

# 三次 Bezier 曲线绘制的一种新的快速算法

郑文明 吴清江

(华侨大学信息科学与工程学院, 泉州 362011)

**摘要** 提出一种基于3次 Bezier 曲线细分算法思想. 利用细分过程中细分前的控制多边形, 同细分后的控制多边形之间的面积大小, 以决定是否再进行下一步的细分. 在具体的算法中, 还考虑控制多边形自身的几何特性来减少判断的次数. 该方法可大大提高三次 Bezier 曲线的生成速度.

**关键词** Bezier 曲线, 控制多边形细分, 矢量的点积

**中图分类号** TP 391. 41

**文献标识码** A

Bezier 曲线是自由曲线、曲面中, 最重要和最基本的方法之一. 对于 Bezier 曲线的绘制, 一般采用细分算法<sup>[1~5]</sup>. 在算法中是否再对控制多边形作进一步细分的判别, 是根据各控制顶点到底边的距离来进行<sup>[6]</sup>. 采用这种方法, 每次均须计算各中间顶点到控制多边形两端点连线的距离. 三次 Bezier 曲线每次的判断至少要计算两次向量的叉乘. 本文提出一种利用控制多边形自身特点, 使三次 Bezier 曲线、曲面的逼近速度大大提高.

## 1 Bezier 曲线的定义及性质

### 1.1 Bezier 曲线的定义

对给定空间  $n+1$  个控制点位置,  $p_k = (x_k, y_k, z_k)$ ,  $k = 0, 1, \dots, n$ . Bezier 曲线定义式为  $P(t) = \sum_{k=0}^n P_k B_{k,n}(t)$ ,  $0 \leq t \leq 1$ , 其中  $B_{k,n}(t)$  是 Nernstein 多项式,  $B_{k,n}(t) = C(n, k) t^k (1-t)^{n-k}$ ,  $C(n, k)$  是二项式系数  $C(n, k) = \frac{n!}{k!(n-k)!}$ .

### 1.2 Bezier 曲线的包凸性

Bezier 曲线  $P(t)$  位于其控制顶点  $p_0, p_1, \dots, p_n$  的凸包之内, 如图1所示. 这是由 Bernstein 基函数的权值性质确定的. 由于  $B_{i,n}(t) \geq 0$ ,  $\sum_{i=0}^n B_{i,n}(t) = 1$ ,  $t \in [0, 1]$ , 可见  $P(t)$  是  $p_0, p_1, \dots, p_n$  各点的凸线性组合. 因此, 曲线  $P(t)$  位于  $n+1$  个顶点  $p_0, p_1, \dots, p_n$  的凸包内.

### 1.3 Bezier 曲线的细分法

Bezier 曲线  $P(t)$ , 经中点分割而得到的两条段线分别为  $Q(t) \left[ = P(t), t \in \left[0, \frac{1}{2}\right] \right]$  和  $R(t) \left[ = P(t), t \in \left[\frac{1}{2}, 1\right] \right]$ . 于是, 分割可表达为  $P(t) = \sum_{i=0}^n P_i B_{i,n}(t) = \sum_{i=0}^n p_i^{[1]} B_{i,n}(2t), t \in \left[0, \frac{1}{2}\right]$  或

$= \sum_{i=0}^n p_i^{[n-i]} B_{i,n}(2t-1), t \in [\frac{1}{2}, 1], p_i^{[j]}$  的递推关系式, 可表示为式(1) 式所示, 其中  $p_{i+1}^{[k+1]} = \frac{1}{2}(p_i^{[k]} + p_{i+1}^{[k]})$ . 即

$$\left. \begin{matrix} p^0, \\ p^1 \quad p^{[1]}, \\ p^2 \quad p^{[2]} \quad p^{[2]}, \\ \vdots \\ p^n \quad p_n^{[1]} \quad p_n^{[2]} \quad \dots \quad p_n^{[n]}. \end{matrix} \right\} \quad (1)$$

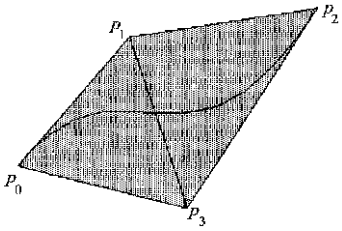


图1 曲线位于控制顶点内

容易证明  $p_n^{[n]}$  在曲线  $P(t)$  上. 式(1) 呈直角三角形的三条边方向都有意义. 其中垂直的直角边上的点  $p_0, p_1, \dots, p_n$  是  $(P(t), t \in [0, 1])$  的控制顶点, 斜边上的点  $p_0, p^{[1]}, p^{[2]}, \dots, p_n^{[n]}$  是  $Q(t) (t \in [0, \frac{1}{2}])$  的控制顶点, 水平的直角边上的点  $p_n, p_n^{[1]}, p_n^{[2]}, \dots, p_n^{[n]}$  是  $R(t) (t \in [\frac{1}{2}, 1])$  的控制顶点.

2 细分算法生成三次 Bezier 曲线的新方法的提出

下面来说明本文所采用的方法来绘制生成三次 Bezier 曲线. 根据细分算法思想, Bezier 曲线的细分过程可表示为图2所示, 其中各顶点的关系满足式(1) 的递推关系式. 本文的基本思路是当图2中阴影部分面积足够小时, 新围成的多边形可用作 Bezier 曲线段的逼近, 而不再对新的控制多边形进行细分. 设图2中阴影部分面积大小为  $S$ . 为了说明  $S$  计算, 我们把阴影部分面积表示为如下小三角面积之和(图3), 即  $S = S_0 + S_1 + S_1^1$ . 此外如图4所示, 我们注意到当控制

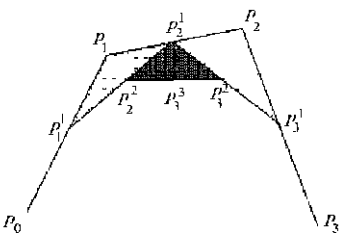


图2 面积示意图

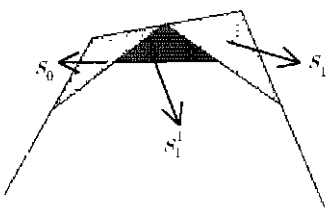


图3 面积的分割

多边形中相邻两边的夹角为锐角时, 控制多边形与 Bezier 曲线外形有较大的差距. 此时控制多边形不宜作为 Bezier 曲线段的逼近, 因而可直接进行下一步细分.



图4 控制多边形中相邻两边的夹角为锐角情形

因此, 当我们检查到控制多边形相邻两边的夹角中存在锐角时, 即可直接进行下一步细分. 而对控制多边形相邻两边夹角大小的计算可用两向量的点积来表示(图5). 即

$$\cos \theta = - \frac{p_i p_{i+1} \cdot p_{i+1} p_{i+2}}{p_i p_{i+1} \cdot p_{i+1} p_{i+2}}$$

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://www

因此,有以下3种情况.(1) 当  $p_i p_{i+1} p_{i+1} p_{i+2} > 0$  时,  $\cos \theta < 0$ ,  $\theta > \frac{\pi}{2}$ , 此时可依面积条件判断是否再进行细分.(2) 当  $p_i p_{i+1} p_{i+1} p_{i+2} < 0$  时,  $\cos \theta > 0$ ,  $\theta < \frac{\pi}{2}$ , 此时可直接再进行细分.(3) 当  $p_i p_{i+1} p_{i+1} p_{i+2} = 0$  时,  $\cos \theta = 0$ ,  $\theta = \frac{\pi}{2}$ , 此时可直接再进行细分. 这样仅有情况(1)需要再进行

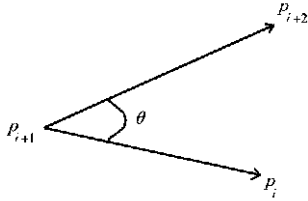


图5 两向量夹角示意图

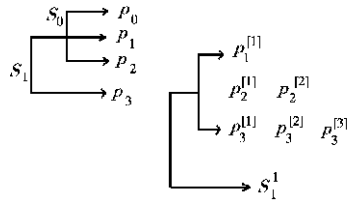


图6 面积计算示意图

下一步的计算. 下面, 我们分析一下如何简化面积  $S$  的计算(图6). 由图2可知

$$S_0 = S_{\Delta p_1^{[1]} p_2^{[1]} p_1} = \frac{1}{4} S_{\Delta p_0 p_2 p_1} = \frac{1}{8} p_0 p_1 \times p_1 p_2,$$

$$S_1 = S_{\Delta p_2^{[1]} p_3^{[1]} p_2} = \frac{1}{4} S_{\Delta p_1 p_2 p_3} = \frac{1}{8} p_1 p_2 \times p_2 p_3,$$

$$S_1^1 = S_{\Delta p_2^{[2]} p_3^{[1]} p_3^{[2]}} = \frac{1}{4} S_{\Delta p_1^{[1]} p_2^{[1]} p_3^{[1]}} = \frac{1}{8} p_1^{[1]} p_2^{[1]} \times p_2^{[1]} p_3^{[1]} =$$

$$\frac{1}{8} \frac{1}{4} p_0 p_2 \times p_1 p_3 = \frac{1}{32} p_0 p_2 \times p_1 p_3.$$

事实上,  $S_0, S_1, S_1^1$  可直接从图6的关系式中得出  $S_0 = \frac{1}{4} S_{\Delta p_0 p_2 p_1}$ ,  $S_1 = \frac{1}{4} S_{\Delta p_1 p_2 p_3}$ ,  $S_1^1 =$

$\frac{1}{4} S_{\Delta p_1^{[1]} p_2^{[1]} p_3^{[1]}}$ . 如图7所示, 我们假设第  $k$  次细分得到的控制顶点为  $p_{0,k}, p_{1,k}, p_{2,k}, p_{3,k}$ , 对应的面积分别记为  $S_{0,k}, S_{1,k}, S_{1,k}^1$ . 下一次细分得到的控制顶点记为  $q_{0,k+1}, q_{1,k+1}, q_{2,k+1}, q_{3,k+1}$  和  $r_{0,k+1}, r_{1,k+1}, r_{2,k+1}, r_{3,k+1}$ . 因此, 有如下关系式为  $p_{0,k} = q_{0,k+1}$ ,  $p_{1,k}^1 = q_{1,k+1}$ ,  $p_{2,k}^2 = q_{2,k+1}$ ,  $p_{3,k}^3 = q_{3,k+1}$ ,  $p_{3,k}^3 = r_{0,k+1}$ ,  $p_{3,k}^{[2]} = r_{1,k+1}$ ,  $p_{3,k}^1 = r_{2,k+1}$ ,  $p_{3,k} = r_{3,k+1}$ . 下面任取一段曲线(不防以  $Q$  段为例)进行分析. 由几何关系得

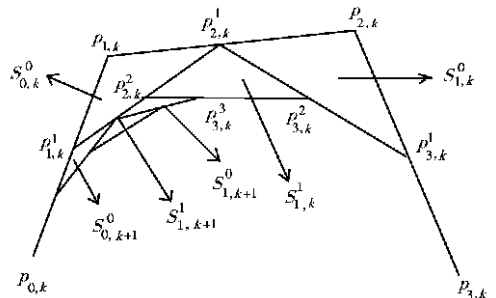


图7 面积的简化计算示意图

$$S_{0,k+1}^0 = \frac{1}{4} S_{\Delta p_{0,k} p_{1,k}^1 p_{2,k}^2} = \frac{1}{8} p_{0,k} p_{1,k}^1 \times p_{1,k}^1 p_{2,k}^2 =$$

$$\frac{1}{32} p_{0,k} p_{1,k}^1 \times p_{1,k}^1 p_{2,k}^1 = \frac{1}{64} p_{0,k} p_{1,k}^1 \times p_{0,k} p_{2,k}^2,$$

$$S_{0,k}^0 = S_{\Delta p_{1,k}^1 p_{2,k}^1 p_{2,k}^2} = \frac{1}{8} p_{0,k} p_{1,k}^1 \times p_{0,k} p_{2,k}^2,$$

$$S_{1,k+1}^0 = \frac{1}{4} S_{\Delta p_{1,k}^1 p_{2,k}^2 p_{3,k}^3} = \frac{1}{8} p_{1,k}^1 p_{2,k}^2 \times p_{2,k}^2 p_{3,k}^3 =$$

$$\frac{1}{32} p_{1,k}^1 p_{2,k}^1 \times p_{2,k}^2 p_{3,k}^2 = \frac{1}{64} p_{1,k}^1 p_{2,k}^1 \times p_{1,k}^1 p_{3,k}^1,$$

$$S_{1,k}^1 = S_{\Delta p_{2,k}^2 p_{2,k}^1 p_{3,k}^2} = \frac{1}{8} p_{1,k}^1 p_{2,k}^1 \times p_{1,k}^1 p_{3,k}^1,$$

$$S_{1,k+1}^0 = \frac{1}{4} S_{\Delta p_{1,k}^1 p_{2,k}^2 p_{3,k}^3} = \frac{1}{8} S_{\Delta p_{1,k}^1 p_{2,k}^2 p_{3,k}^3} = \frac{1}{8} S_{1,k}^1,$$

$$S_{1,k+1}^1 = \frac{1}{32} p_{0,k}^0 p_{2,k}^2 \times p_{1,k}^1 p_{3,k}^3.$$

$$\text{所以, } S_{k+1} = S_{0,k+1}^0 + S_{1,k+1}^0 + S_{1,k+1}^1 = \frac{1}{8} (S_{0,k}^0 + S_{1,k}^1) + \frac{1}{32} p_{0,k}^0 p_{2,k}^2 \times p_{1,k}^1 p_{3,k}^3.$$

### 3 讨论

通过以上的推导看出,我们在计算第  $k+1$  次的面积时,可以充分利用第  $k$  次已计算过的面积.于是,每次求面积时所花的时间主要用在求一次矢量的叉乘上.相比之下,传统的方法是采用计算中间两顶点到两端点连线的距离,对于点到直线的距离,要计算一次的矢量叉乘.因此,对于三次 Bezier 曲线,就必须计算两次的矢量叉乘.在本算法中,只要计算一次矢量的叉乘即可,其计算量将近减少一半.试验结果表明,用本文所采用的方法来生成 Bezier 曲线、曲面,其绘制速度确实比传统方法来得快,且其光滑性并不亚于传统法.

### 参 考 文 献

- 1 马利庄,王荣良.计算机辅助造型技术及其应用[M].北京:科学出版社,1996.1~100
- 2 孙家广,杨长贵.计算机图形学[M].北京:清华大学出版社,1999.301~308
- 3 罗振东,廖光裕.计算机图形学[M].上海:复旦大学出版社,1993.114~128
- 4 金廷赞.计算机图形学[M].杭州:浙江大学出版社,1988.152~159
- 5 Heam D, Pauline Baleer M P 著.计算机图形学[M].蔡士杰等译.北京:电子工业出版社,1998.245~251

## A New Fast Algorithm for Drawing Cubic Bezier Curve

Zheng Wenming      Wu Qingjiang

(College of Info. Sci. & Eng., Huaqiao Univ., 362011, Quanzhou)

**Abstract** An idea of subdivision algorithm based on cubic Bezier curve is presented. During the process of subdivision, whether or not to go on further subdivison can be decided by making use of the area between control polygon before subdivision and that after subdivision. In specific algorithm, the characteristic of control polygon itself has to be considered so as to reduce the frequency of judgement and to accelerate greatly the generation of cubic Bezier curve.

**Keywords** Bezier curve, subdivision of control polygon, dot product of vector