

服装样片的自顶向下设计方法*

张全伙 范慧琳

(华侨大学计算机科学系, 泉州 362011)

摘要 介绍服装样片分类和款式的自顶向下设计方法. 用户利用鼠标操作, 通过对已有图形进行伸展和切割来构造样片. 方法基于封闭三次贝齐埃曲线和卡斯特利乌算法的实现.

关键词 贝齐埃曲线, 卡斯特利乌算法, 自顶向下, 交互式绘图, 样片

分类号 TP 391.72; TS 941.52

传统的交互式绘图通常采用自底向上方法, 用户通过选择一套离散的工具, 诸如直线、多边形、圆、椭圆、矩形、曲线等, 用它们来增强构图能力, 以便构造出更复杂的图形. 本文介绍的是采用自顶向下方法来构造一个图形^[1]. 用户开始选择一个基本形状, 例如一个矩形, 然后通过定义的一些基本操作, 对它进行分割、拉伸、压缩、移动等操作, 逐步改造成用户所需要的形状. 这种方法相当于给用户几片可变形的材料和一把剪刀. 现在, 利用计算机的交互图形系统可提供服装制造业中的款式设计, 给出某一尺寸服装的一组样片, 裁剪者就必须对这种款式的服装尺寸进行加工, 这是一个很细致的问题. 因为每个人的自身各部分尺寸不一样, 例如某人 X 比 Y 高 Z 个百分点, 但这并不表示 X 的手臂与脖子长均比 Y 多 Z 个百分点. 另外, 从服装的一组基本样片开始, 款式设计师要对这些样片进行编辑, 以形成不同的服装款式.

1 样片模型的数学描述

通常, 一个样片模型是通过数字化(一般为鼠标设备)输入到计算机中的. 一个模型的轮廓线是一条封闭曲线, 它可以用一个近似多边形来表示. 为消除原始模型和数字化之间产生错误轮廓线, 多边形要有足够多的采样顶点组成. 现在, 我们引入分段三次贝齐埃(Bezier)曲线, 并用它来表示样片模型曲线^[2]. 一段三次贝齐埃曲线是由四个控制点 P_0, P_1, P_2 和 P_3 表示的, 令曲线的参数为变量 $t(0 \leq t \leq 1)$, 则三次贝齐埃曲线上任一点由下式确定:

$$P(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3.$$

显然, 当 $t=0$ 时, $P(t)=P_0$; 当 $t=1$ 时, $P(t)=P_3$. 也就是说, 曲线通过第一个和最后一个控制点. 将 $P(t)$ 对 t 求一阶导数, 则可得到在贝齐埃曲线的起点有 $P'(0)=3(P_1-P_0)=3 \cdot P_0 P_1$, 在贝齐埃曲线的终点有 $P'(1)=3(P_3-P_2)=3 \cdot P_2 P_3$, 这说明贝齐埃曲线在起点和终点的切矢量方向分别与控制多边形的第一边和最后一边走向一致, 如图 1 所示. 下面我们考虑两段三次贝齐埃曲线的连接, 设第一段曲线由控制点 P_0, P_1, P_2 和 P_3 确定, 第二段曲线由 Q_0, Q_1 ,

* 本文 1996-02-28 收到; 国务院侨办科研基金资助项目

Q_2 和 Q_3 确定(图2). 由于在连接点处两段曲线应连续, 于是有 $Q_0 = P_3$; 又第二段曲线在起点的切矢量方向应与第一段曲线在终点的切矢量方向一致, 于是有 Q_0 (即 P_3), Q_1 , P_2 三点共线.

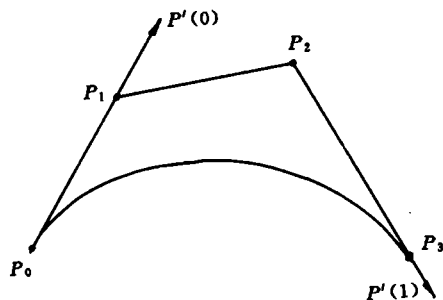


图1 贝齐埃曲线在起点和终点的切矢量

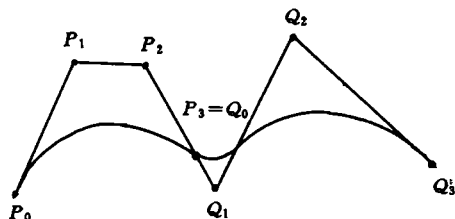


图2 两段三次贝齐埃曲线的连接

对贝齐埃曲线来说, 移动控制点可直观地改变曲线形状, 而且曲线形状与控制多边形有一定程度的近似. 当然, 对贝齐埃曲线的计算要用坐标分量形式, 考虑到我们所讨论贝齐埃曲线是 XOY 平面上的平面曲线, 写成坐标分量形式为

$$P_x(t) = (1-t)^3 P_{0x} + 3(1-t)^2 t P_{1x} + 3(1-t)t^2 P_{2x} + t^3 P_{3x},$$

$$P_y(t) = (1-t)^3 P_{0y} + 3(1-t)^2 t P_{1y} + 3(1-t)t^2 P_{2y} + t^3 P_{3y}.$$

样片模型用分段三次贝齐埃曲线表示后, 就可以对它进行一系列操作. 用户可以用鼠标器交互地选择任一个分段点并移动它, 点的移动不改变两曲线在分段点处的切矢量方向. 这样, 可保持整个样片的大致形状. 此外, 样片可通过改变其边界曲线的弯曲度来进行编辑, 这可通过显示曲线的切矢量来完成. 用户选择其中一个切矢量, 并通过遵守赫米特(Hermite)多项式的性质来改变曲线取向或长度. 所有操作都是基于卡斯特利乌(P. de Casteljau)算法的实现.

2 基本操作与用户界面

在服装款式设计系统中, 样片是封闭的连续曲线, 称为一片; 两曲线接合的顶点称为有效点, 对于一个片, 一条曲线, 一组有效点, 有下面三种基本操作.

(1) 移动(Moving): (a) 一个片可以移动到显示屏的不同位置; (b) 一条曲线可在某种意义上移动, 它的曲率可以改变, 而两端点不变; (c) 一个点可以在不改变两曲线在该点的切矢量方向情况下移动.

(2) 分裂(Splitting): (a) 一个片可以分裂成两片, 每片都表示一个有效的片. 片的分裂是由沿着该片被分裂的地方插入一条新曲线而实现的; (b) 一条曲线可由在其上插入一个新的有效点而分裂成两条(其端点除外); (c) 一个有效点可通过在原来位置上插入一条新曲线而分裂成两点.

(3) 消除(Removing): (a) 一个片可从图形中完全消除; (b) 一条曲线可以消除, 此时它的两个端点由一个单一的端点所代替; (c) 一个点可以消除, 此时它所邻接的两条曲线统一成一条单一曲线.

此外, 曲线和片可以“弄直”, 弄直一条曲线意味着用直线代替它, 弄直一个片意味着将该片上每条曲线都弄直.

现在,形象直观的图形界面已获得极大成功,上述操作能很好地表示在以对象为基础的用户界面上.一个片、一条曲线或一个点可以由点到点及移动鼠标进行选择.成功的选择一个对象就是所谓“激活”(active).被激活的对象显示以高亮度.例如,一个片被激活就画成填充某种颜色,一条曲线被激活就显示以粗实线,而一个点被激活就代之以小实心圆.这些被激活的对象可以进行移动、分裂或消除操作.

3 实现

3.1 数据结构

曲线是由四个控制点 $[P_0, P_1, P_2, P_3]$ 的数组所表示的三次贝齐埃曲线,数组类型称为 Controlpt,片作为一个环用 Controlpt 的双向链表表示.显然,一个有效点是在一个片上每个 Controlpt 数组的第一点或最后一点,贝齐埃曲线用卡斯特利乌中点细分算法描述,令 P 是四个控制点的数组,那末 Function 用于画一条贝齐埃曲线,用伪代码描述如下

```

fun DrawBezier(P) =                                DrawBezier(q);
  if Colinear(P) then                                DrawBezier(r);
    DrawLine(P0, P3)                                end
  else Let val(q, r) = Split(P)                      end if
  in                                                  end

```

其中 Colinear 检查在直线接合点 P_0 和 P_3 上的点 P_1 和 P_2 所给定的容许偏差, Split 提供对应于中点细分算法的控制点的两个数组,计算采用定点算术运算.

3.2 移动一个点

一个点邻接着两条曲线,不妨称为左曲线和右曲线.只要对该点作如下更改即可:左曲线的最后一点和右曲线的第一点改成新位置;左曲线的第三点和右曲线的第二点重新计算,以保证在移动点处切矢量偏差不变.

3.3 移动一条曲线

显然,移动一条曲线应使得该曲线的第一点和最后一点保持不变.问题是如何将光标位置的变化映射到两个中间控制点的变化上,这有多种可能的选择规则.为了移动曲线,用户选择曲线上的一个位置.当光标在曲线上时,选择由按下鼠标左键完成.当鼠标左键连续按下并移动光标,曲线应连续或至少很接近通过当前光标位置(图 3).假定某时刻光标在 q_0 ,下一个采样

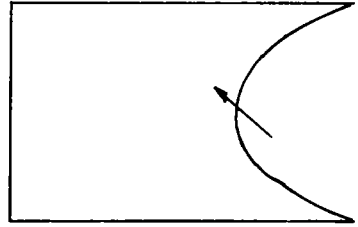


图 3 一条矩形边移动后变成一条曲线

时刻光标在 q_1 ,令 $t=T$ 在 Bezier 曲线上 q_0 处,显然在 $t=T$ 时该曲线的变化也应通过 q_1 .如果 $P_0(t)$ 和 $P_1(t)$ 作为移动前后的两条曲线的表示,并假定第二条曲线新的中间控制点 P'_1 和 P'_2 ,则有

$$P_0(T) = (1-T)^3 P_0 + 3(1-T)^2 T P_1 + 3(1-T) T^2 P_2 + T^3 P_3 = q_0,$$

$$P_1(T) = (1-T)^3 P_0 + 3(1-T)^2 T P'_1 + 3(1-T) T^2 P'_2 + T^3 P_3 = q_1.$$

由此有

$$P_1(T) - P_0(T) = 3(1-T)^2T(P_1' - P_1) + 3(1-T)T^2(P_2' - P_2) = q_1 - q_0.$$

这里必须知道 P_1' 和 P_2' 两点的 x, y 坐标, 以及在 x 和 y 上的两个等式, 因而有四个未知数和两度自由变量. 可采用如下简单规则来给未知数提供唯一的解: 如果 $T < 1/3$, 则只有左切线变化(即只有 P_1 移动); 同样地, 如果 $T > 2/3$, 则只有右切线变化(即只有 P_2 移动); 否则 P_1 和 P_2 都移动, 但它们的变化相同. 这样, 下面简单程序段即可求解

$$\begin{aligned} &\text{if } T < 1/3 \text{ then } P_1' - P_1 = (q_1 - q_0)/(3T(1-T)^2) \\ &\quad \text{else if } T > 2/3 \text{ then } P_2' - P_2 = (q_1 - q_0)/(3T^2(1-T)) \\ &\quad \text{else } P_2' - P_2 = P_1' - P_1 = (q_1 - q_0)/(3T(1-T)) \end{aligned}$$

这个方法依赖于 T 值的确定. 下面我们来说明如何计算对应于一个点的 t 值. 对于曲线上一个给定点 P , 由 Bezzer 曲线的参数表示有 $P(t) = P$, Newton-Raphson 可应用于关于 t 的两个三次多项式中(对于每个 x 和 y), 但如果点 P 不在曲线上, 迭代的结果不会收敛, 而且基于合成的已有代数方法将产生无序的结果. 事实上, 对于点 P “近似地”在曲线上, 卡斯特利乌算法能够解决 t 值的确定问题. 该算法生成一个控制图形的序列, 这个序列可看成曲线真实形状的近似闭合器, 对于每个图形其端点 t 值是已知的, 分裂过程中每个阶段的 t 值, 根据中点细分算法知道, 它等于前一阶段控制图形的每个端点 t 值的一半. 当近似于一直线的个别的控制图形被求到, 该近似直线相对于点 P 进行检查, 这只要给出点 P 的一个小的平方根近似值, 并相对于此平方根裁下这条线. 如果线相对于包含点 P 的裁剪盒是可见的, 那末 t 值对应于裁剪盒范围内的直线部分的中点可作为整条直线对于点 P 的近似 t 值. 当然, 如果分裂算法没有这样一个成功的裁剪, 则说明点 P 不接近于给定的曲线.

3.4 分裂一个点

分裂一个点要求在一个单一点的地方插入一条新的曲线, 如何实现现有多种可能的选择, 一个较好的方法是保持与该点邻接的两条曲线近似相同. 前一条曲线的点 P_3 和后一条曲线的点 P_0 . 分别沿着它们各自的切矢量方向移动到的其相关的内部控制点距离的一半, 即 P_3 的移动量为 $0.5 * (P_2 - P_3)$, P_0 的移动量为 $0.5 * (P_1 - P_0)$. 这两个位置是新曲线的首点和末点位置. 确定新曲线的形状的最简单的选择是作为直线, 因为用户总是能够根据需要将其弯曲.

3.5 分裂一条曲线

分裂一条曲线要沿该曲线插入一个新的点, 这样用两条新曲线代替原来一条曲线, 并使两条新曲线再现原来单一曲线的形状. 这是容易实现的. 当用户在曲线上选择一个点, 其对应的 t 值可以计算出来, 此时 $P(t)$ 等同于那个点, 应用卡斯特利乌算法并以 t 值作为分裂值来分裂曲线.

3.6 分裂一个片

分裂一个片的结果是要使两个新片合起来组成原来的片. 分裂总是沿直线进行, 用户选择一个片作为整片, 然后从菜单上选择一个分裂选项, 用户还必须选择两条曲线, 并在它们中各选择一个点(或执行一个空选择以便终止交互作用). 当鼠标选择第一条曲线后按下左键, 接着根据分裂的反馈信息选择第二条曲线, 两条曲线上的点选择后 t 值就被计算出来, 并在这两点的连线上将片一分为二. 片被分裂后要为两个新片构造新的数据结构, 并执行必要的内务处理和重画.

3.7 消除一个点

在两条邻接曲线上删除一个点的结果是代之以单一的曲线,这种操作与“分裂一条曲线”在一定程度上相反.但是,如果一条曲线被分裂后,在新曲线没有任何改变之前,紧接着将它们的邻接点删去,那末对片来说没有变化.不过,不能简单地将“分裂一条曲线”理解为与“删去一个点”相反,这是因为点一旦被删去就没有足够的信息计算它以前的位置.然而应用卡斯特利乌算法分裂一条曲线,却能够从分裂曲线的有效信息中计算出原来的曲线.值得注意的是,两条曲线的邻接点被删去后,并非构成原来单一的 Bezier 曲线.

3.8 删除一条曲线

这个操作与分裂一个点在某种意义上相反,如果一个点被分裂,并且所产生的新曲线又被直接删去,那末对片来说没有变化.采用的算法与分裂一条曲线相反.

3.9 移动和删除一个片

通过变换矢量修改一个片上所有的点可实现片的移动.删除一个片则要求从数据结构中删去,该数据结构要保持显示当前片链表,光标必须重画从显示该片的移动偏量.

4 结束语

本文是一种基于修改一个整片的自顶向下的交互绘图方法,这种提法是相对于应用以逼近为基础的增量工具而言^[3].这里讨论的仅以分段三次 Bezier 曲线为基础,方法的实现并不困难,但它容易推广到更复杂的表示法.由于建立了一套基础方法(移动、分裂、消除),可适用于不同的对象(片、曲线和有效点),因而能很自然地表示原始对象.虽然目前的实现只是实验性的,并且方法依赖于描绘和卡斯特利乌算法的交互选择.但随着表示法的多样性以及从二维扩展到三维,方法的潜力和实用性勿容置疑.

参 考 文 献

- 1 Slater M. A top down method for interactive drawing. Computer Graphics Forum, 1988, (7):323~329
- 2 王东泉. 计算机图形学与 CAD 技术. 上海:上海交通大学出版社,1992,156~196
- 3 张全伙,曾晓帆,范慧琳等. 任意两个多边形的求交算法. 华侨大学学报(自然科学版),1995,16(1):111~115

A Top-Down Design of Garment Pattern Pieces

Zhang Quanhua Fan Huilin

(Dept. of Computer Science, Huaqiao Univ., 362011, Quanzhou)

Abstract A grading and styling design of garment pattern pieces is described in a top-down way. By mouse operating, the pattern pieces are constructed from stretching and cutting the existing graphics. The implementation of the method is based on a ring of cubic Bezier curves and de Casteljau algorithm.

Keywords Bezier curves, de Casteljau algorithm, top-down, interactive drawing, pattern pieces