数字图像处理

Digital Image Processing

表示与描述

- 概述
- 表示方法
- 边界描述子
- 关系描述子

- 概述

✓ 图像分割结果是得到了区域内的像素集合，或位于区域边界上的像素集合，这两个集合是互补的

✓ 与分割类似，图像中的区域可用其内部(如组成区域的像素集合)表示，也可用其外部（如组成区域边界的像素集合）表示

✓ 一般来说，如果关心的是区域的反射性质，如灰度、颜色、纹理等，常用内部表示法；如果关心的是区域形状，则选用外部表示法

✓ 表示是直接具体地表示目标，好的表示方法应具有节省存储空间、易于特征计算等优点

表示与描述

- 概述

✓ 描述是较抽象地表示目标。好的描述应在尽可能区别不同目标的基础上，对目标的尺度、平移、旋转等不敏感，这样的描述比较通用

✓ 描述可分为对边界的描述和对区域的描述。此外，边界和边界或区域和区域之间的关系也常需要进行描述

✓ 表示和描述是密切联系的。表示的方法对描述很重要，因为它限定了描述的精确性；而通过对目标的描述，各种表示方法才有实际意义

✓ 表示和描述又有区别，表示侧重于数据结构，而描述侧重于区域特性以及不同区域间的联系和差别

- 概述

✓ 对目标特征的测量是要利用分割结果进一步从图像中获取有用信息，为达到这个目的需要解决两个关键问题：

  - 选用什么特征来描述目标
  - 如何精确地测量这些特征

✓ 常见的目标特征分为灰度（颜色）、纹理和几何形状特征等。其中，灰度和纹理属于内部特征，几何形状属于外部特征

表示与描述

- 概述
- 表示方法
- 边界描述子
- 关系描述子

表示与描述

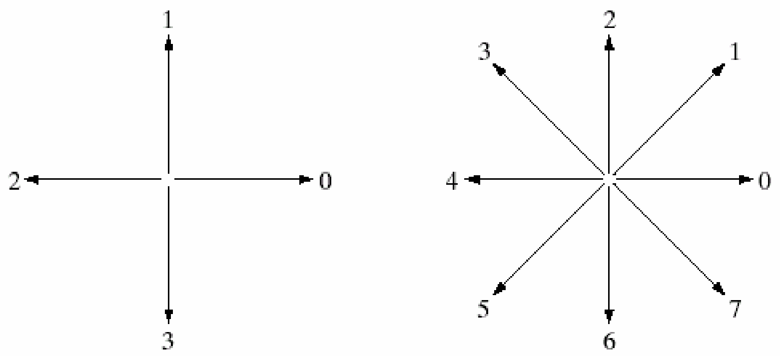
- 表示方法
  - ✓ 链码
  - ✓ 多边形近似
  - ✓ 外形特征
  - ✓ 边界分段
  - ✓ 区域骨架

- 链码
  - 链码用于表示由顺序连接的具有指定长度和方向的直线段组成的边界线
  - 这种表示方法基于线段的4或8连接
  - 每一段的方向使用数字编号方法进行编码

4链码　　8链码

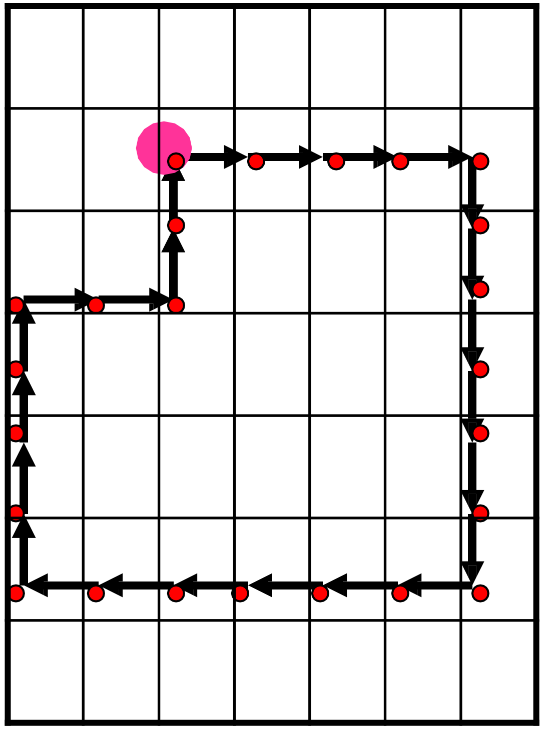
a b

FIGURE 11.1 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.



表示与描述

- 链码举例：



4-链码：000033333322222211110011

表示与描述

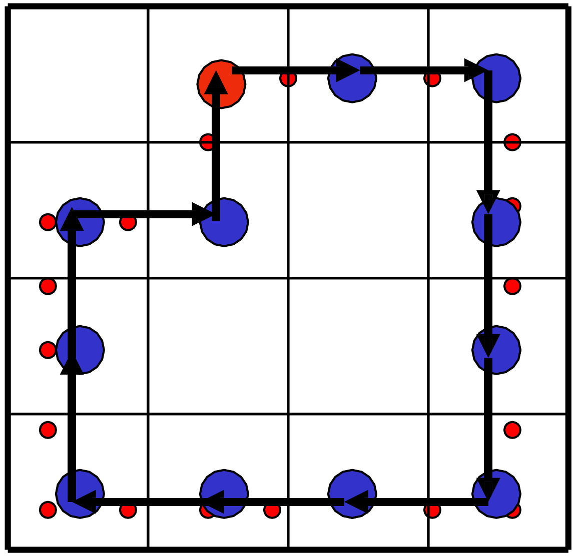
- 链码
  - ✓ 算法：
    - 给每一个线段边界一个方向编码
    - 有4链码和8链码两种编码方法
    - 从起点开始，沿边界编码，至起点被重新碰到，结束一个对象的编码

■ 链码

- 问题1：
  - ✓1）链码相当长
  - ✓2）噪音会产生不必要的链码
- 改进1：
  - ✓1）加大网格空间
  - ✓2）依据原始边界与结果的接近程度，来确定新点的位置

表示与描述

- 链码举例：



4-链码：003332221101

■ 链码

➢问题2：

✓1）由于起点的不同，造成编码的不同

✓2）由于角度的不同，造成编码的不同

➢改进2：

✓1）从固定位置作为起点(最左最上)开始编码

✓2）通过使用链码的差分代替码字本身的方式

- 链码
  - ✓ 循环差分链码：用相邻链码的差代替链码

例如：4-链码　10103322

循环差分为：　33133030

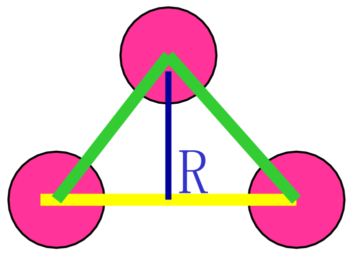
循环差分：	1 - 2 = -1(3)	3 - 0 = 3
	0 - 1 = -1(3)	3 - 3 = 0
	1 - 0 = 1	2 - 3 = -1(3)
	0 - 1 = -1(3)	2 - 2 = 0

- 多边形近似
  - ✓ 基本思想：用最少的多边形线段，获取边界形状的本质
  - ✓ 寻找最小基本多边形的方法一般有两种：

    1）点合成法

    2）边分裂法

- 多边形近似



✓ 点合成算法：

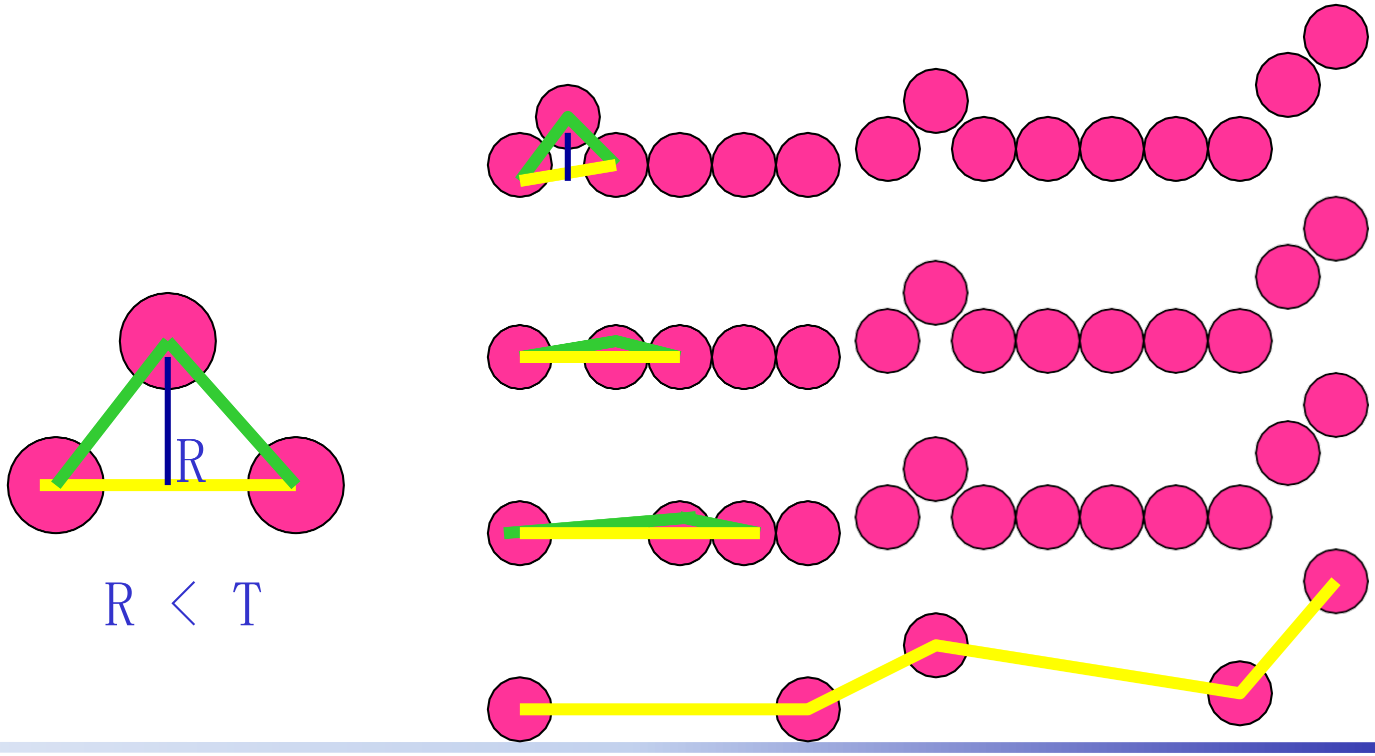
1）沿着边界选两个相邻的点对，计算首尾连接直线段与原始折线段的误差R。

2）如果误差R小于预先设置的阈值T。去掉中间点，选新点对与下一相邻点对，重复1）；否则，存储线段的参数，置误差为0，选被存储线段的终点为起点，重复1）2）。

3）当程序的第一个起点被遇到，算法结束。

表示与描述

- 多边形近似
  - ✓ 点合成算法思想举例：



- 多边形近似
  - ✓ 点合成算法的问题：
    - ➢ 顶点一般不对应于边界的拐点（如拐角）。因为新的线段直到超过误差的阈值才开始画

      例如：如果沿着一条长的直线追踪，而它出现了一个拐角，在超过阈值之前，拐角上的一些点会被丢弃
    - ➢ 下面讲到的分裂法可用于缓解这个问题

- 多边形近似
  - ✓ 分裂边算法：

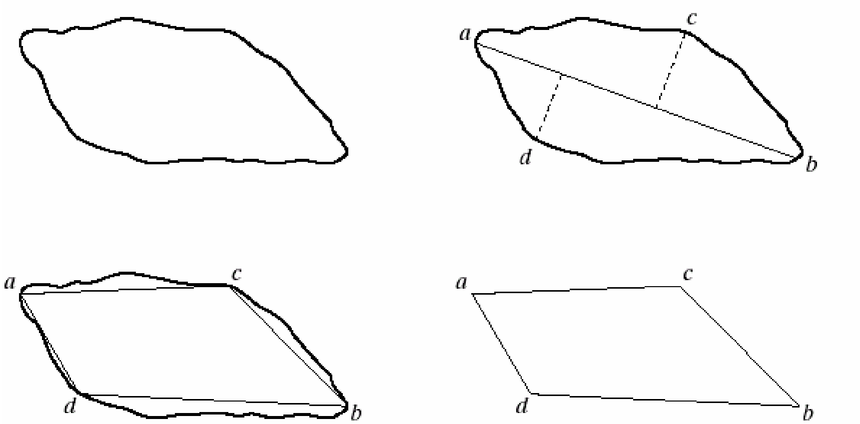
（1）连接边界线段的两个端点（如果是封闭边界，连接最远点）；

（2）如果最大正交距离大于阈值，将边界分为两段，最大值点定位一个顶点。重复（1）；

（3）如果没有超过阈值的正交距离，结束。

表示与描述

- 多边形近似
  - ✓ 边分裂算法思想举例：



a b
c d

FIGURE 11.4
(a) Original boundary. (b) Boundary divided into segments based on extreme points. (c) Joining of vertices. (d) Resulting polygon.

使用直线ab长度的0.25倍作为阈值的拆分过程结果。由于在新的边界线段上没有超过阈值的垂直距离的点，分割过程终止

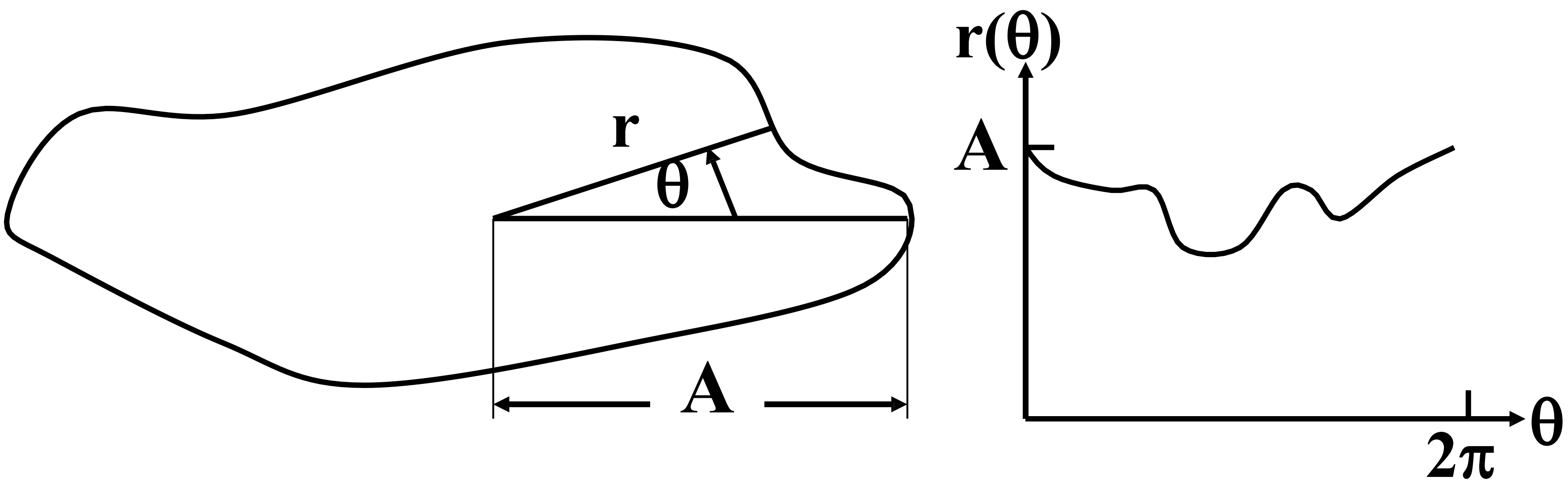
表示与描述

- 外形特征
  - ✓ 基本思想：

    外形特征是一种用一维函数表达边界的方法。基本思想是把边界的表示降到一维函数

表示与描述

- 外形特征
  - ✓ 函数定义——质心角函数：边上的点到质心的距离r，作为夹角θ的函数r(θ)

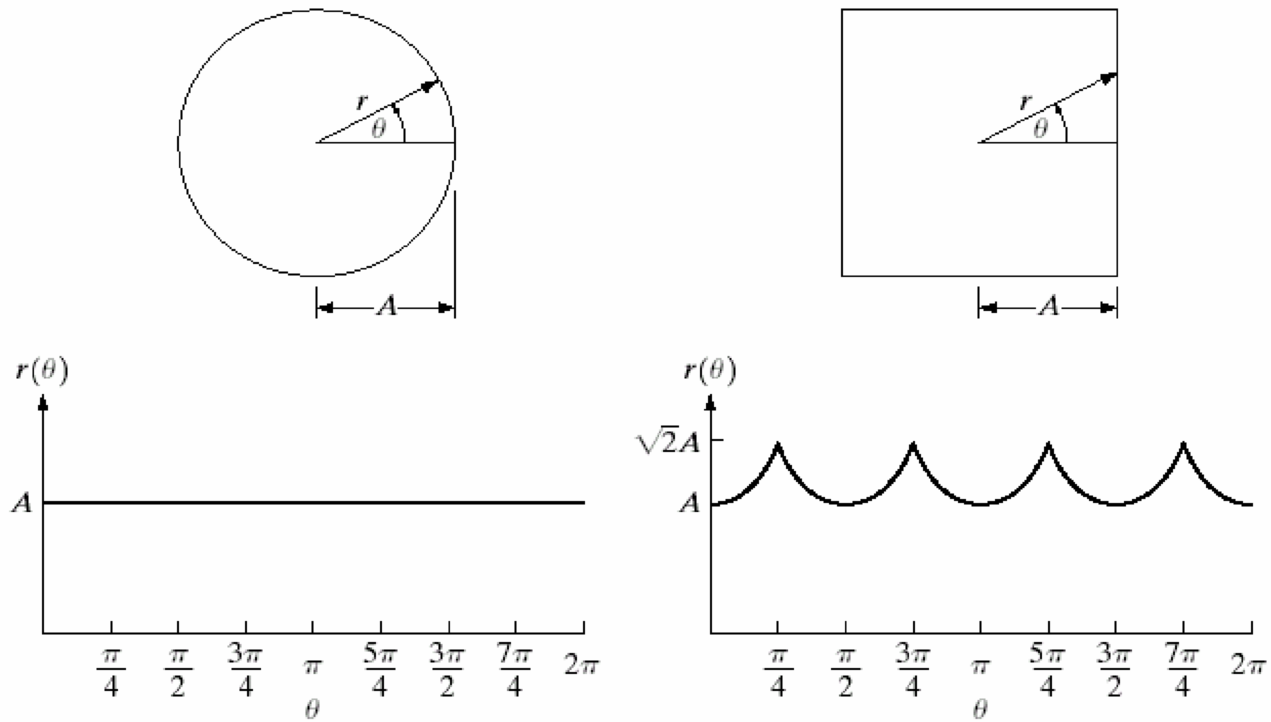


外形特征举例

到达正方形的4个对角上达到最大值

a b

FIGURE 11.5 Distance-versus-angle signatures. In (a) $r(\theta)$ is constant. In (b), the signature consists of repetitions of the pattern $r(\theta) = A\sec\theta$ for $0 \le \theta \le \pi/4$ and $r(\theta) = A\csc\theta$ for $\pi/4 < \theta \le \pi/2$.



表示与描述

- 外形特征
  - ✓ 问题：函数依赖于旋转和比例缩放变换
  - ✓ 改进：
    - ➢对于旋转——两种改进：

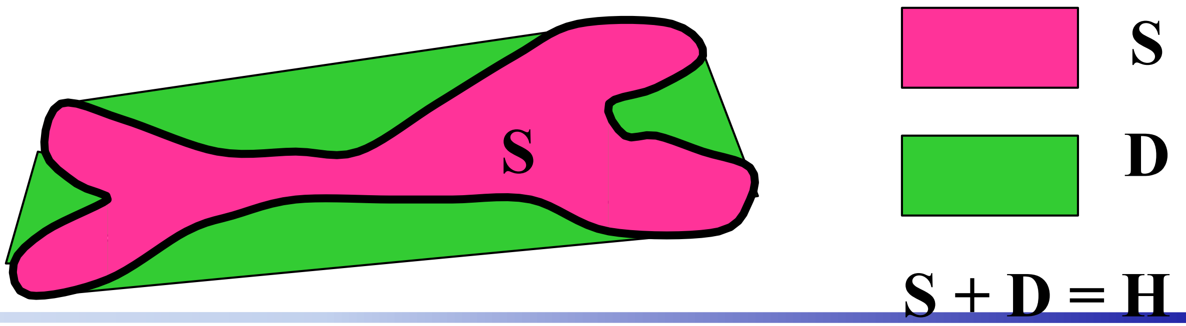
      a. 选择离质心最远的点作为起点 b. 选择从质心到主轴最远的点作为起点

    - ➢对于比例变换：

      对函数进行正则化，使函数值总是分 布在相同的值域里，比如说[0，1]

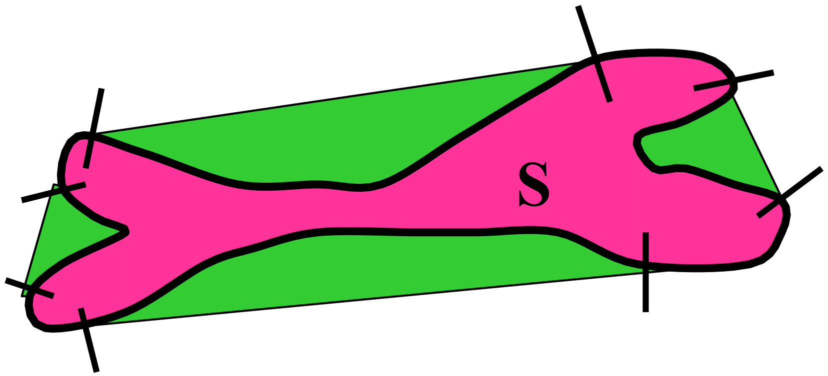
表示与描述

- 边界分段
  - ✓ 基本概念：
    - ➢ 一个任意集合S（区域）的凸起外缘H是：包含S的最小凸起的集合
    - ➢ H-S的差的集合被称为集合S的凸起补集D



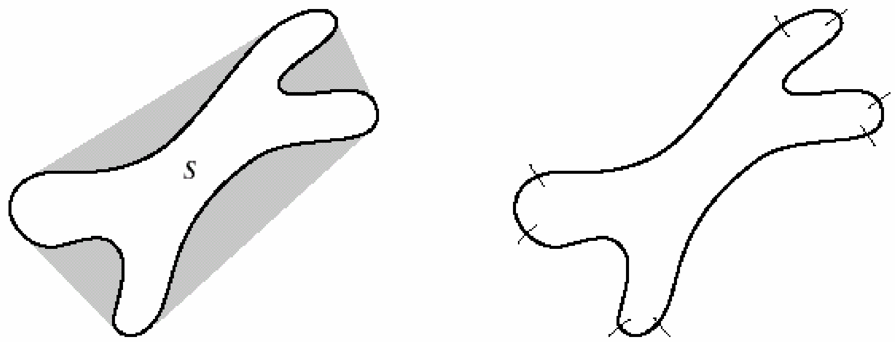
表示与描述

- 边界分段
  - ✓ 分段算法：
    - ➢ 给进入和离开凸起补集D的变换点打标记来划分边界段。
    - ➢ 优点：不依赖于方向和比例的变化



表示与描述

- 边界分段举例



a b

FIGURE 11.6 (a) A region, *S*, and its convex deficiency (shaded). (b) Partitioned boundary.

区域S和它的凸起补集D　　　　被分割的边界

- 边界分段
  - ✓ 问题： 噪音的影响，导致出现零碎的划分。
  - ✓ 解决的方法：

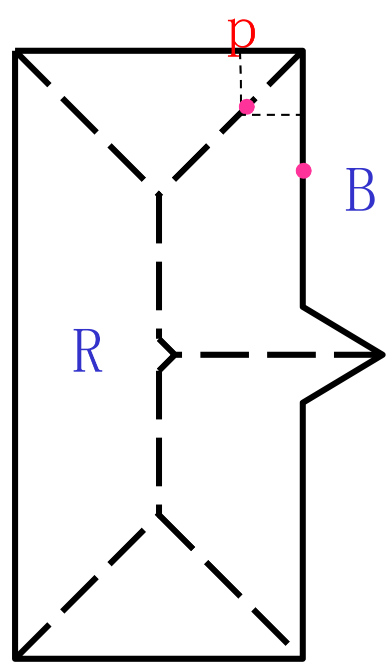
    先平滑边界，或用多边形逼近边界，然后再分段

- 区域骨架
  - ✓ 基本思想
    - 表示一个平面区域结构形状的重要方法是把它削减成图形。这种削减可以通过细化（也称为抽骨架）算法，获取区域的骨架来实现
    - Blum的中轴变换方法（MAT）

      设:R是一个区域，B为R的边界点，对于R中的点p，找p在B上“最近”的邻居。如果p有多于一个的邻居，称它属于R的中轴（骨架）

- 区域骨架（续）
  - ✓ 问题：计算量大

    包括计算区域的每个内部点到其边界点的距离



表示与描述

- 区域骨架
  - ✓ 算法改进思想 在保证产生正确骨架的同时，改进算法的效率。比较典型的是一类细化算法，它们不断删除区域边界点，但保证删除满足：

    （1）不删除端点

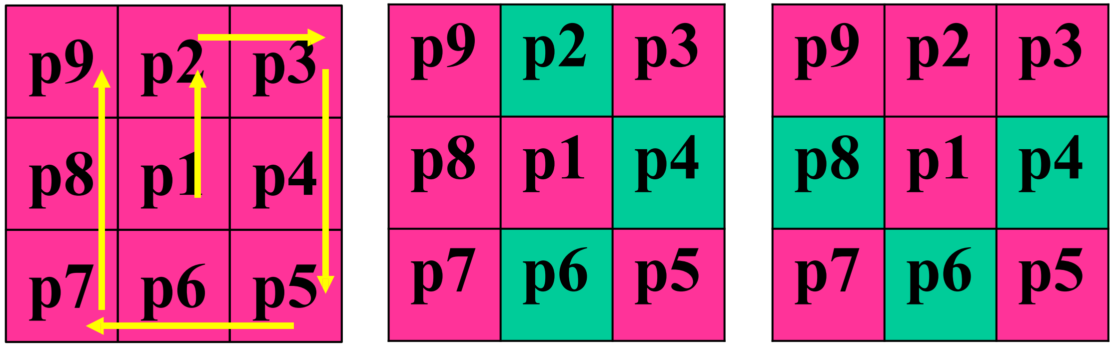
    （2）不破坏连通性

    （3）不造成对区域的过度腐蚀

表示与描述

- 区域骨架
  - ✓一种细化二值区域的算法
    - 假设区域内的点值为1，背景值为0
    - 这个方法由对给定区域的边界点连续进行两个基本操作构成
    - 这里边界点是指任何值为1且至少有一个8邻域上的点为0的像素

- 区域骨架



✓ 基本操作1

对于满足以下四个条件的边界点打标记准备删除：

(a) $2 \le N(p_1) \le 6$ 其中$N(p_1)$是点p_1的邻域中1的个数，

即： $N(p_1)=p_2+p_3+\ldots+p_9$

(b) $S(p_1) = 1$

其中$S(p_1)$是按$p_2, p_3, \ldots, p_9$顺序，0-1转换的个数

(c) $p_2 \cdot p_4 \cdot p_6 = 0$ （p_2, p_4, p_6 至少有一个0）

(d) $p_4 \cdot p_6 \cdot p_8 = 0$ （p_4, p_6, p_8 至少有一个0）

表示与描述

- 区域骨架

  所有条件都满足，才打删除标记。删除并不立即进行，而是等到对所有边界点都打完标记后，再把作了标记的点一起删除

✓ 举例：

$N(p_1) = 4$

$S(p_1) = 3$

$p_2 \cdot p_4 \cdot p_6 = 0$

$p_4 \cdot p_6 \cdot p_8 = 0$ 　第2个条件没满足不打标记

0	0	1
1	p1	0
1	0	1

p9	p2	p3
p8	p1	p4
p7	p6	p5

p9	p2	p3
p8	p1	p4
p7	p6	p5

p9	p2	p3
p8	p1	p4
p7	p6	p5

p9	p2	p3
p8	p1	p4
p7	p6	p5

- **区域骨架**
  - ✓ 基本操作2

    条件(a)、(b)与操作1相同

    条件(c)、(d)改为：

    c’)　　$p_2 \cdot p_4 \cdot p_8 = 0$

    d’)　　$p_2 \cdot p_6 \cdot p_8 = 0$

- 区域骨架
  - ✓ 细化算法 细化算法的一轮操作包括：
    - 按操作1，给边界点打标记——删除点
    - 按操作2，给边界点打标记——删除点
    - 这个基本过程反复进行，直至没有点可以删除为止。此时算法终止。

- 概述
- 表示方法
- 边界描述子
- 关系描述子

表示与描述

- 边界描述子
  - ✓ 简单描述子
  - ✓ 形状数
  - ✓ 傅里叶描述子
  - ✓ 矩量

- 简单描述子
  - ✓ 边界的周长：

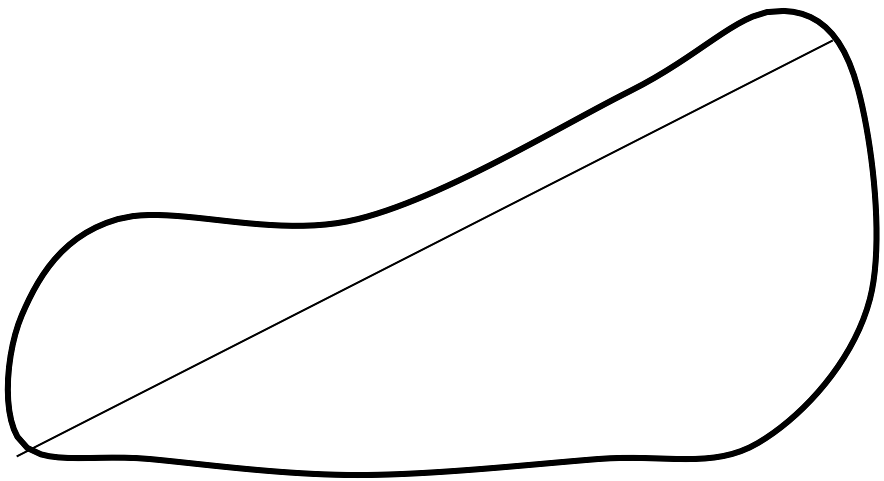
    是最简单的描述符之一。沿轮廓线计算像素的个数，给出了一个长度的近似估计
  - ✓ 边界的直径：边界B的直径是：

$$Diam(B) = \max_{i,j}\left[D\left(p_i, p_j\right)\right]$$

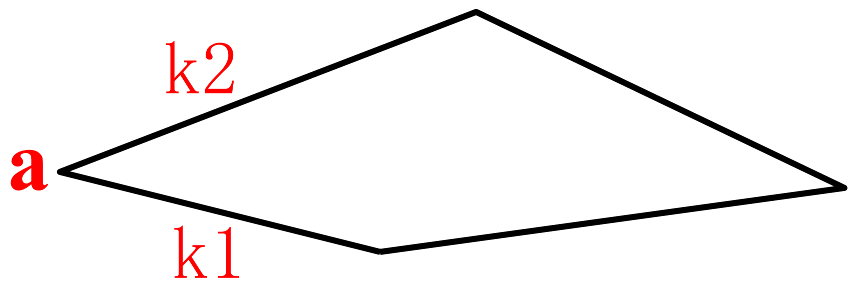
D是欧氏距离或几何距离，p_i，p_j是边界上的点。直径的长度和直径的两个端点连线（这条线被称为边界的主轴）的方向，是关于边界的有用的描述符。

表示与描述

- 简单描述子
  - ✓ 边界的直径举例



表示与描述

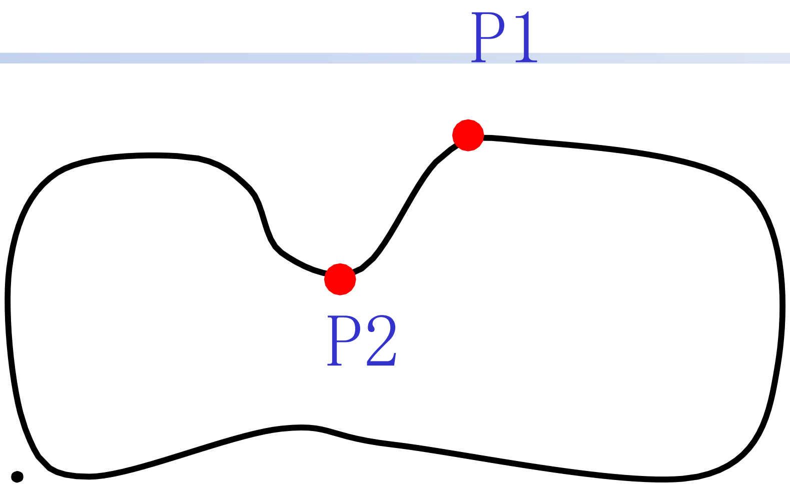


- 简单描述子
  - ✓ 边界的曲率：

    曲率被描述为斜率的变化率。近似：用相邻边界线段（描述为直线）的斜率差作为在边界线交点处的曲率描述子。

交点a处的曲率为 $dk = k1 - k2$

其中k1、k2 为相邻线段的斜率



- 简单描述子
  - ✓ 边界的凸线段点：

    当顶点p上的曲率是为负时，称其为凸线段上的点
  - ✓ 边界的凹线段点：

    当顶点p上的曲率非负时，称其为凹线段上的点

- 形状数——链码的实用化
  - ✓ 形状数定义：最小循环首差链码

循环首差链码：用相邻链码的差代替链码

例如：4链码 10103322 循环首差为：

33133030

循环首差:1 － 2 ＝ −1(3)　　3 － 0 =3

0 － 1 ＝ −1(3)　　3 － 3 ＝ 0

1 － 0 ＝ 1　　2 － 3 ＝ −1(3)

0 － 1 ＝ −1(3)　　2 － 2 ＝ 0

表示与描述

- 形状数
  - ✓ 形状数定义：最小循环首差链码

    例如：　4-链码　：10103322

    　　　循环首差　：33133|030

    　　　形状数　：03033133

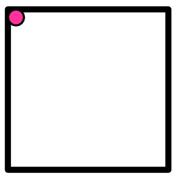
  - ✓ 形状数序号n的定义：　形状数表达形式中的位数。上例序数为8

    对于封闭边界序号一定是偶数。如4、6、8。

表示与描述

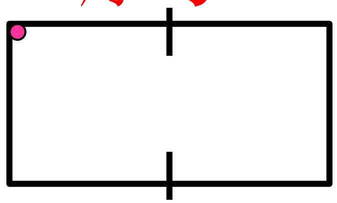
- 形状数
  - ✓ 序号为4、6、8的形状数举例：

序号4



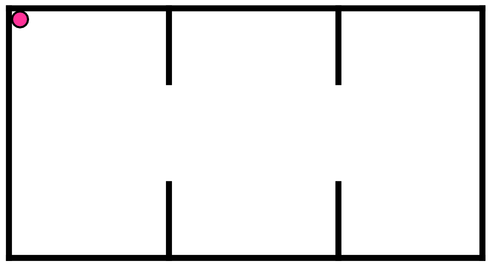
链码：0321
首差：3333
形状：3333

序号6



链码：003221
首差：303303
形状：033033

序号8

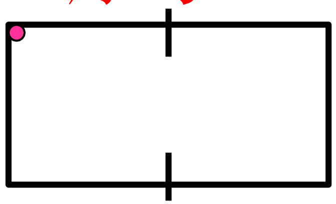


链码：00032221
首差：30033003
形状：00330033

表示与描述

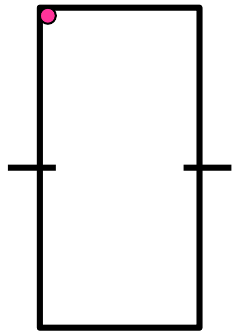
- 形状数
  - ✓ 序号为6的形状数举例：

序号6



链码：003221
首差：303303
形状：033033

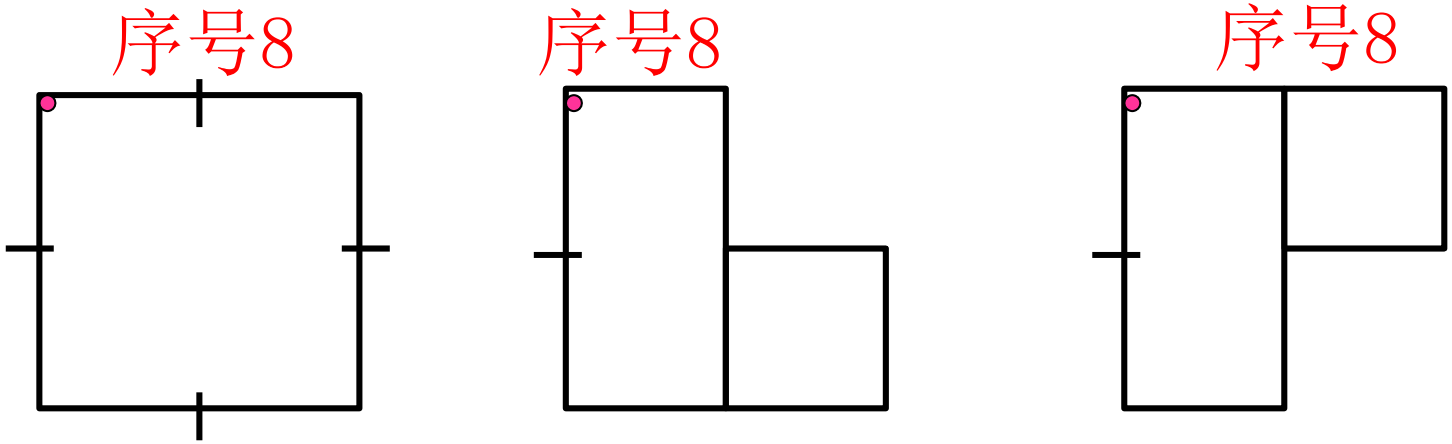
序号6



链码：033211
首差：330330
形状：033033

形状数与方向无关

- 形状数
  - ✓序号为8的形状数举例：

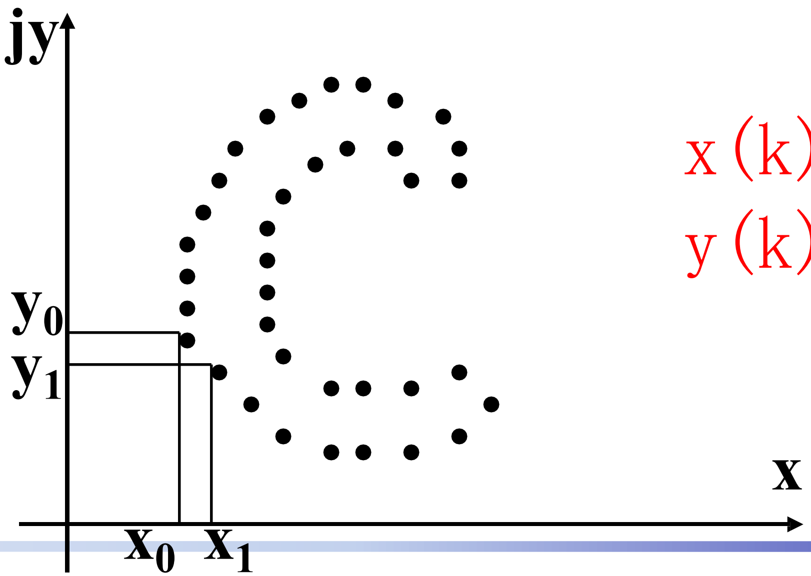


链码：00332211	链码：03032211	链码：00323211
首差：30303030	首差：33133030	首差：30331330
形状：03030303	形状：03033133	形状：03033133

- 傅里叶描述子：将一个二维问题简化成一个一维问题

  ✓ 1）基本思想：

  （1）对于XY平面上的每个边界点，将其坐标用复数表示为： $s(k) = x(k) + jy(k)$， $k=0, 1, \ldots, N-1$



$x(k) = x_k$

$y(k) = y_k$

- 傅里叶描述子
  - 1）基本思想：

（2）进行离散傅里叶变换

$$a(u) = 1/N \sum_{u=0}^{N-1} s(k)\exp(-j2\pi uk/N) \qquad u=0, 1, \ldots, N-1$$

$$s(k) = \sum_{u=0}^{N-1} a(u)\exp(j2\pi uk/N) \qquad k=0, 1, \ldots, N-1$$

系数a(u)被称为边界的傅里叶描述子

- 傅里叶描述子
  - ✓ 1）基本思想：

（3）选取整数 P≤N-1，进行逆傅里叶变换（重构）

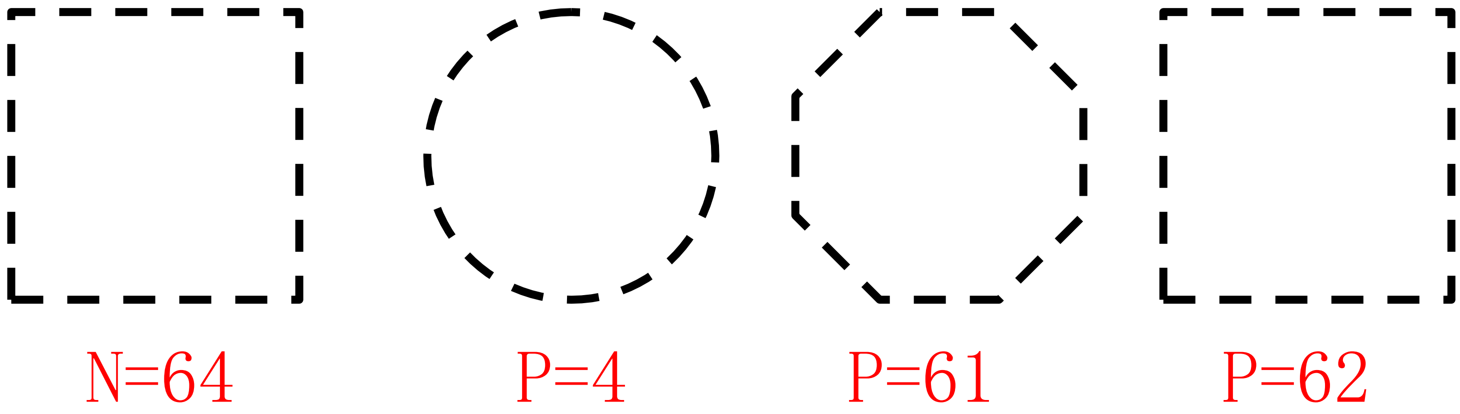
$$s'(k) = \sum_{u=0}^{P-1} a(u)\exp(j2\pi uk/N) \qquad k=0, 1, \ldots, N-1$$

这时，对应于边界的点数没有改变，但在重构每一个点所需要的计算项大大减少了。如果边界点数很大，P一般选为2的指数次方的整数。

- 傅里叶描述符

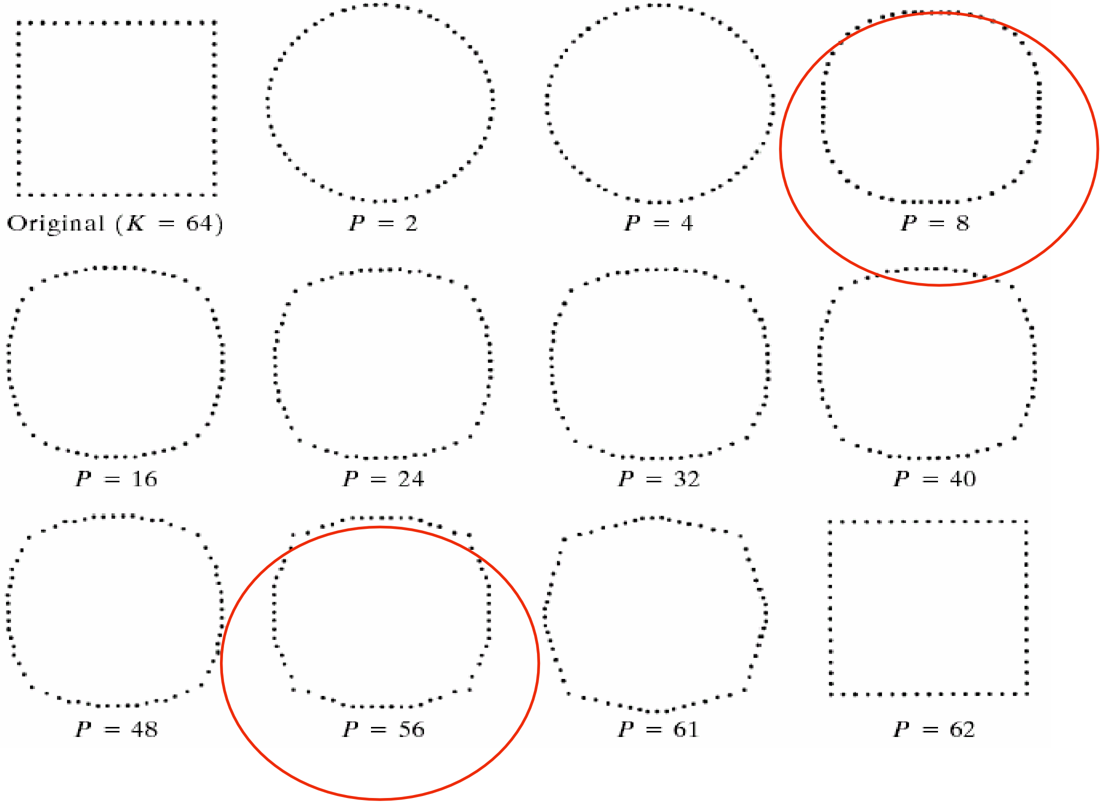
✓ 2）P的选取与描述符的关系

在上述方法中，相当于对于u ＞ P-1的部分舍去不予计算。由于傅里叶变换中高频部分对应于图像的细节描述，因此P取得越小，细节部分丢失得越多。



原图的方形边界

FIGURE 11.14 Examples of reconstruction from Fourier descriptors. P is the number of Fourier coefficients used in the reconstruction of the boundary.



P=56，拐角点开始突出 结论：低阶系数能够反映大体形状，高阶系数可以精确定义形状特征，少数傅里叶描述子携带了形状信息，能够反映边界的大略本质。

表示与描述

- 傅里叶描述符
  - ✓ 3）使用价值

（1）较少的傅里叶描述子（如4个），就可以获取边界本质的整体轮廓

（2）这些带有边界信息的描述子，可以用来区分明显不同的边界

- 傅里叶描述符

✓ 4）优点

（1）使用复数作为描述符，对于旋转、平移、缩放等操作和起始点的选取不十分敏感。

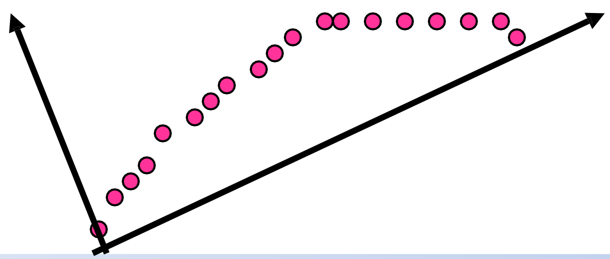
（2）几何变换的描述子可通过对函数作简单变换来获得

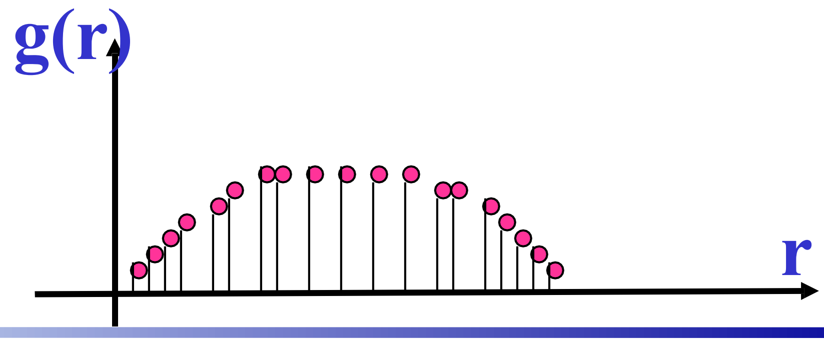
几何变换	傅里叶描述子
原形	$a(u)$
旋转	$a_r(u) = a(u)\ e^{j\theta}$
平移	$a_t(u) = a(u) + \Delta_{xy}\delta(u)$
缩放	$a_s(u) = \alpha a(u)$
起点	$a_p(u) = a(u)\ e^{-j2\pi k_0 u/N}$

- 矩量
  - ✓基本思想：

    将描述形状的任务减少至描述一个一维函数，边界段和特征的形状可以用矩量来 量化地描述

  - ✓矩量的定义：
    - 把边界当作直方图函数：g(r)





- 矩量
  - 矩量的定义：

$$\mu_n(r) = \sum_{i=1}^{L} (r_i - m)^n g(r_i)$$

其中 $$m = \sum_{i=1}^{L} r_i g(r_i)$$

这里L是边界上点的数目，$\mu_n(r)$是边界的矩量

- 矩量
  - ✓ 矩量的优点：
    - ➢ 实现是直接的
    - ➢ 附带了一种关于边界形状的“物理”解释
    - ➢ 对于旋转的不敏感性
    - ➢ 为了使大小比例不敏感，可以通过伸缩r的范围来将大小正则化。

表示与描述

- 概述
- 表示方法
- 边界描述子
- 关系描述子

表示与描述

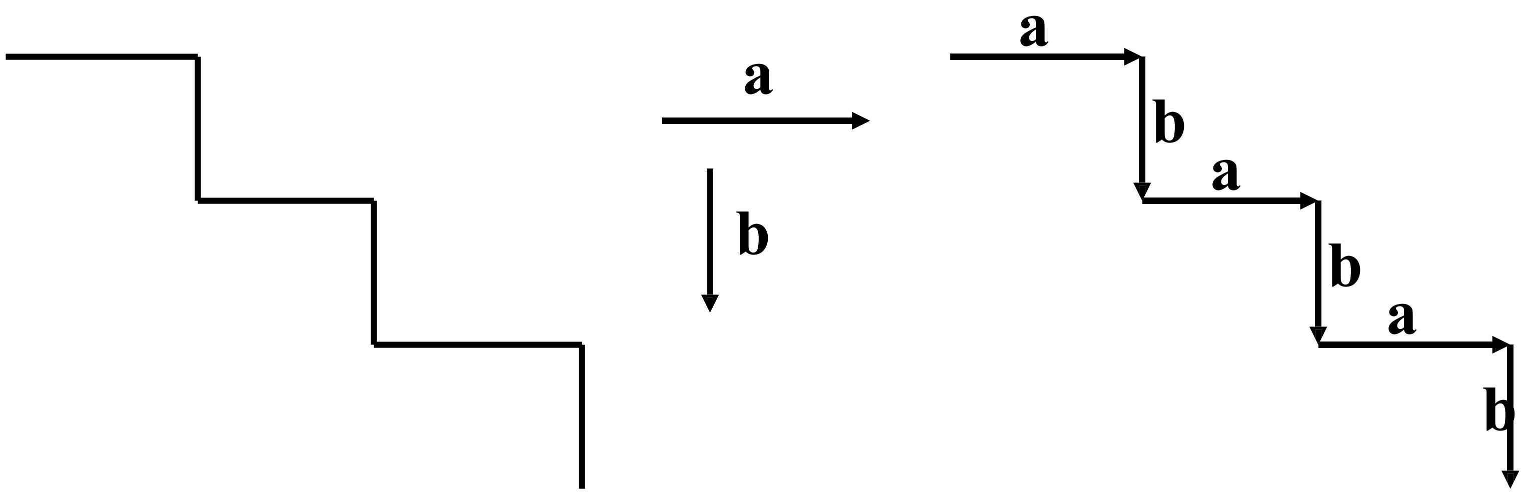
- 关系描述子
  - ✓ 基本思想
  - ✓ 阶梯关系编码
  - ✓ 骨架关系编码
  - ✓ 方向关系编码
  - ✓ 内角关系编码
  - ✓ 树结构关系编码

表示与描述

- 基本思想：
  - ✓ 通过挖掘各个成分之间的结构关系来描述边界
  - ✓ 图像中各个部分间的结构关系是二维的，而串是一维的，期望找到一种方法 把二维关系转化为一维的串
  - ✓ 主导思想是考虑物体各个部分的连接线段

表示与描述

- 阶梯关系编码
  - ✓ 对于如下阶梯形边界，定义两个基本元素a, b



表示与描述

● 阶梯结构关系

✓ 定义如下产生规则：

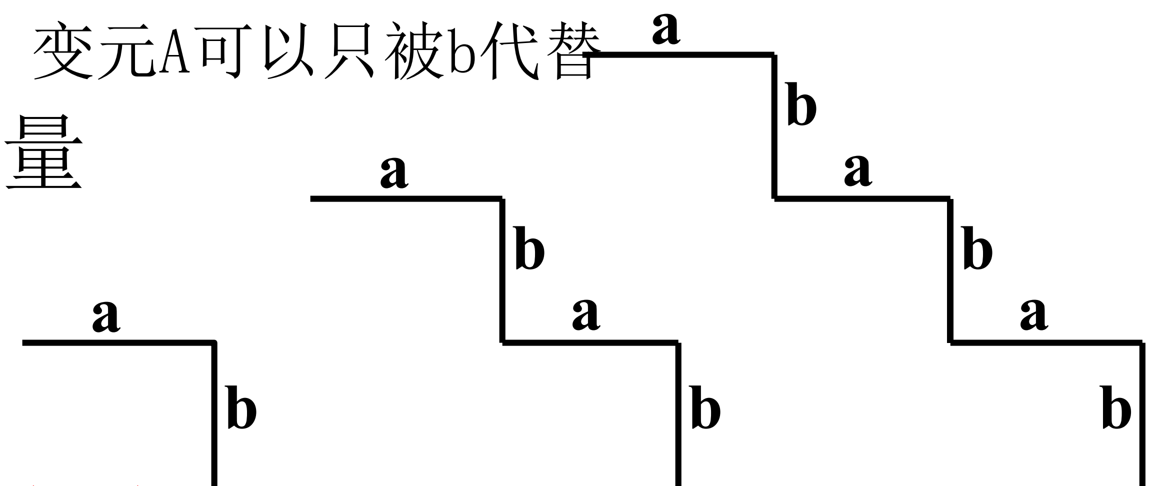
(1) $S \longrightarrow aA$ 表明起始符S可以被图元a和变元A代替

(2) $A \longrightarrow bS$ 表明变元A可以被b和S代替

(3) $A \longrightarrow b$ 变元A可以只被b代替

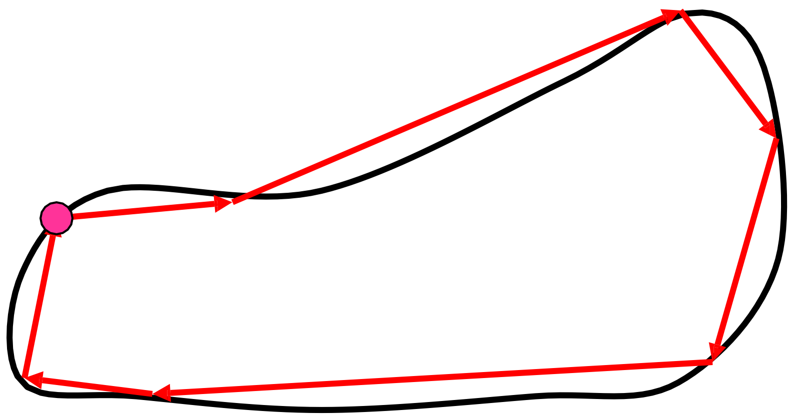
其中S、A是变量

举例：

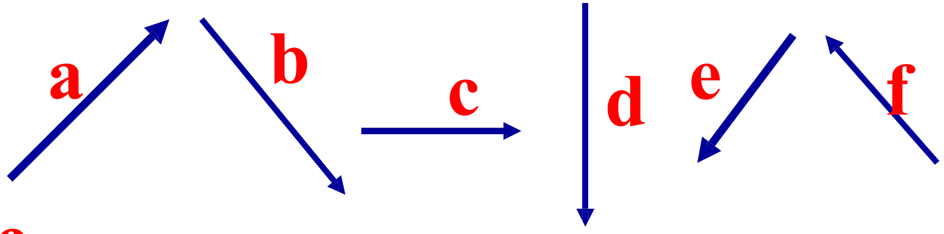


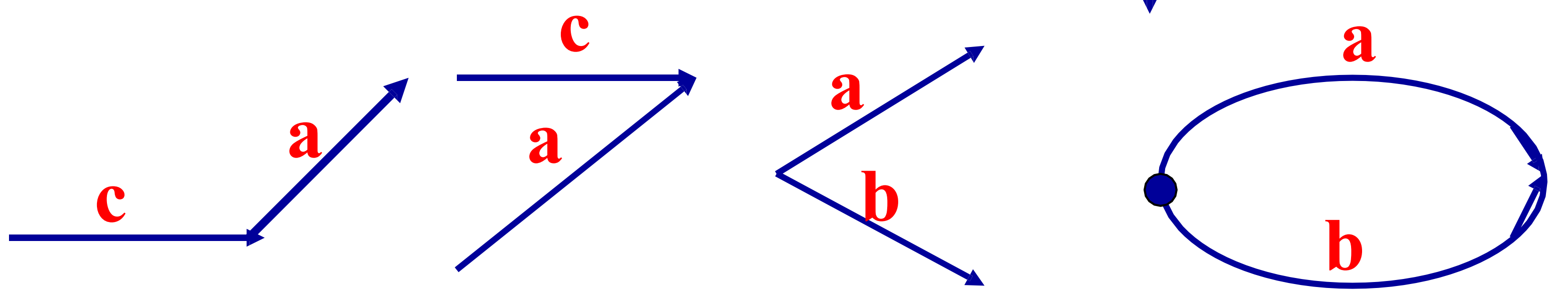
表示规则序号 ← (1, 3)　　(1, 2, 1, 3)　　(1, 2, 1, 2, 1, 3)

- 骨架关系编码
  - ✓ 用有向线段来描述一个图像的各个部分（例如同构区域），这个线段是通过头尾连接等方法得到的。线段之间的不同运算代表了区域的不同组合。
  - ✓ 当图像的连通性可以通过首尾相接或其它连续的方式描述的时候，最适于使用这种串来描述。

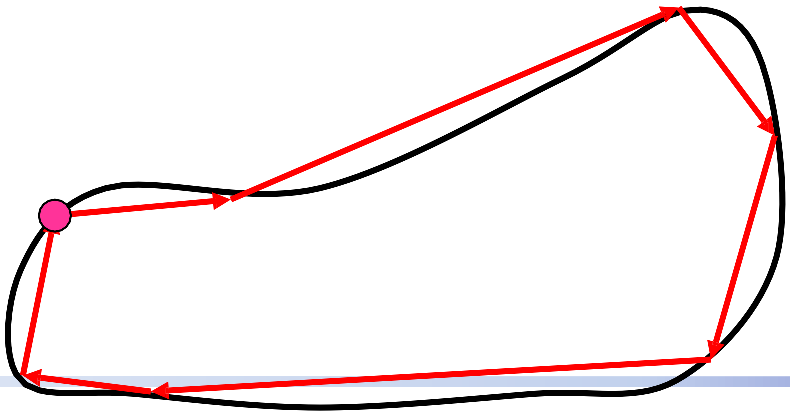


- 骨架关系编码





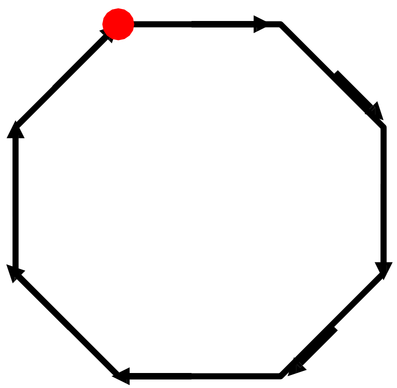
编码 c + a c - a a × b a * b



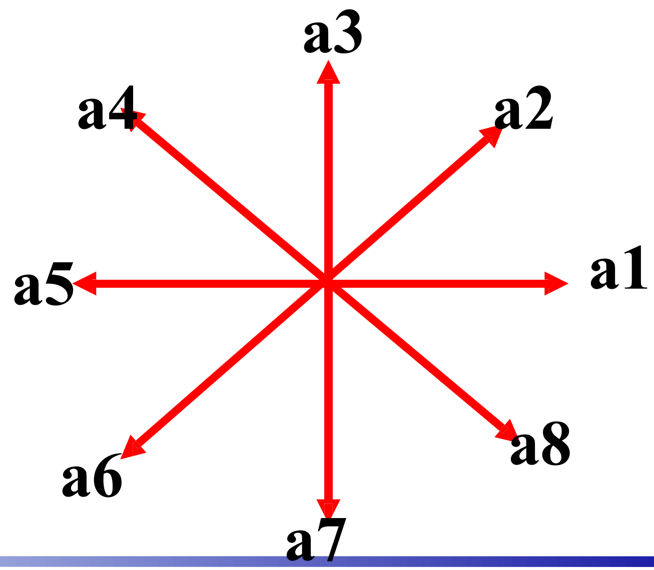
a + a + b + e + e + e + a

表示与描述

- 方向关系编码
  - ✓ 跟踪对象的边界，将跟踪得到的线段按照方向或长度来编码



a1a8a7a6a5a4a3a2



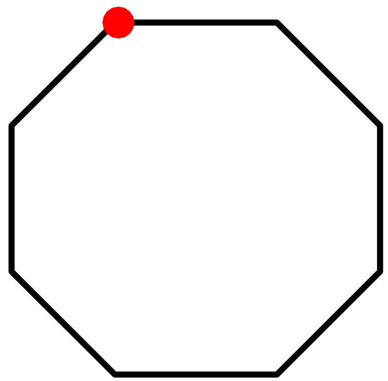
表示与描述

- 内角关系编码
  - 根据内角角度范围不同，编码为8个符号

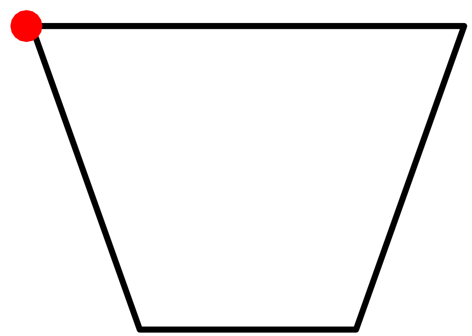
    即： a1:0-45; a2:45-90;a3:90-135;...;

    a8:315-360

举例：



a3a3a3a3a3a3a3a3

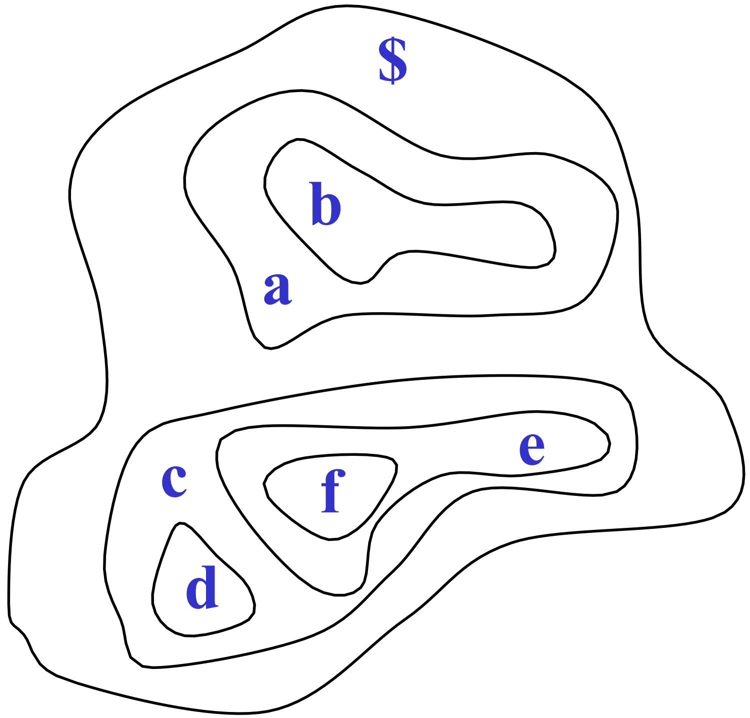


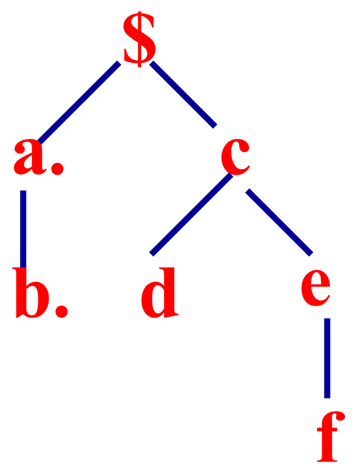
a2a2a3a3

表示与描述

- 树结构关系
  - ✓ 树结构中每个结点的意义和结点之间的关系最为重要

举例：





Any Question?