第七章 轮廓表示

把边缘连接起来就成为轮廓(contour)．轮廓可以是断开的，也可以是封闭的．封闭轮廓对应于区域的边界，而区域内的像素可以通过填充算法来填满．断开的轮廓可能是区域边界的一部分，也可能是图像线条特征，如手写体笔画、图画中的线条等．区域之间的对比度太弱或边缘检测阈值设置太高都有可能产生间断的轮廓．

轮廓可以用边缘序列表或曲线来表示．曲线通常称为轮廓的数学模型．曲线表示包括线段、二次曲线、三次样条曲线等．下面是几种轮廓表示的评价标准：

高效：轮廓应该是一种简单和紧凑的表示．

精确：轮廓应能精确地逼近图像特征．

有效：轮廓应适合于后处理阶段的计算．

轮廓表示的精确性由以下三个方面因素决定：① 用于轮廓建模的曲线形式；② 曲线拟合算法的性能；③ 边缘位置估计的精确度．轮廓的最简单表示形式是边缘有序表．这种表示的精确度就是边缘估计的精确度，但其表示的紧凑性是最差的，因此不是一种有效的后续图像分析表示方法．用适当的曲线模型来拟合边缘会提高精确度，这是因为曲线模型拟合边缘时往往具有均值化效应，因而可以减少边缘位置误差．曲线模型也会提高轮廓表示的经济性，为后处理提供了一种更适合、更紧凑的表示，例如，一条直线上的边缘集用一直线来拟合是表示这些边缘的最简单和最有效的方法，这一表示也简化了后续处理（如确定线的长度和方向）；另外，由于估计直线与真实直线的均值方差小于真实直线与任何其它边缘之间的均值方差，因此可以说这种表示也增加了精确度．

轮廓曲线拟合通常采用内插曲线或逼近曲线来实现．已知一组称为控制点的坐标点，内插是指一条曲线拟合这组控制点，使得曲线通过所有的控制点；逼近是指一条曲线拟合这组这组控制点，使得这条曲线非常接近这些控制点而无需一定通过这些点．在下面几节中，假定由边缘检测器得到的边缘十分准确，并使用内插值方法进行边缘曲线拟合．

定义 7．1 边缘表是边缘点或边缘段的有序集合．

定义 7．2 轮廓是边缘表或用于表示边缘表的曲线．

定义 7．3 边界是包围一个区域的封闭轮廓．

在无特别说明的情况下，边缘通常是指边缘点．对大多数曲线拟合算法来说，只需

要边缘的位置信息。在很少的几种情况下，即需要边缘位置信息，也需要方向角信息，此时的边缘是指边缘段.

平面曲线函数可以表示为三种形式：显式 $y=f(x)$，隐式 $f(x,y)$，或参数式 $(x(u),y(u))$，其中 u 是某一参数．函数的显式表示很少用在机器视觉中，主要原因是 $x-y$ 平面上的曲线可能卷曲，使得一个 x 值可能对应曲线上多个 y 值.

7．1 数字曲线及其表示

本节将讨论一组计算曲线几何元素的算法，几何元素包括轮廓长度、正切方向角和曲率等．由于邻接像素之间的量化增量是 45°，因此，精确计算斜率和曲率是很困难的.

估计正切方向角的基本思路是使用边缘表中非邻接的边缘点，这就允许存在一个较大的可能正切方向角集合．设 $\mathbf{p}_i=(x_i,y_i)$ 是边缘表中第 i 个边缘坐标．k 斜率是在边缘表相距 k 个边缘点的两个边缘点之间的方向向量．左 k 斜率是 $\mathbf{p}_{i-k}$ 指向 $\mathbf{p}_i$ 的方向，右 k 斜率是 $\mathbf{p}_i$ 指向 $\mathbf{p}_{i+k}$ 方向．k 曲率是左右 k 斜率之差值.

假定在边缘表中有 n 个边缘 $(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)$．数字曲线的长度可以近似为像素之间的线段和：

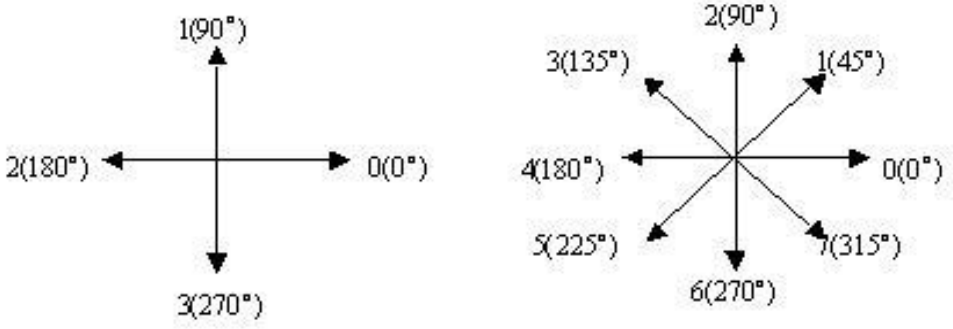
$$S=\sum_{i=2}^{n}\sqrt{(x_i-x_{i-1})^2+(y_i-y_{i-1})^2} \tag{7．1}$$

轮廓端点之间的距离为：

$$D=\sqrt{(x_n-x_1)^2+(y_n-y_1)^2} \tag{7．2}$$

7．1．1 链码

链码是沿着轮廓记录边缘表的一种表示方法．链码规定了边缘表中每一个边缘点所对应的轮廓方向，其中的轮廓方向被量化为 4 －邻接链码或 8 －邻接链码中的一个，如图 7．1 所示．图 7．2 所示的是一条曲线及其 8 －邻接链码的表示，8－邻接链码从边缘表中第一个边缘开始，沿着轮廓按逆时针方向行走，行走方向用八链码中的一个表示.



（a） （b）

图 7．1　连接边缘点方向的链码示意图，（a） 4 －邻接链码，(b) 8－邻接链码

链码有一些很特殊的的性质．一个物体很容易实现 45° 角旋转．如果一个物体旋转 $n \times 45^\circ$，旋转后的物体链码可由原链码加上 n 倍的模 8 得到．链码的微分，也称差分码，可由原码的一阶差分求得．链码差分是关于旋转不变的边界描述方法．比如，

图 7． 2 曲线的链码是：60222220210134444444454577012

其差分链码是： 22000062771210000017120111

图 7． 3 是图 7． 2 曲线逆时针旋转 90° 后得到的，

曲线的链码是：02444442432356666666711234

其差分链码是： 22000062771210000017130111

由此可见，一条曲线旋转到不同的位置将对应不同的的链码，但其差分链码不变，即差分链码关于曲线旋转是不变的．

区域的一些其它性质，如面积和角点，也可以由链码直接求得．这种表示的局限性是表示某一点正切方向的集合是有限的（ 4 －邻接链码有 4 个，8－邻接链码有 8 个），这一局限性可以通过下面几节介绍的曲线表示方法来克服．

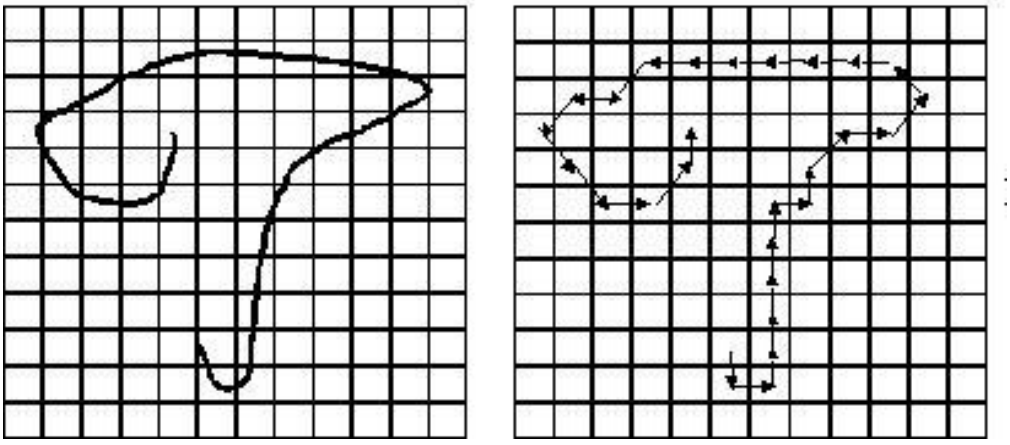


图 7．2 一条曲线及其 8 －邻接链码表示

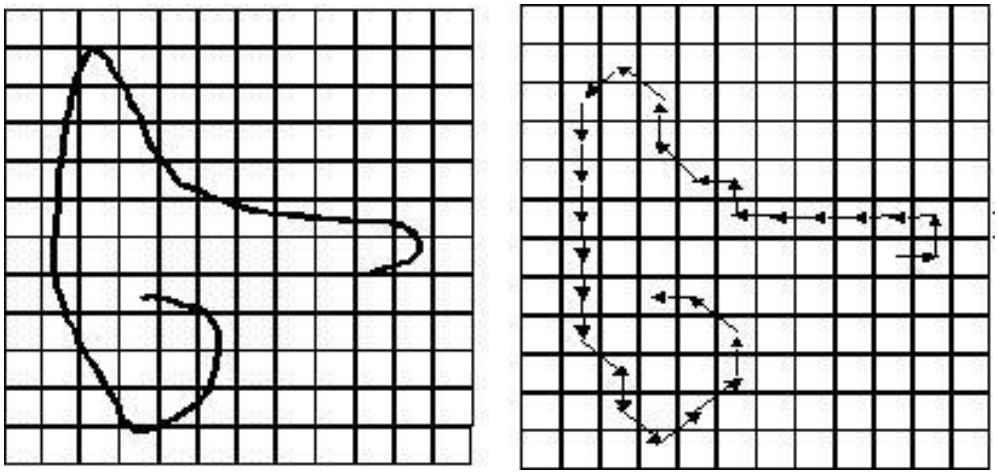


图 7． 3 一条曲线及其 8 －邻接链码表示

7．1．2 斜率表示法

用任意的正切方向来表示轮廓可以克服链码的只能用有限个正切方向来表示轮廓的局限性．假定从边缘表开始，使用上面给出的公式计算正切和弧长，可以画出正切 Ψ 同弧长 s 的关系图，称作 $\Psi - s$ 图．$\Psi - s$ 图是轮廓形状在 $\Psi - s$ 空间的表示，是一种轮廓形状的紧凑描述．图 7． 4 所示的是包含有直线段和圆弧段的轮廓在 $\Psi - s$ 空间中的

表示，它是一个直线段序列．对于封闭轮廓，$\Psi - s$ 图是一个周期曲线．在 $\Psi - s$ 图中，水平方向的直线段对应轮廓中的直线段，这是由于直线段对应的斜率是恒定值。其它方向的直线段对应圆弧段，非直线段部分对应曲线基元．

如果把 $\Psi - s$ 图分割成直线段，也就把轮廓分割成直线段和圆弧段．许多研究人员采用这一方法来分割轮廓，由此产生了轮廓分段方法的多种形式．

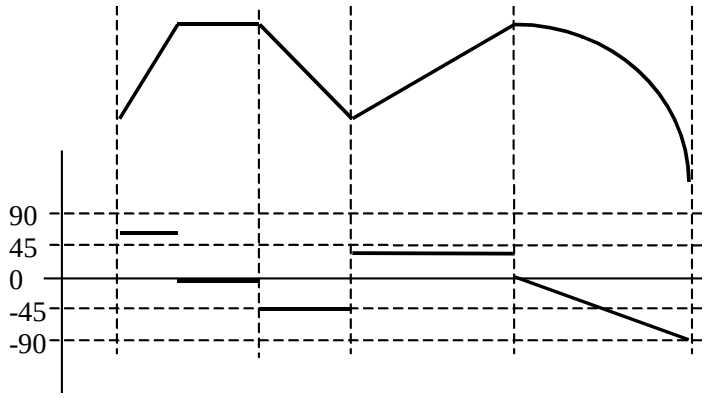


图 7．4　轮廓的斜率表示

7．2 曲线拟合

本章将讨论三种常用的曲线模型拟合边缘点的方法：直线段(Line Segment)，圆锥曲线段(Conic Section)和三次样条曲线段(Cubic Spline)．一般来说，在用曲线模型拟合边缘点之前应考虑如下两个问题：

用什么方法进行边缘点的曲线模型拟合？

如何测量拟合的逼近程度？

下面几节将讨论曲线模型拟合边缘点方法，其中假设边缘位置足够精确，不会对拟合结果产生影响．

设 d_i 是边缘点到一条拟合曲线的距离，该距离值有正负符号，在曲线同一侧的边缘具有相同的正符号或负符号．目前有许多种测量曲线与候选边缘点的拟合效果方法，每一种都取决于拟合曲线和候选点之间的误差．下面是一些常用的方法．

(1)　最大绝对误差

测量最坏情况下边缘点偏离曲线的距离：

$$MAE = \max_i | d_i | \qquad (7．3)$$

(2) 均方差

给出边缘点偏离拟合曲线的总的测度：

$$MSE = \frac{1}{n}\sum_{i=1}^{n} d_i^2 \qquad (7.4)$$

(3) 最大规范（normal）误差

最大绝对误差与曲线长度 S 之比：

$$\varepsilon = \frac{\max_i |d_i|}{S} \qquad (7.5)$$

(4) 误差符号变化次数

这里的误差就是指 d_i，即边缘点偏离拟合曲线的距离。误差符号变化次数可用来表示轮廓边缘模型的曲线适合程度的测度.

(5) 曲线长度与端点距离之比

曲线复杂程度的测度.

符号变化是一种评价拟合好坏的很有用的参数. 比如，用直线段逼近边缘表，并检测符号变化数. 如果符号变化一次，则说明边缘点可以由直线段来逼近；符号变化两次，说明边缘可以由二次曲线逼近；符号变化三次，说明边缘模型是三次曲线，依此类推. 如果符号变化数量很大，则意味着曲线复杂度增加一点将不能显著地改善拟合效果. 一种好的拟合所对应的符号变化具有随机模式. 相同符号连续出现多次说明存在拟合系统误差，这种误差可能是由于错误的曲线模型引起的.

曲线拟合模型的选择取决于应用场合. 如果场景是由直线段组成，则使用直线段（或多线段）模型比较合适. 直线段模型也可作为其它拟合模型的初始拟合模型. 圆弧段是估计曲率的最有用的一种表示，因为曲线可以分割成具有分段恒定曲率线段. 圆锥曲线段是一种表示直线段和圆弧段序列以及椭圆和高次弧段序列的有效方法. 三次样条曲线适合于平滑曲线模型，因为三次样条曲线并不要求正切向量和曲率的估计值一定是分段恒定的.

7．2．1 多直线段

多直线段是指端点连结端点的直线段序列，直线段序列的连接点称为顶点. 多直线段适合具有线段序列的边缘列表的拟合. 多线段算法的输入值是边缘点有序表 $\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$. 边缘点坐标可以计算到子像素精度. 由于线段的两个端点对应两个边缘点，即线段拟合在这两个边缘点之间进行，因此仅需要精确计算对应端

点的两个边缘点的坐标.

拟合边缘表并把第一和最后一个边缘点 (x_1, y_1) 和 (x_k, y_k) 连接起来的直线线公式如下：

$$\frac{y - y_1}{x - x_1} = \frac{y_k - y}{x_k - x} \tag{7．6}$$

上式可以改写为由端点表示的隐式函数：

$$Ax + By + C = 0 \tag{7．7}$$

其中，

$$A = (y_k - y_1) \qquad B = (x_k - x_1) \qquad C = x_1 y_k - x_k y_1$$

而 $D = \sqrt{A^2 + B^2}$ 是边缘点 (x_1, y_1) 和 (x_k, y_k) 之间的距离.

任意一点 (x_i, y_i)，设 $r = Ax_i + By_i + C$，则 r 的符号可以用来计算符号变化次数 C．点 (x_i, y_i) 与上述直线段的距离为：

$$d_i = \frac{|Ax_i + By_i + C|}{D} \tag{7．8}$$

最大规范误差为：

$$\varepsilon = \frac{\max_i |d_i|}{D} \tag{7．9}$$

最大规范误差常常作为线段拟合边缘列表好坏的量度．需要指出的是上面的公式都是在点向直线段的垂直投影落在线段内这个假设下进行的．对于其它情况，则应修正公式，以便计算点到最近的线段端点的距离．下面介绍两种拟合多线段的方法：自顶而下的分裂和自底而上的合并.

(1) 多直线段分裂

自顶而下的分裂算法(top-down splitting)是将整条曲线作为初始曲线，通过反复增加顶点数来进行直线段拟合曲线．考虑图 7．5 所示的边缘点曲线（可以认为是由离散边缘点构成），将第一个和最后一个边缘点连成的直线作为曲线的初始拟合，用 AB 标记.在边缘表中计算最大规范误差，如果该误差值高于某一阈值，则在离直线段最远的边缘点上设置一个顶点，用 C 来标记．这样，将形成两个拟合直线段 AC 和 CB，边缘表也分割成对应于两个新直线段的两个子边缘表．在每一个子边缘表中，重复上面所述的分裂算法，形成两个新的直线段及对应的两个更小的子边缘表．这样的分裂过程可以一直进行下去，直到所有的直线段对应的最大规范误差均低于某一阈值为止．多线段分裂也称为迭代分解.

(2) 线段合并

线段合并(merging)是指用一条直线段尽量多地拟合边缘表中的边缘点．当边缘点离直线段太远而无法用该直线段拟合时，则开始新的直线段拟合．合并方法也称为自底而上(bottom-up merging)的多线段拟合方法．

确定边缘点离直线段的距离有许多种方法．一种方法是使用序贯最小二乘法，完成直线段到边缘点的最小二乘法拟合，并在每次处理新的边缘点时递增地更新线段参数．拟合算法将计算直线段模型和边缘点之间的残差平方．当误差超过某一阈值时，引进一个顶点，并将上一个线段的端点作为新的起点开始新的直线段拟合．

误差带算法是另一种确定顶点位置的方法，如图 7．6 所示，主要工作是计算两条离中心线距离为 ε 且平行于拟合边缘点的直线段． ε 值表示离有差拟合直线的绝对偏离值．只要新的边缘在误差带内，就把这些边缘增加到当前线段内．当新的边缘增加到线段内时，线段的参数要重新计算．逼近直线段没有与误差带边保持平行．位于线段端点的顶点是下一线段的起点．这一方法常常产生大量的线段．由于算法须行进到边缘直到误差带表示的边界才产生角点，因此，不能精确估计角点位置和角度．

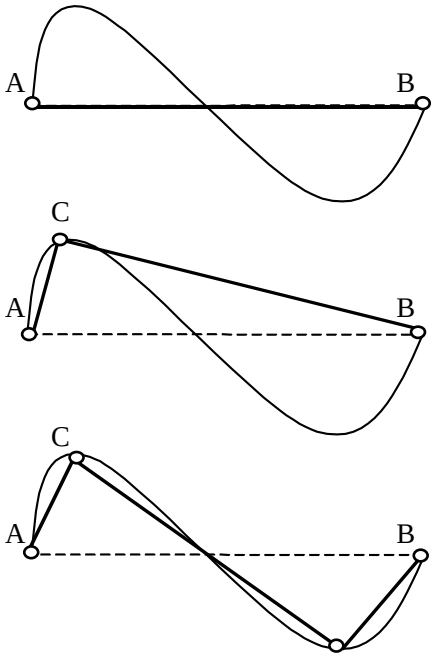


图 7．5 多直线段分裂方法

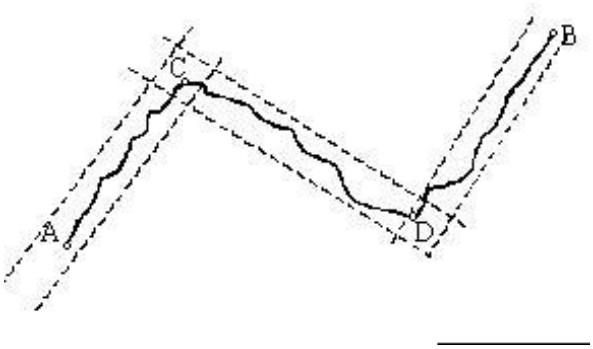


图 7．6　拟合直线段的误差带方法图

(3) 分裂和合并

自顶而下的迭代分解方法和自底而上的合并方法组合起来，形成合并和分裂算法．单独使用分裂或合并算法时，成功率往往不是很高，改进的方法是交叉使用分裂和合并算法．分解过程以后，如果新的线段以很小的规范误差拟合边缘，则允许用单一直线段代替邻接线段．请注意，由于多直线段总是比单直线段的拟合误差小，因此很有必要使用规范误差．在线段合并后，新的线段可能在不同点处分裂．这样，分裂和合并交替作用直到没有线段被合并和分裂为止．图 7．7 所示的是先分裂后合并来修补坏顶点位置的示意图．

一种有效的分裂和合并算法是从边缘表中的前 k 个边缘构成的子列表开始，而不是整个边缘列表．用直线段拟合子表中第一和最后一个边缘点之间的边缘点．如果最大规范误差太大，则将子列表缩到最大误差对应的边缘点处，这样一直进行下去，就可以得到第一条拟合直线段，这实际上是分裂算法．置当前拟合的直线段为旧线段，再在剩下的边缘点集中取前 k 个边缘构成新子列表，用分裂算法求取第二条拟合直线段．比较当前直线段和原直线段的方向，如果它们具有相似的方向，则将这两条直线段合并，这是合并算法．

实际的轮廓曲线并不全是由直线段组成，可能还包含有各种弧线或自由曲线．因此，仅使用直线段拟合比较粗糙，自然人们想到了用弧线段逼近．通常的弧线有二次／三次或更高次的曲线，这些都称为多项式曲线．下面我们主要讨论二次曲线和三次样条曲线．

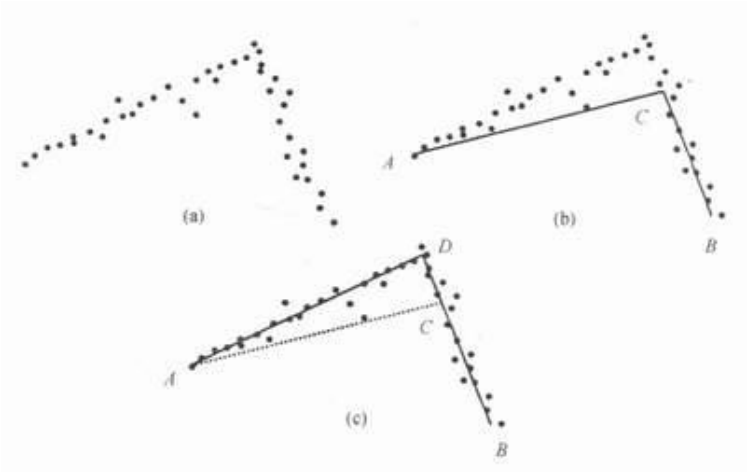


图 7．7　(a)　原始边缘点集．(b)自底而上的边缘合并方法产生的坏角点估计．(c) 漏掉的真实角点位置由分裂和合并过程来修补

7．2．2　圆锥曲线

下面讨论如何用圆锥曲线逼近边缘表．圆锥曲线的一般表示如下：

$$f(x,y)=ax^2+2hxy+by^2+2ex+2gy+c=0 \qquad (7．10)$$

圆锥曲线也称二次曲线，因为二次曲线都是用平面切割正圆锥面的截线，如图 7．8 所示．若平面不通过锥顶，且不平行任一母线，则截线为椭圆，其中圆是椭圆的一种特殊情况，此时的平面垂直于锥轴；若平面不通过锥顶但平行于一条母线时，截线为抛物线；若平面不通过锥顶，且平行于锥轴，截线为双曲线．当平面通过通过锥顶时，椭圆变为一点，双曲线变为一对相交的直线，抛物线变为与圆锥相切的一条直线．由此可见，使用二次曲线来逼近数字轮廓曲线，可以有效地拟合数字曲线中的直线和圆弧等各种二次曲线．由于二次曲线中除了特殊的直线外，最简单的情况时圆弧，因而得到了大量的研究．我们将单独讨论圆弧逼近方法．

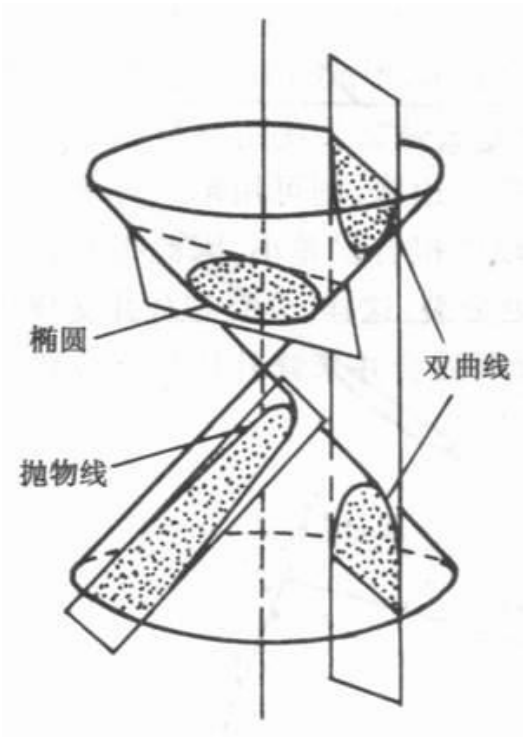


图 7．8　圆锥和平面定义的圆锥曲线

(1) 圆弧段

用直线段逼近边缘表以后，其中的一些直线段序列可以由圆弧段来代替，比如，用直线段拟合一个圆弧是可能需要许多个直线段才能满足拟合误差．如果将这些直线段用用圆弧段来拟合，则仅需要一条圆弧段即可．因此可见，圆弧段拟合是在多边形的顶点上进行的．

具有半径 r 和圆心坐标 (x_0, y_0) 的圆弧隐式方程为：

$$(x-x_0)^2+(y-y_0)^2=r^2 \tag{7．11}$$

考虑三点 $p_1=(x_1,y_1)$，$p_2=(x_2,y_2)$，$p_3=(x_3,y_3)$，不失一般性，设 p_1 位于坐标原点，即 $x_1=0, y_1=0$．把点 p_1，p_2，p_3 代入上述方程：

$$\begin{aligned} x_0^2+y_0^2-r^2&=0\\ (x_2-x_0)^2+(y_2-y_0)^2-r^2&=0\\ (x_3-x_0)^2+(y_3-y_0)^2-r^2&=0 \end{aligned} \tag{7．12}$$

分别用第二和第三方程减去第一方程，得到如下两个方程：

$$\begin{aligned} 2x_2x_0+2y_2y_0&=x_2^2+y_2^2\\ 2x_3x_0+2y_3y_0&=x_3^2+y_3^2 \end{aligned} \tag{7．13}$$

这是两个关于未知参数 x_0, y_0 的非线性方程，求解 x_0, y_0 ,即可由（7．12）的第一个式子得到园的半径.

为了计算圆弧拟合误差，将点 Q 到圆的距离定义为沿着通过园心直线方向点 Q 与圆的距离．计算点 $Q(x_i, y_i)$ 到园中心 (x_0, y_0) 的距离 q :

$$q = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \tag{7．14}$$

点 Q 到圆弧段的距离是

$$d = q - r \tag{7．15}$$

用圆弧段拟合多直线段时，圆弧段的两个端点要经过多直线段的某两个顶点，第三个点位于这两个顶点之间。选择第三个点时可能有如下几种情况：

1．选用距离连接这两个顶点直线最远的多直线段顶点.

2．选用距离连接这两个顶点直线最远的边缘点.

3．选用这两个顶点之间所有顶点的中点.

4．选用这两个顶点之间所有边缘的中点

计算所有边缘点和圆弧段之间的距离误差，计算最大绝对误差和符号变化次数．如果最大规范误差低于某一阈值，而符号变化次数很大，则接受这一圆弧段；否则，保留多线段逼近．关于圆弧段逼近的文献很多，感兴趣的读者可以参见文献[]。

用直线段和圆弧段可以有效地表示数字轮廓曲线，但是用两种不同的线段基元表示轮廓会造成不便．下一节我们将讨论圆锥曲线，圆锥曲线允许直线段、圆弧段、和其它基元共同出现在同一种表示中．圆锥曲线段表示也提供了曲线之间平滑过渡的方法，也是角点的显式表示.

(2) 圆锥曲线

圆锥曲线可以拟合轮廓多直线段上的三个顶点．将圆锥曲线段连接在一起的点称为结点．圆锥样条曲线是圆锥曲线的一个序列，它们的端点和端点连接在一起，在结点处具有相等的正切，以便使两个邻接曲线段之间平滑过渡．设多直线段端点为 V_i ．圆锥逼近如图 7．9 所示.

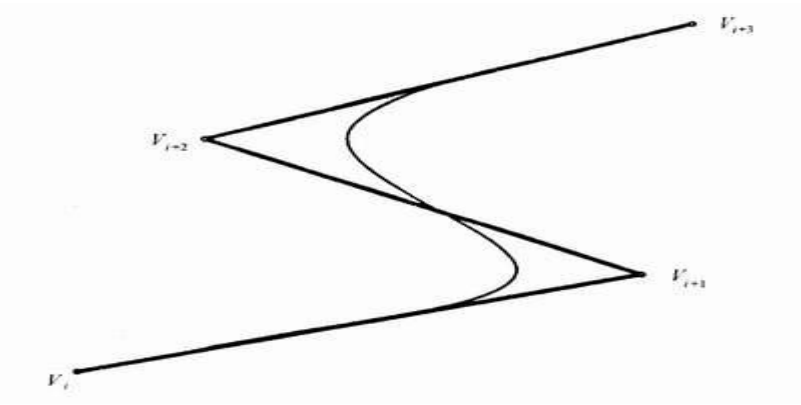


图 7．9　由三点定义的圆锥曲线逼近

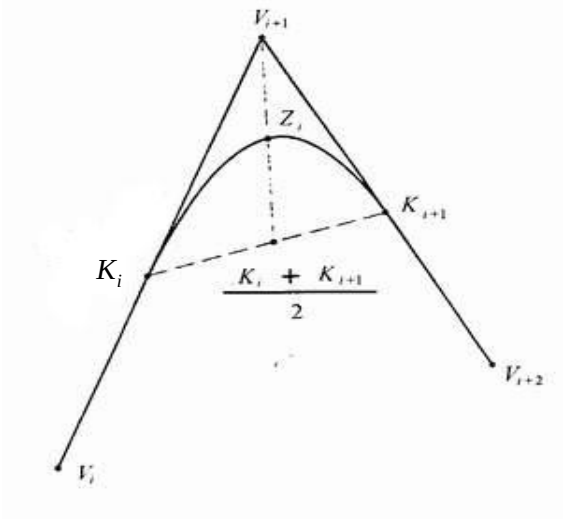


图 7．10　由两个端点、三顶点构成的正切和第三点定义的圆锥曲线

圆锥样条中的每一个圆锥曲线由两个端点、两个正切和第三点确定．结点 K_i 位于多线段顶点之间：

$$K_i = (1-v_i)V_i + v_i V_{i+1} \tag{7．16}$$

其中 v_i 取 0 和 1 之间的数值．正切由三个顶点 V_i, V_{i+1} 和 V_{i+2} 组成的三角形来定义．第三点如图 7．10 所示，由下面式子计算

$$Z_i = \gamma_i V_{i+1} + (1-\gamma_i)\frac{K_i + K_{i+1}}{2} \tag{7．17}$$

有几种特殊的情况可以通过如下方法来处理．如果 $v_{i+1} = 0$，那么第 i 个圆锥样条

曲线是 K_i 到 V_{i+1} 的直线段．如果 $v_i=1$ 和 $v_{i+1}=0$，那么 K_i， K_{i+1} 和 V_{i+1} 对应于同一点，在圆锥曲线序列中有一个角点．这些性质使得直线段和角点可以明显地表示出来，无需求助于不同的基元和特殊的标志．

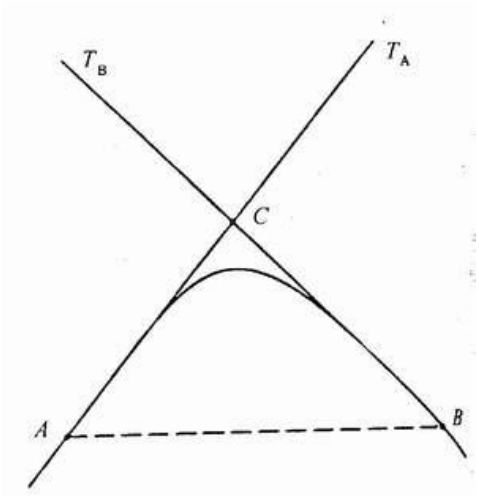


图 7．11　圆锥曲线的导向形式

这里所示的计算圆锥样条算法使用了一个圆锥曲线的导向形式，以表示由三条直线约束的圆锥曲线，如图 7．11 所示．直线方程为

$$a_0+a_1x+a_2y=0 \tag{7．18}$$

设多直线段中的第一和最后一个顶点为 A 和 B，C 是多线段的中间顶点，用弦 AB 连接第一和最后一个顶点．圆锥曲线导向形式是端点位于 A 和 B 的圆锥体的家族，正切 AC 和 BC 由下面方程定义：

$$(a_0+a_1x+a_2y)(b_0+b_1x+b_2y)=\rho(u_0+u_1x+u_2y)^2 \tag{7．19}$$

其中

$$a_0+a_1x+a_2y=0 \tag{7．20}$$

是包含直线段 AC 的直线，而

$$b_0+b_1x+b_2y=0 \tag{7．21}$$

是包含直线段 BC 的直线，以及

$$u_0+u_1x+u_2y=0 \tag{7．22}$$

是包含弦 BC 的直线．　圆锥曲线的家族由参数 ρ 来定义．

圆锥曲线拟合边缘表的算法是从直线段开始的，并把顶点分为角点(corner)、软顶

点(soft vertex)或结点(knot)三类．软顶点对应的角接近180°，相邻的直线段几乎共线，并可以用圆锥曲线来代替．软顶点序列对应于缓变方向角的直线段序列，很可能是对光滑曲线的采样点进行拟合的结果．角点对应的顶点角大于$180^\circ + T_1$或小于$180^\circ - T_1$，其中T_1是阈值，角点不可能成为圆锥曲线的一部分．结点位于某一条直线段上，该直线段的端点上具有软顶点．一个圆锥曲线不可能有拐点，所以两个圆锥曲线在结点处连接．结点在直线段上的位置由直线段端点的软顶点相对角度确定．设两个软顶点V_i和V_{i+1}的角度分别是A_i和A_{i+1}．如果$A_i = A_{i+1}$，那么结点可以位于顶点的中间，也就是说，在方程7．16中，$v = 1/2$．如果角度不相等，那么，结点位置应该偏离具有较大角度的那个角点，这是由于圆锥曲线不可能偏离直线段而足够迅速地弯曲去跟踪角点．方程7．16的v值可以由下式计算：

$$v = \frac{A_1}{A_1 + A_2} \tag{7．23}$$

由软顶点连接的每一个直线段序列可由穿过第一和最后一个顶点（或结点）的导向圆锥代替．第一和最后一个直线段的方向角定义了正切．正切和端点确定圆锥5个自由度的4个．让圆锥穿过位于序列中央的软顶点可以完全确定圆锥．

7．3 样条曲线

早期绘图员在制图时，使用一种富有弹性的细长条，称为样条。用压铁将样条固定在若干样点上，然后沿样条可以画出很光滑的曲线，人们将该曲线称为样条曲线．从数学上讲，样条曲线是用分段多项式表示的一个函数，在其连接点处具有连续的一阶和二阶导数．

样条曲线有着许多应用．在数据分析中，当没有合适的函数模型时，可以选用样条函数拟合数据点．在计算机图形和计算机辅助设计中，样条函数用来表示自由曲线．在机器视觉中，若没有表示曲线的合适模型时，样条函数可以提供曲线的通用表示形式．

需要指出，几何等效和参数等效是两个不同的概念．两条曲线在几何上等效，是指它们连接相同的点集，即它们在空间上对应着相同的形状（或点集）．如果两条曲线的方程一样，则两条曲线在参数上等效．显然，参数等效性比几何等效性更稳定．两条曲线可以是几何上等效但可以具有不同的参数表示式，这是机器视觉中曲线拟合的一个重要概念．比如，机器视觉系统可以产生基于三次样条曲线的表示，其在几何上非常接近于物体轮廓的真实表示，但在参数意义上，表示可能完全不同．在物体识别应用方面和工业零件图像与其模型匹配应用中，通过比较三次样条曲线的参数形式实现匹配几乎是

不可能的，在这种情况下，比较必须基于几何等效性.

7．3．1　三次样条曲线

样条函数最常见的形式是三次样条函数，它是分段三次多项式的一个序列. 上一节所讨论的直线段、圆弧段和圆锥曲线序列是样条函数的典型实例. 三次样条函数可以用很少几个样条段表示很复杂的曲线. 三次样条函数已经广泛用于计算机图形领域和字符轮廓表示.

三次样条具有足够的自由度来逼近边缘段位置和方向. 我们知道，大多数边缘检测算法同时提供边缘方向（梯度角）和边缘位置估计. 在直线段、圆弧段和圆锥曲线的拟合中，仅仅使用了边缘的位置信息. 下面介绍一种三次样条曲线拟合中如何使用由边缘检测器产生的方向信息的例子.

平面三次曲线方程如下:

$$\mathbf{p}(u) = (x(u), y(u)) = \mathbf{a}_0 + \mathbf{a}_1 u + \mathbf{a}_2 u^2 + \mathbf{a}_3 u^3 \qquad (7.\ 24)$$

其中系数 $\mathbf{a}_0$，$\mathbf{a}_1$，$\mathbf{a}_2$，$\mathbf{a}_3$ 是二元向量，参数 u 取值范围在 0 和 1 之间. 三次曲线起始点为 $\mathbf{p}(0) = (x(0), y(0))$，终结点为 $\mathbf{p}(1) = (x(1), y(1))$. 三次样条是由三次曲线段 $\mathbf{p}_1(u)$，$\mathbf{p}_2(u), \ldots,$ $\mathbf{p}_n(u)$ 构成的一个序列，这一序列定义在连续间隔 $[0,1]$，$[1,2], \ldots,$ $[n-1, n]$ 上，并将端点连结起来使得在端点处 $\mathbf{p}_i(u) = \mathbf{p}_{i+1}(u)$，见图 7．12. 样条中的每一个三次曲线段称为样条段，连结样条段的边缘点称为结点.

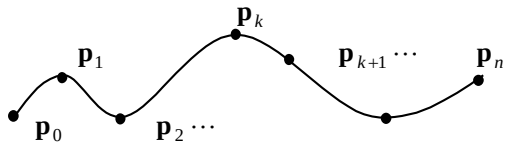


图　7.12　$n+1$ 个控制点的分段连续三次样条插值曲线

和前面讨论的曲线拟合算法一样，把边缘点序列分成一个个子序列，每一个子序列的第一个和最后一个边缘点为样条曲线的结点，然后再用样条段拟合这些结点. 由公式 7．24 可知，样条中每一个三次曲线段都需要确定四个二维向量共计八个参数，其中，曲线段的两个端点（或结点）共提供四个约束，结点处的一阶连续性（等正切向量）提供另两个约束，结点处的方向信息仅提供一个附加约束（由于结点由两个样条段共享），

结点处的二阶连续性（等曲率）提供两个约束，这样所产生的方程数量为九个，多于三次样条段所需的八个参数.

在结点处光滑连接样条段是非常重要的。在计算机图形学中，光滑连接是通过增加二阶连续性来实现．由上面分析可知，二阶连续性会提供两个约束，从而对样条段产生超约束。为了避免超约束，同时又要使结点处光滑，可以采用结点处二阶不连续性的极小化条件，也就是说把结点处的曲率差值极小化作为一个约束代替二阶连续性提供两个约束。

对于整个三次样条曲线，求 n-1 个结点处二阶导数差值平方和的极小化：

$$\chi^2 = \sum_{i=1}^{n-1} (\Delta \ddot{\mathbf{p}})^2 \tag{7．25}$$

其中两个样条段在其公共结点处的二阶导数差值是

$$\begin{aligned} \Delta \ddot{\mathbf{p}} &= \ddot{\mathbf{p}}_{i-1}(1) - \ddot{\mathbf{p}}_i(0) \\ &= 2(\mathbf{t}_{i-1} + 4\mathbf{t}_i + \mathbf{t}_{i+1} + 3(\mathbf{p}_{i-1} + \mathbf{p}_i)) \end{aligned} \tag{7．26}$$

变量 $\mathbf{t}_i$ 是结点 i 处的正切向量，该正切向量由边缘方向（梯度角）$\hat{\mathbf{t}}_i$ 和一个有正负号的未知参数 γ_i 来决定：

$$\mathbf{t}_i = \gamma_i \hat{\mathbf{t}}_i \tag{7．27}$$

换句话说，在结点处的边缘方向可以用一个单位向量表示，但三次样条需要具有大小及正负号的正切量，以表示穿越结点的方向和速度．用该算法求解线性系统方程，可以得到 n 个未知数 γ_i，由此提供构造结点之间三次样条段的丢失信息.

本算法没有任何附加参数和阈值，但是，同前面介绍的多线段、圆弧段和圆锥截面拟合算法一样，结点必须从边缘表中选出．使用上面所述的多线段算法进行轮廓多线段逼近可以确定结点位置。多线段顶点也可以作为结点的位置。调节结点的位置和数量可以改善三次样条对整个边缘点集的拟合.

三次样条拟合算法仅仅需要求解一个小的线性系统就可以得到正切值的符号和量值，因此该算法十分有效．如果需要的话，可以使用许多交互式图形界面，在其上可以很方便地调节三次样条曲线.

7．3．2 B样条曲线

B样条曲线是由结点引导的逐段多项式曲线，是一种平滑和内插技术．B样条与上述样条曲线不同，它不必通过结点，我们将这种结点称为引导结点．B样条曲线的三次多项式是最常用的，因为这些样条曲率连续度为最低．已知引导结点序列为 $\mathbf{p}_0$，$\mathbf{p}_1$，...，

$\mathbf{p}_n$，则三次 B－样条多项式为：

$$\mathbf{P}_i(t)=B_0(t)\mathbf{p}_{i-1}+B_1(t)\mathbf{p}_i+B_2(t)\mathbf{p}_{i+1}+B_3(t)\mathbf{p}_{i+2} \qquad (7.28)$$

式中 B_0， B_1， B_2， B_3 为参数 $t(0\le t\le 1)$ 的三次多项式．下面确定上式四个多项式所对应的 16 个参数．对任意的 $\mathbf{p}_k(k=i-1,i,\cdots,i+3)$，相邻曲线段 $\mathbf{P}_i(t)$，$\mathbf{P}_{i+1}(t)$ 以及二阶导数必须连续的条件提供 15 个等式，例如，$\mathbf{P}_i(1)=\mathbf{P}_{i+1}(0)$，则有：

$$\begin{aligned}&B_0(1)\mathbf{p}_{i-1}+B_1(1)\mathbf{p}_i+B_2(1)\mathbf{p}_{i+1}+B_3(1)\mathbf{p}_{i+2}\\&=B_0(0)\mathbf{p}_i+B_1(0)\mathbf{p}_{i+1}+B_2(0)\mathbf{p}_{i+2}+B_3(0)\mathbf{p}_{i+3}\end{aligned} \qquad (7.29)$$

并提供以下五个等式：

$$B_0(1)=0，\ B_1(1)=B_0(0)，\ B_2(1)=B_1(0)，\ B_3(1)=B_2(0)，\ B_3(0)=0$$

$\mathbf{P}_i(t)$ 的形状对坐标转换不变的条件提供了另一个方程：

$$B_0(t)+B_1(t)+B_2(t)+B_3(t)=1 \qquad (7.30)$$

通过解 16 个等式，可以得到如下 16 个系数：

$$\begin{aligned}B_0(t)&=\frac{1}{6}(1-t)^3\\B_1(t)&=\frac{1}{6}(3t^3-6t^2+4)\\B_2(t)&=\frac{1}{6}(-3t^3+3t^2+3t+1)\\B_3(t)&=\frac{1}{6}t^3\end{aligned} \qquad (7.31)$$

显然这四个多项式都是非负的，根据式（7.30），曲线上的任何点都是四个控制结点的插补，即曲线段包含在由四个控制结点确定的方块内．如图 7．13 所示．通常，m 次的 B 样条包含在由 $m+1$ 控制点确定的多边形内。由于 B 样条的优良特性和对于任何控制点导数的连续性，故在实际中较为常用。

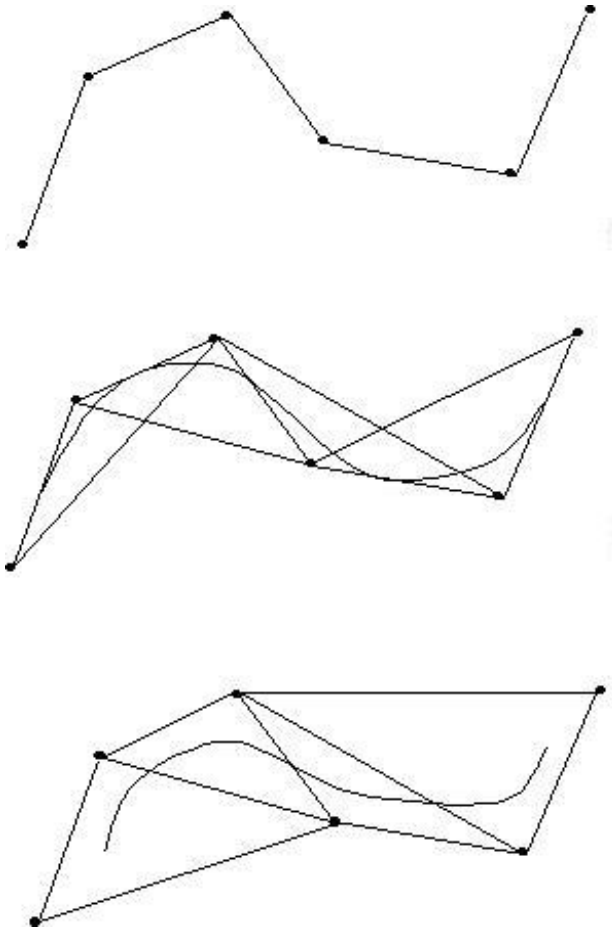


图 7．13 B样条曲线示意图。（a）直线，(b) 二次B样条曲线，（3）三次B样条曲线

7．4 曲线回归逼近

目前有许许多多曲线逼近方法，其区别主要取决于边缘点组成轮廓点的可靠度。如果将所有边缘点连接起来可以组成一个轮廓，那么就可以用最小二次回归法来进行边缘点曲线拟合．如果存在组合误差（Grouping Error），则可以使用鲁棒回归方法来进行曲线逼近．如果边缘点组成的轮廓不可靠，或者边缘点太分散以至于无法用连接或跟踪方法来组成轮廓，那么必须使用聚类分析（Cluster Analysis）技术同时进行边缘点的组合和曲线逼近．同时用组合和曲线逼近方法处理分散边缘点算法的最好例子是 Hough 变换，这一内容将在下面讨论

前面几节讨论的直线段、圆弧段、圆锥曲线和三次样条拟合边缘点的方法属于一般回归问题，主要用于端点之间的曲线拟合．这些算法都假定边缘的位置可以精确计算，甚至可能使用子像素计算方法．这种回归算法没有使用端点之间的边缘点信息，因此曲

线的拟合精度主要取决于端点的位置精度．本节讨论的曲线最佳逼近方法将使用所有边缘点信息．

一般曲线拟合问题是一个曲线的回归问题，曲线用 p 个参数的隐函数表示：

$$f(x,y;a_1,a_2,\ldots,a_p)=0 \qquad (7.32)$$

曲线预估问题就是指这种曲线模型拟合一个边缘点集 $\{(x_1,y_1),(x_2,y_2)$，$\ldots,(x_n,y_n)\}$．

在无噪声情况下，通过 p 次观测产生的 p 个方程来求解 p 个未知曲线参数．不幸的是在大多数场合，由于噪声的原因，这一直接方法不是很有效．实际的应用常常需要使用边缘表中所有信息来获取这些参数的最佳估计．

下一节我们将讨论最小二次回归方法，这种方法在计算机视觉中主要用于曲线拟合．当误差服从正态分布时，最小二次方法是最合适的拟合方法．7.4.3 将讨论曲线拟合的鲁棒性方法，这种方法在一些边缘点被错误地连接到某一轮廓时特别有用．这种被错误地连接的点被称为局外点（Outlier）．

７．４．１ 全回归方法

传统的线性回归方法仅仅在一维独立变量空间中对数据和模型差值求极小化．比如，一个函数模型具有下列形式，

$$y=f(x;a_1,a_2,\ldots,a_p) \qquad (7.33)$$

上式把独立变量 y 和独立变量 x 联系起来，具有 p 个模型参数（$a_1\ldots a_p$），并假定独立变量没有误差．在机器视觉中，x 和 y 的坐标位置误差很可能相等，其对应的曲线模型是一条垂直线，这样就无法用上述方程来表示．在机器视觉中，边缘点的直线和曲线模型拟合是通过使用全回归方法实现的，即对所有的数据点与回归模型之间距离的平方和进行极小化，这一技术的优点是能够补偿 x 和 y 方向的误差．

为了避免垂直直线引起得问题，可以使用如下极坐标形式来表示直线方程：

$$x\cos\theta+y\sin\theta-\rho=0 \qquad (7.34)$$

求所有点 (x_i,y_i) 到该直线垂直距离的平方和极小化：

$$\chi^2=\sum_i\left(x_i con\theta+y_i\sin\theta-\rho\right)^2 \qquad (7.35)$$

全回归问题的解是：

$$\rho=\bar{x}con\theta+\bar{y}\sin\theta \qquad (7.36)$$

其中

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{7．37}$$

全回归直线方向角为θ，由下式给出：

$$\tan 2\theta = a / b \tag{7．38}$$

其中

$$a = 2\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$$b = \sum_{i=1}^{n}(x_i - \bar{x})^2 - \sum_{i=1}^{n}(y_i - \bar{y})^2$$

全回归方法使用了最小二次误差范数，如果误差服从正态分布，则误差范数是最优的。该方法不适用于数据中有局外点的情况．在用曲线模型拟合边缘点时，如果边缘连接算法从其它轮廓上吸收一个或一个以上边缘点到本轮廓边缘表中，即产生了边缘局外点局外点．即使边缘连接算法完美无缺，也免不了局外点的产生．例如，考虑对应一个矩形两条邻接边的边缘表，在用直线拟合边缘前，必须识别出角点以便把边缘表分割两条边．如果不能正确地识别角点，那么，一些边缘点可能被分配给错误的边，这些边缘点即为局外点．

通常在分类时产生的误差会引入到回归问题里，其中的误差不服从正态分布．在这种情况下，误差可以由一个混合模型来表示，即把描述正态分布误差的高斯分布函数和描述由于分类不理想而引起的局外点分布的拖尾分布函数混合起来．

7．4．2 角点估计

角点估计的最佳方法是使用直线段逼近边缘点来求出直线段序列，然后计算直线段之间的交点．这一方法补偿了由边缘检测算子对角点的平滑作用而引入的误差，并且比利用局部信息的角点检测算子求出的角点更精确．

已知两条直线的隐函数方程为

$$a_1 x + b_1 y + c_1 = 0 \tag{7．39}$$

$$a_2 x + b_2 y + c_2 = 0 \tag{7．40}$$

交点坐标为：

$$x = \frac{c_1 a_2 - c_2 a_1}{a_1 b_2 - a_2 b_1} \qquad y = \frac{c_2 b_1 - c_1 b_2}{a_1 b_2 - a_2 b_1} \tag{7．41}$$

如果$a_1b_2 - a_2b_1$非常接近零，那么两条直线几乎平行，没有交点.

一种较好的探测角点方法是沿着轮廓用一对直线拟合$2n+m$个边缘点中的一个个连续子表，其中，参数n是边缘点数量，用于精确直线拟合；参数m是位于角点周围不予考虑的边缘点的数量，即跨越角点圆弧部分的边缘点. 设置一个阈值，检测$a_1b_2 - a_2b_1$的幅值是否大于该阈值即可决定角点是否存在.

7．4．3 鲁棒回归法

如果误差不服从正态分布，那么最小二乘法就不是最佳拟合方法. 图 7．14 所示的是数据集合包含一些局外点时，使用最小二乘法所遇到的问题的一个例子. 在实际中，可能一个局外点就足够把回归直线推向远离其正确的位置. 鲁棒性回归方法要对数据的各种子集进行测试，从中选择一个产生最佳拟合的子集.

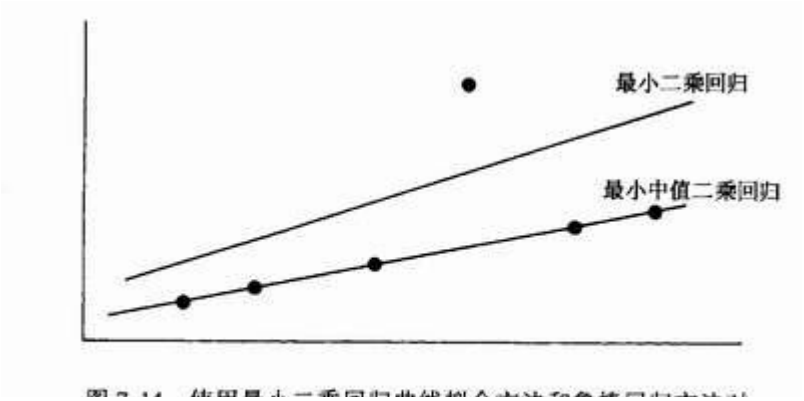


图 7．14　使用最小二次回归曲线拟合方法和鲁棒性回归方法对包含有局外点的一组数据进行拟合的差别

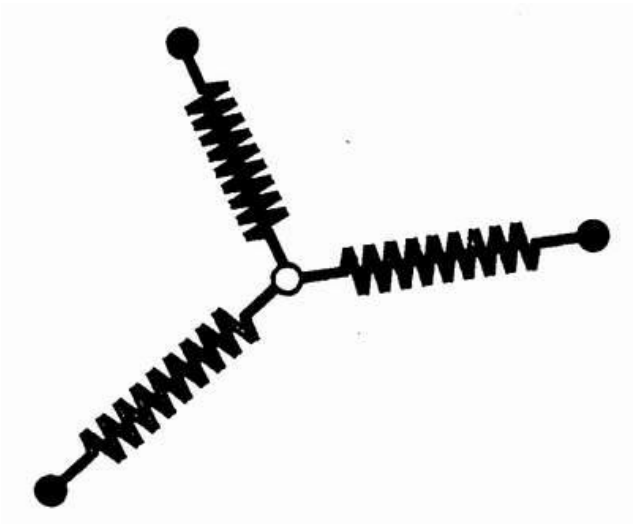


图 7．15　最小二次方法对局外点敏感度的物理模拟图[Jain].

图 7．15 是一个物理模拟图，可以使问题变的更加清楚．设想一下你准备求出平面上一组点的质心．把具有相同弹簧系数的若干弹簧的一端连接到一个可以自由运动的物体上，另一端分别系在不同的固定点上，物体将会被拉到一个平衡（平均）位置上．弹簧将通过弹簧势能方程进行最小二乘法范数运算．这个物理模拟对应于一个均值计算的微分，其中均值是指残差平方和（即点与均值之差的平方和）极小化判据中的均值．现在假定其中的一点可以移动，并称这一点为杠杆点．用力扯动杠杆点到足够远的地方，可以迫使平衡点移动到任意一个位置．这一现像说明基于最小二乘法的预估器对局外点及其敏感．理想情况下，人们希望把弹簧和局外点隔开，使得估计值不受伤害．改变弹簧常数使得局外点对估计值影响最小，可以实现基于影响函数的鲁棒性预估．割断弹簧与局外点的联系对应于再取样策略，其中一致性子集取样是确定的．再取样方法是指重复抽取随机子集并选择可以产生最佳估计的子集．再取样算法的例子包括随机抽样一致性、最小中值二次回归、以及其它计算机中常用的回归方法．

弹簧模拟方法推广到线性回归系统也有一样的结论，即单一的局外点可以扭曲回归估计．对第 i 个数据点，n 阶线性多变量模型可以用下面的方程表示：

$$y_i = \theta_1 x_{i1} + \theta_2 x_{i2} + \ldots + \theta_n x_{in} \tag{7．42}$$

其中 $\hat{\theta}_i$ 是模型参数 θ_i 的估计值．每一点的残差（预估模型数据点的微分）是 $r_i = y_i - \hat{y}_i$．在最小二次回归算法中，模型参数的估计由残差平方和的极小化求得：

$$\min_{\theta} \sum_{i=1} r_i^2 \tag{7．43}$$

同上述的弹簧模拟方法一样，如果仅有一个数据点是局外点，模型参数可能是任意的.

通常，噪声和局外点可以用联合分布表示，即噪声的正态分布和局外点的宽尾分布的线性组合．在这种情况下可以清楚的看到，预估器的范数同小误差的最小二次范数一样，同时对大误差不敏感，这样就可以忽略局外点的影响．这种方法称为影响函数法.

为了定量说明局外点对函数逼近的影响，我们引进溃点（breakdown point）这一术语。溃点是指局外点占整个数据的一个最小比例值，当局外点数不超过这个值时，无论局外点如何不正确，都不会使预估算法产生任意的错误估计[207]．设 Z 是 n 个数据点的集合，将集合 Z 中任意 m 个点的坐标设置成任意值（局外点），构成一个含有 m 个任意点的集合 Z'．设一个回归预估器 $\hat{\theta}=T(Z)$，由局外点造成的预估偏差为：

$$Bias=\sup_{Z'}\left\|T(Z')-T(Z)\right\| \tag{7．44}$$

设置溃点的基本思想是定量分析局外点的数量 m 在数据点数量 n 中所占比例增加时对 *Bias* 值的影响程度．当 m 增加到一定数量时，*Bias* 无界，这就是溃点．当低于溃点时，回归估计器可以完全拒绝局外点，或使得局外点对预估结果影响很小．当高于溃点时，局外点可以驱使预估器产生任意的解案，答案将取决于局外点而不是合法数据．换句话说，预估器所提供的结果是不可预估的．溃点定义为：

$$\varepsilon_n^*=\min\left\{\frac{m}{n}:Bias(m;T,Z)\text{是有限值}\right\} \tag{7．45}$$

对于最小二次回归，$\varepsilon_n^*=1/n$，在极限情况下，当数据点数量变的很大时，$\varepsilon_\infty^*=0$．换句话说，最小二次回归方法对局外点无免疫能力，一个局外点即可毁掉结果.

最小二次中值回归方法是一种非常简单的技术，并被证明是解决大量局外点回归问题的非常有效的方法，算法 7．3 概括了该算法。最小中值二次方法可以容错高达百分之五十的局外点，也就意味着数据点集内，有一半的数据可以取任意值而不会严重地影响回归结果．如果有多于百分之五十的点为局外点，最小二次回归方法不会工作的很好，此时需要更有效的方法，如 Hough 变换.

在最小中值二次回归方法中，模型参数的估计由极小化残差平方的中值求得：

$$\min_{\hat{\theta}}\operatorname{median}_i\left(r_i^2\right) \tag{7．46}$$

算法 7．3　最小中值二次回归方法

假定有 n 个数据和 p 个参数的线性模型

在 n 各个数据点集中，随机地选择 p 个点.

用模型拟合 p 个点.

计算残差平方的中值.

拟合过程重复进行直到得到足够小的残差平方中值,或者达到预定的再取样步长数值.

7．5 Hough 变换

最近几年，使用表决原理的参数估计技术应用不断增加．最常用的表决方法之一是 Hough 变换．在 Hough 变换中，曲线上的每一点可以表决若干参数组合，赢得多数表决的参数就是胜者．下面考虑一下直线拟合数据的方法．直线方程为：

$$y = mx + c \tag{7．47}$$

上述方程中，x 和 y 是观测值，m 和 c 表示参数．如果已知参数值，则该点坐标之间的关系即可确定．把上述方程重新表示为：

$$c = y - mx \tag{7．48}$$

假定 m 和 c 是我们感兴趣的变量，而 x 和 y 是常数，则上述方程表示的是 (m,c) 空间中的一条直线，斜率和截距由 x 和 y 决定．点 (x,y) 对应于 (m,c) 空间的直线，如图 7．16(a)所示．如果直线上有 n 个点，那么这些点对应参数空间 (m,c) 上的一个直线族，且所有直线都经过 (m,c) 上的一点，该点的坐标当然反应 (x,y) 空间上的直线参数，如图 7．16(b)所示．

需要指出，参数空间曲线的形状取决于用于表示曲线的原始函数．实际中，常常使用直线的极坐标形式，而不是其显式表示，这样可以避免直线是垂直线时带来的问题．直线的极坐标方程表示如下：

$$\rho = x\cos\theta + y\sin\theta \tag{7．49}$$

上述方程中，点 (x,y) 被映射到 (ρ,θ) 空间上，如图 7．16（c）所示．如果直线上有 n 个点，那么这些点对应参数空间 (ρ,θ) 上的一个正弦曲线族，且所有正弦曲线都经过 (ρ,θ) 上的一点。

如果我们对求 n 点的最佳直线拟合感兴趣，那么，我们就可以使用上述图像空间到参数空间的映射．这种方法称为 Hough 变换．在这种方法中，我们把参数空间表示为一个累加器阵列，表示离散参数值．依照变换方程，图像中的每一点可以表决几个参数．为了求出表征直线的参数，我们应该探测参数空间的峰值．这种一般的想法概括在算法 7．4 中．

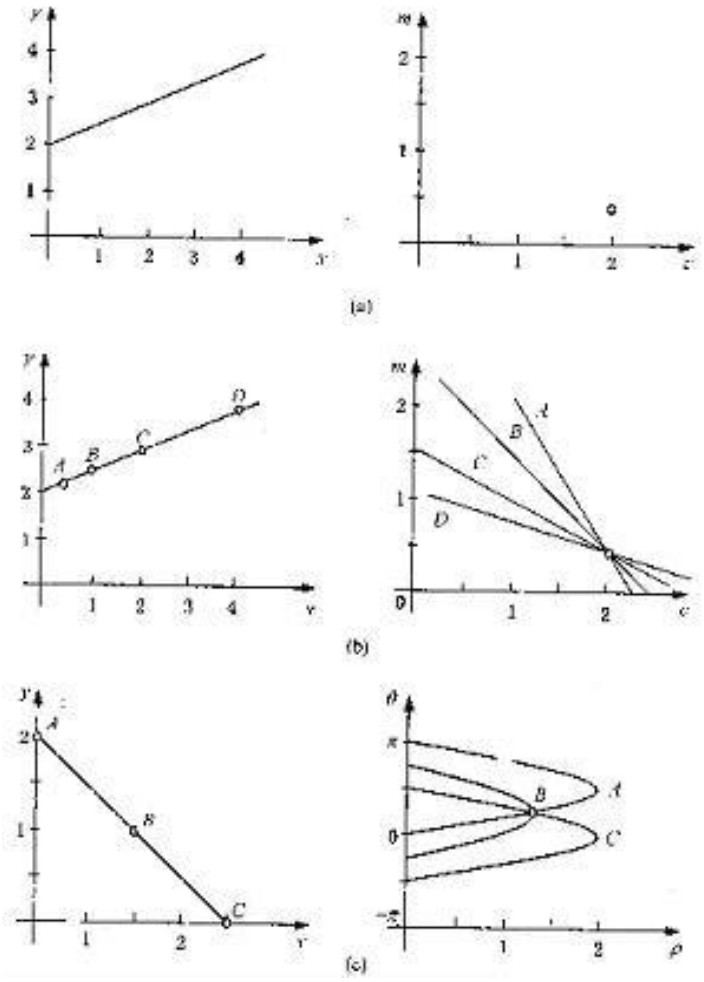
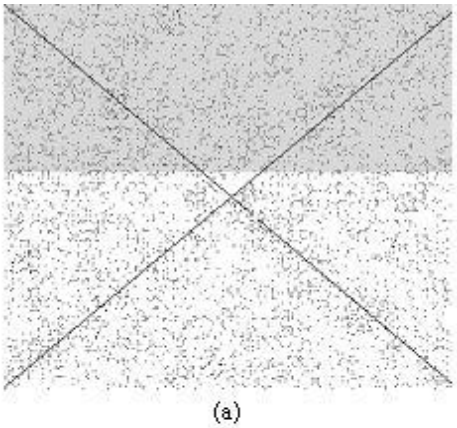


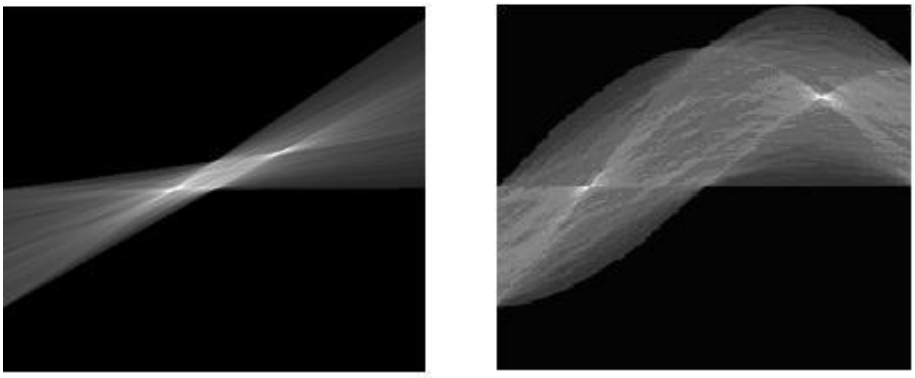
图 7．16 Hough 变换示意图。左边为图像空间，右边为 Hough 变换空间。

（a）一条直线对应一个点，(b) 一条直线上的多个点对应多条交于一点的直线

(c) 一条直线上的多个点对应多条交于一点的正弦曲线



(a)



(b)

图 7.17　Hough 变换算法实验结果。（a）合成图像，（b）直角坐标参数空间映射图，（c）极坐标参数空间映射图

算法 7．4　Hough 变换算法

适当地量化参数空间．

假定参数空间的每一个单元都是一个累加器，把累加器初始化为零．

对图像空间的每一点 (x, y)，在其所满足的参数方程对应的累加器上加 1．

累加器阵列的最大值对应模型的参数．

Hough 变换不需要预先组合或连结边缘点．位于感兴趣曲线上的边缘点可能构成图

像边缘的一个小部分．特别指出，Hough 变换可以允许位于曲线上的边缘数量少于实际的边缘数量，而大多数鲁棒性回归算法是无法适用这种情况．Hough 变换所基于的假设是在大量噪声出现的情况下，最好是在参数空间中去求满足图像边缘最大数量的那个点．如果参数空间的峰值包括了一个以上的累加器，则包含峰值的区域的矩心就是参数的一个估值． 利用上述算法的实验结果见图 7.17

如果图像中有几条曲线和给定模型相匹配，则在参数空间中会出现几个峰值．此时，可以探测每一个峰值，去掉对应于某一个峰值的曲线边缘，再检测余下的曲线，直到没有明显的边缘．但是，确定峰值的显著性是件很困难的事．

Hough 变换的另一个问题是离散参数空间随着参数的数量增加迅速增加. 对一个圆弧段，参数空间是三维的，对其它曲线，其维数可能更高．由于累加器的数量随着参数空间的增加成指数增加，Hough 变换可能对于复杂模型的计算效率很低．已经提出几种方法来改进 Hough 变换的性能．一种方法是使用边界梯度信息来减少参数空间的工作量．假定曲线模型是圆．模型有三个参数，二个参数为圆的圆心坐标，另一个是圆的半径．如果可以得到边缘梯度角，就可以提供减少自由度数量的约束，从而得到所要求的参数空间尺寸．从圆的中心到每一个边缘点的向量方向由梯度角确定，剩下的未知参数只有圆的半径．其它一些减少参数空间的方法也在视觉中得到应用．

下面介绍一下使用梯度角减少参数尺寸的圆拟合方法。算法过程见算法 7．5．圆的方程为：

$$(x-a)^2+(x-b)^2=r^2 \qquad (7.50)$$

圆的极坐标方程为：

$$\begin{aligned} x &= a + r\cos\theta \\ y &= b + r\sin\theta \end{aligned} \qquad (7.51)$$

则圆的参数为：

$$\begin{aligned} a &= x - r\cos\theta \\ b &= y - r\sin\theta \end{aligned} \qquad (7.52)$$

在边缘点 (x,y) 处给定梯度角θ ，计算 conθ 和 sinθ ．从上述方程中消除半径，得到：

$$b = a\tan\theta - x\tan\theta + y \qquad (7.53)$$

这是一个用于升级参数空间累加器的方程．对于每一个在 (x,y) 处并具有边缘方向角θ 的边缘点来说，沿着方程 7.53 给定的直线，在 (a,b) 参数空间给累加器一个增量. 如果半径已知，则只需由方程(7．52)在 (a,b) 点处给累加器一个增量．

这里无需使用参数来描述由 Hough 变换检测的曲线．Hough 变换可以一般化为表

决算法（见算法 7.6），这种算法可以有效地完成模板匹配．算法 7．6 把物体的边界形状编码成表，以便有效地进行读取操作．其中物体上的一点被选为参考点．根据定义，图像中参考点的位置就是物体的位置．对于每一个在 (x,y) 处并具有梯度角θ 的图像梯度点，参考点可能的位置有下面的方程给出：

$$\begin{aligned} a &= x - r(\theta)\cos(\alpha(\theta)) \\ b &= y - r(\theta)\sin(\alpha(\theta)) \end{aligned} \qquad (7.54)$$

给每一个可能的参考点的位置一个增量.参数空间的峰值位置是物体位置的一个估计．把这一技术推广到可以处理尺度和旋转变换不太容易.

算法 7．5 圆拟合算法

量化参数空间 a 和 b.

置累加器阵列 $M(a,b)$ 为零.

计算梯度值 $G(x,y)$ 和角 $\theta(x,y)$.

对于 $G(x,y)$ 中的每一个边缘点，沿着直线（见方程 7.53）给累加器阵列 $M(a,b)$ 中的所有点一个增量.

累加器中局部最大值对应于图像中的圆中心.

算法 7．6 广义 Hough 变换算法

在物体上取出一个参考点.

沿着物体的边界计算梯度角 θ_i .

对每一个梯度角 θ_i，存储对于参考点的距离 r_i 和角度 α_i .

7．6 傅里叶描述子

由于沿着封闭轮廓的位置函数是周期性的，因此傅里叶级数可以用来逼近轮廓．轮廓逼近的分辨率由傅里叶级数的项数来确定.

假定物体的边界表示为一个坐标的序列 $u(n)=[x(n),y(n)]$，其中 $n=0,1,2,\ldots,N-1$．用复数来表示每一个坐标对，即

$$u(n)=x(n)+jy(n) \qquad (7.55)$$

其中 $n=0,1,2,\ldots,N-1$． 换句话说，x 轴为实轴，y 轴是复数序列的虚轴．注意，对于封闭边界，这一序列是周期的，其周期是 N，这样，边界就可以在一维空间表示.

一维序列函数 $u(n)$ 的离散傅里叶变换（DFT）定义为：

$$u(n)=\sum_{k=0}^{N-1}a(k)e^{\frac{j2\pi kn}{N}},\quad 0\le n\le N-1 \qquad (7.56)$$

$$a(k)=\frac{1}{N}\sum_{n=0}^{N-1}u(n)e^{-\frac{j2\pi kn}{N}},\quad 0\le n\le N-1 \qquad (7.57)$$

复数系数 $a(k)$ 称为边界的傅里叶描述子.

傅里叶描述子是轮廓表示的简洁表示方法. 然而，更简洁的表示形式是使用傅里叶序列的低阶项的低分辨率逼近. 如果仅使用前 M 个系数，即等同于 $a(k)=0, k>M$，则对 $u(n)$ 的逼近如下：

$$\hat{u}(n)=\sum_{k=0}^{M-1}a(k)e^{\frac{j2\pi kn}{N}},\quad 0\le n\le N-1 \qquad (7.58)$$

尽管仅使用 M 项来求边界 $u(n)$ 的每一项，n 仍然取 0 到 $N-1$的区间. 换句话说，被逼近的边界具有相同的点数，但用于重建每一个点的项数不同.

边界的简单几何变换，如平移、旋转和尺度变换，将关系到边界傅里叶描述子的简单操作（见练习 7．19）. 这使得傅里叶描述子在边界匹配中的应用非常具有吸引力. 然而，傅里叶描述子在描述具有遮挡物体是的确存在一些问题. 使用其它边界表示也可以得到类似的描述子.

习题

7.1 什么是轮廓？轮廓与区域有什么关系？一个非封闭轮廓表示什么？

7.2 请列出选择轮廓表示的判据，并讨论这些因素在物体识别中的含义.

7.3 插值和逼近方法有何区别？那一种方法更好？

7.4 为了完成 $n\times 22.5$ 的旋转，可能使用 16 方向的链码编码. 你如何来完成这一编码？为什么 8 方向链码是最常用的链码？

7.5 逆时针求取图 7．19 所示物体的 8-方向链码和链码差分，其中的空心圆点是起点.

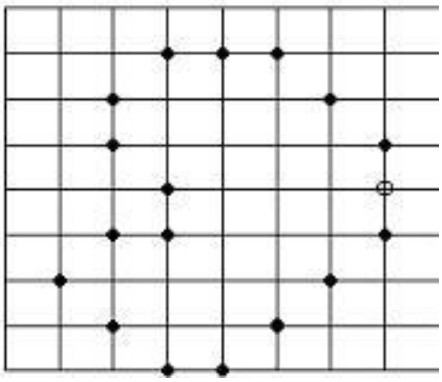


图 7.19　思考题 7．5 的轮廓

7.6　考虑一下 7．8．2 节的角点位置估计方法．当两条直线成直角时，$a_1b_2 - a_2b_1$ 的值是多少？当两条直线成某一角度θ 时，又为何值？（这是用于设置角点探测阈值的公式）

7.7　考虑一下 7．8．2 节的角点位置估计方法．假定边缘位置 x 和 y 的坐标误差服从方差为 σ_2 的正态分布．角点的位置误差分布是什么？角点对应的角度是如何影响该误差的？

7.8　产生一个相对于均匀背景的均匀强度的人工图像．请使用第五章任何一种边缘检测器计算边缘，并用多线段拟合边缘．请问所求得的顶点与真实角点位置的差距有多大？这种误差是否一定偏向一个方向？

7.9　为什么 $\Psi - s$ 图被认为是连续链码？其最吸引人的特点是什么？

7.10 如何使用 $\Psi - s$ 图来比较两个物体？

7.11 讨论一下你所使用的不同的逼近算法误差测量．请问在逼近算法中，误差算法测量起什么作用？

7.12 为什么说三次样条相对于多线段和圆锥曲线段是一种更有效的表示方法？

7.13 试论述几何等效和参数等效的差别．

7.14 为什么在全回归方法中使用最小二次测量方法？试列出它的优点和缺点．

7.15 试说明鲁棒性方法是如何克服全回归方法的局限性？为什么鲁棒性方法在逼近算法中用的不是很普遍？

7.16 傅里叶描述子在逼近和表示封闭曲线时的优点和弱点是什么？

7.17 什么是 Hough 变换？它是否与鲁棒性回归方法有关？为什么？

7.18 你能把 Hough 变换推广到检测任意物体形状？如何把 Hough 变换用于探测旋转和

尺度变换后的物体？

上机作业题

6.1 编制一个程序求一条给定曲线的链码．你能否利用链码来确定角点？如果能，请完成这一算法，并用几幅图像来测试．

6.2 编制一个用 Hough 算法检测图像中的直线算法．使用这一算法来求一幅图像中的所有大于规定长度的直线段，设规定的长度为 20 点．

6.3 请产生一幅在均匀背景下的具有均匀灰度的矩形物体图像，使用第五章中任何一种边缘检测方法检测矩形物体的边缘，并用多线段拟合这一边缘．请问多线段中的顶点与真实的角点接近的程度如何？这些顶点与真实角点的偏差是否总是在一个方向上？

6.4 请产生一幅四个角点都为圆角的矩形图像，圆角可用四分圆表示．首先使用边缘探测和多线段拟合，然后使用本章介绍的圆弧段拟合方法替代多线段拟合，并测量圆弧段端点处的误差．误差是否对称？是否角点不均匀使得矩形变的扭曲？

6.5 完成用于直线拟合边缘点集表的最小中值二次回归算法．在边缘点集表中增加假边缘，以模拟成组误差．请画出估计线段与真实线段之间的距离相对于假边缘数量的曲线．

6.6 考虑一个相对于一个垂直轴对称的物体．假定可以得到物体左右两边的两个边缘段表．采取一种三次样条拟合边缘的算法以确保这一对称性约束，并把这一算法推广到轴在任意一个位置．

6.7 假定可以得到一个相对于某一个轴是对称的物体的轮廓边缘点集表．把边缘表处理成一个直线段和圆弧段序列．编制一个匹配直线段和圆弧段算法来检测物体是对称的，并估计出对称轴．在确定对称轴以后，用这一信息来改善直线段和圆弧段的估计．请编制一个细化轮廓表示的算法．做迭代算法探测对称轴的实验，用这一信息细化轮廓表示，然后用改进的轮廓表示细化轴位置估计，重复这一过程，直到轴和轮廓表示收敛．