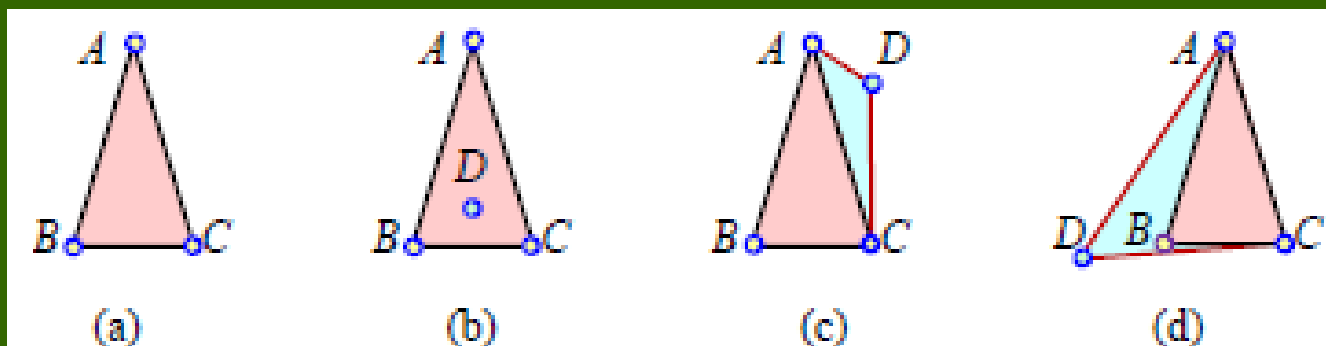
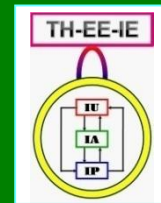


图 6.2.7 凸包检测示意



6.2.6 骨架

1. 骨架的定义和特点

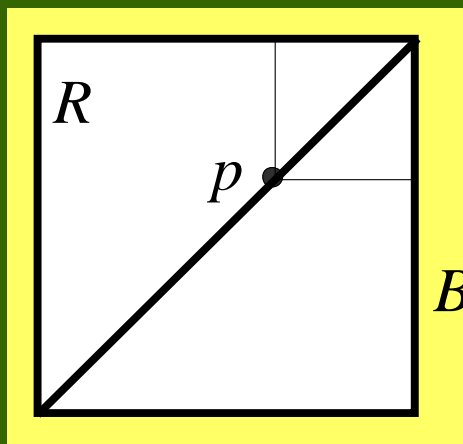
➤ 骨架点

与（两个）轮廓点距离都最小的点

$$d_s(p, B) = \inf\{d(p, b) \mid b \in B\}$$

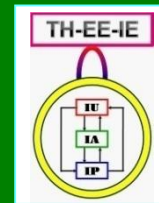
➤ 骨架点的确定

- 区域 R
- 轮廓 B
- 骨架点 p



{P.147}

(图6.2.9)



6.2.6 骨架

2. 计算骨架的一种实用方法

轮廓点/边界点:

- ① 本身属于区域（标记为1）
- ② 邻域中不属于区域的点（标记为0）

8-邻域

$$N(p_1) = 5$$

$$S(p_1) = 1$$

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

1	1	1
1	p_1	1
0	0	0

1	1	1
1	p_1	1
0	0	0

6.2.6 骨架

2. 计算骨架的一种实用方法 •消除标记的点

不满足

$$(1.1) \quad 2 \leq N(p_1) \leq 6$$

$$(1.2) \quad S(p_1) = 1$$

$$(1.3) \quad p_2 \cdot p_4 \cdot p_6 = 0$$

$$(1.4) \quad p_4 \cdot p_6 \cdot p_8 = 0$$

$$(2.1) \quad 2 \leq N(p_1) \leq 6$$

$$(2.2) \quad S(p_1) = 1$$

$$(2.3) \quad p_2 \cdot p_4 \cdot p_8 = 0$$

$$(2.4) \quad p_2 \cdot p_6 \cdot p_8 = 0$$

满足

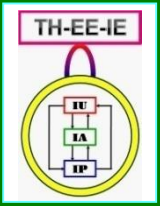
0	1	0
0	1	0
0	0	0

1	1	1
1	1	1
1	0	1

0	0	1
0	1	0
1	0	0

0	0	0
0	1	1
0	1	1

1	1	0
1	1	0
0	0	0



6.3 基于变换的表达

利用一定的变换将目标从图象空间变换到新空间（变换空间）并利用变换参数来表达目标

6.3.1 技术分类

6.3.2 傅里叶变换表达

6.3.1 技术分类

- (1) 线性：采用的变换技术是线性的，则表达也是线性的
- (2) 非线性：采用的变换技术是非线性的，则表达也是非线性的

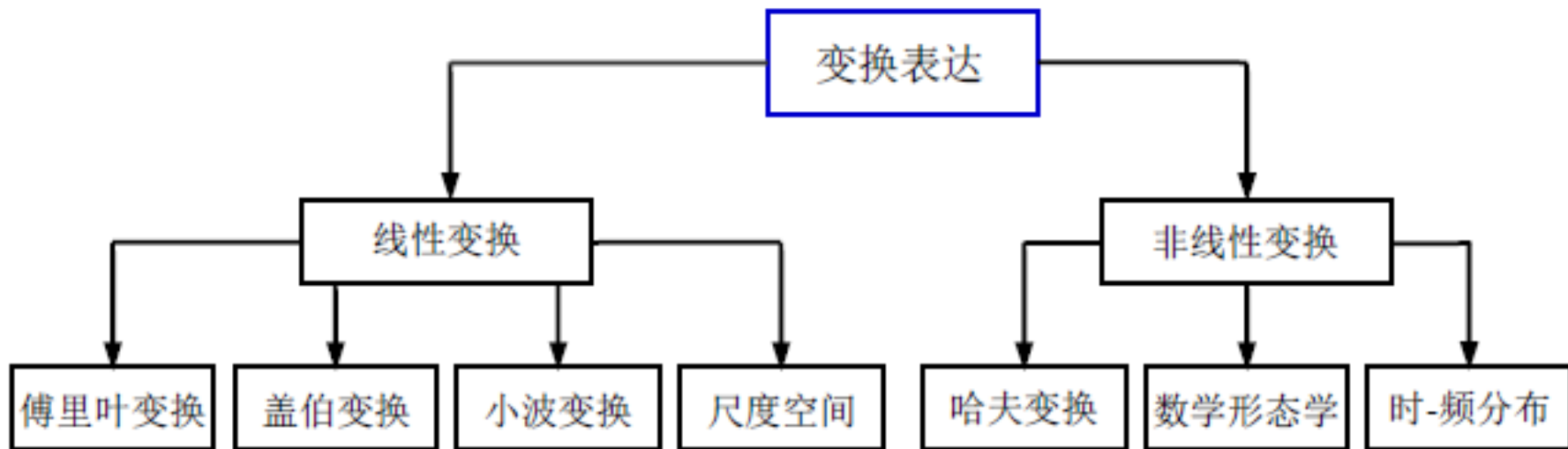
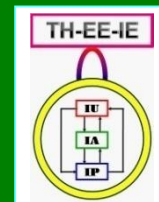


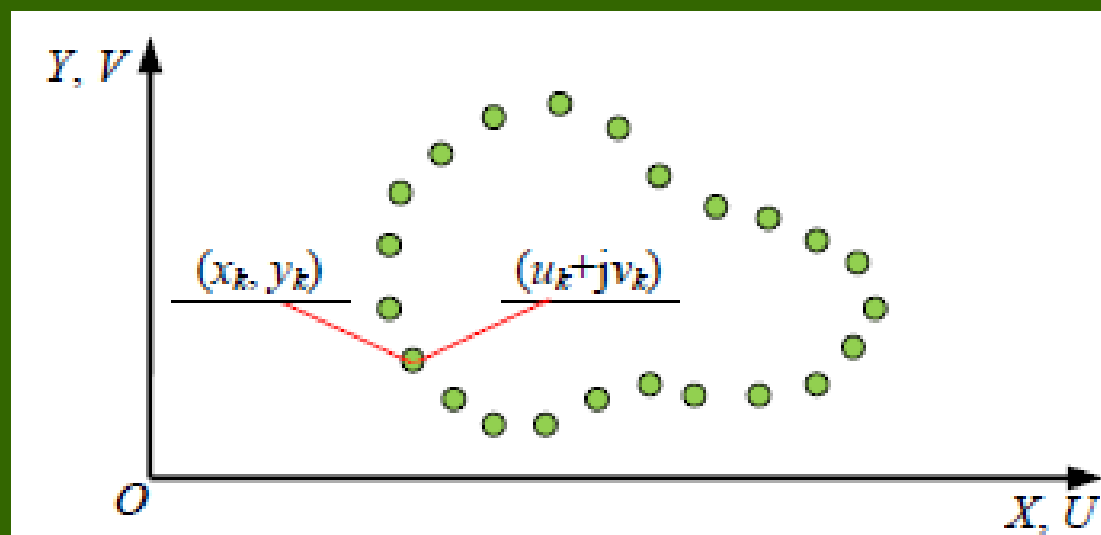
图 6.3.1 基于变换表达技术的分类



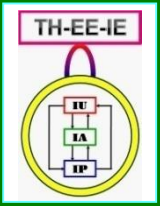
6.3.2 傅里叶变换表达

对目标边界的离散傅里叶变换表达

将 XY 平面中的曲线段转化为
复平面 UV 上的点序列



将2-D的问题简化为1-D的问题



6.3.2 傅里叶变换表达

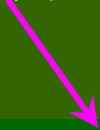
- 从1个封闭边界可得到1个复数序列

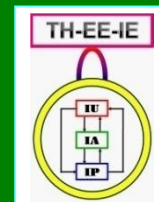
$$s(k) = u(k) + jv(k) \quad k = 0, 1, \dots, N-1$$

- 将序列进行傅里叶变换（数据量不变）

$$S(w) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) \exp[-j2\pi wk / N] \quad w = 0, 1, \dots, N-1$$

- 取（部分）傅里叶变换系数表达轮廓
（节省数据量）


$$\hat{s}(k) = \sum_{w=0}^{M-1} S(w) \exp[j2\pi wk / N] \quad k = 0, 1, \dots, N-1$$



6.3.2 傅里叶变换表达

傅里叶变换表达受边界平移、旋转、尺度变换以及计算起点（傅里叶表达与从边界点建立复数序列对的起始点有关）的影响情况 {上册4.2.2小节}

变换/变化	边界点序列	傅里叶变换系数序列
平移($\Delta x, \Delta y$)	$s_t(k) = s(k) + \Delta xy$	$S_t(w) = S(w) + \Delta xy \bullet \delta(w)$
旋转(θ)	$s_r(k) = s(k) \exp(j\theta)$	$S_r(w) = S(w) \exp(j\theta)$
尺度(C)	$s_c(k) = C \bullet s(k)$	$S_c(w) = C \bullet S(w)$
起点(k_0)	$s_p(k) = s(k - k_0)$	$S_p(w) = S(w) \exp(-j2\pi k_0 w / N)$



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/