

一种参数多项式曲面片的逐点生成算法

黄有度 朱功勤

(合肥工业大学数学与信息科学系, 合肥 230009)

摘要 在计算机绘图中, 一般来说, 曲线实际上是由折线代替, 而曲面实为小平面拼接而成. 在使计算量降到最低的情况下画出真正的曲线方面, 已有许多文章研究了曲线的逐点生成方法, 并取得了一定的进展, 但是尚无有效的快速逐点生成曲面的方法. 为了快速逐点生成曲面, 在建立多项式函数递推计算公式和算法的基础上, 给出了一种逐点生成参数多项式曲面片的算法, 由于此算法中只用到整数加法运算, 且点数的适当选取可使计算量达到极小, 因此是一种很有效的算法. 该方法还可加以改进, 而用于有理函数, 这无疑对有理曲线曲面(如 NURBS 曲线曲面)的快速生成以及对计算机图形的其他一些领域都是有意义的.

关键词 参数多项式曲面 快速逐点生成 整数加法运算 Bernstein 基函数

中图分类号: TP391.41 **文献标识码:** A **文章编号:** 1006-8961(2002)07-0663-04

A Point-by-Point Generating Algorithm for Parametric Polynomial Surface

HUANG You-du, ZHU Gong-qin

(Dept. of Mathematics & Information Science, Hefei University of Technology, Hefei 230009)

Abstract In computer drawing, generally, curves are represented by line segments and surfaces are tessellated up with small plane patches. To draw genuine curve and reduce computational costs as much as possible, curve-generating point-by-point has been studied in many papers, some good results have been achieved. But there is lack of effective approaches for fast surface-generating. If we suppose that surfaces consist of points of space lattice and the coordinates of points on a surface could be calculated one by one, then we would obtain the surface. There are already some algorithms for calculating polynomial curves on a plane, we can generalized them to space curves and surfaces. This paper presents an algorithm based on recursive formulas and algorithms for polynomial to generate a parametric polynomial surface patch point-by-point, in which only integral additive operation is employed, and properly choosing point number minimizes the computational costs, so it is quite efficient. The method in the algorithm can be modified for rational function, therefore it is of significance in fast generating parametric rational curve and surface (such as NURBS curve and surface), as well as in many other areas of Computer Graphics.

Keywords Parametric polynomial surface, Fast generating point-by-point, Integral additive operation, Bernstein bases

0 引言

曲线的逐点生成算法已有较多的研究^[1-5], 但有关曲面逐点生成算法的研究尚未见到, 许多绘图

软件所绘曲面皆为小平面拼接而成, 无论是标准几何体的表面(球面、锥面等)还是 Bézier 曲面都是如此^[6]. 本文给出了空间参数多项式曲面片的逐点生成算法, 计算中只用到整数加法, 因此计算量很小. 实验证明, 那些广泛应用的 Bézier 曲面片和 Coons

基金项目: 国家自然科学基金(19671002)

收稿日期: 2001-05-08; 改回日期: 2001-09-28

曲面片等都可用它来生成。

与平面上曲线逐点生成算法所采用的方法类似,这里也假设曲面是由空间三维点阵中的点组成,且每一坐标面上点阵的规模与像素一致。用该方法逐点求出曲面上每一点的坐标,即得到整个曲面片。

本文方法的主要思想是把曲面片看作由曲线拼成,通过逐条生成曲线,即得到曲面。由于构成多项式曲面片的曲线是多项式曲线,因此可用文献[5]的方法来生成。

1 多项式曲线的逐点生成算法简述^[5]

设曲线的参数多项式方程为 $x = x(t)$, $y = y(t)$, $0 \leq t \leq 1$, 为计算方便,且不影响精度,可设其系数为有理数,那么依次求出 $t = \frac{i}{n}$ ($0 \leq i \leq n$) 的 $n+1$ 个整数坐标点,即构成曲线。为使曲线连续, n 不能小于 $\max_{0 \leq t \leq 1} |x'(t)|$ 和 $\max_{0 \leq t \leq 1} |y'(t)|$ 。但由于 n 越大,计算量也越大,因此,选取 $n = \max\{\max_{0 \leq t \leq 1} |x'(t)|, \max_{0 \leq t \leq 1} |y'(t)|\}$, 可使计算量达到极小。

由于像素坐标必须为整数,因此 x 由 $x = x(t)$ 化为如下的整数方程

$$Nx_i = \varphi(i) + z_i \quad (1)$$

所确定,其中,常数 N 为一个正整数, x_i 和 z_i 为由 i 决定的整数,且 $|z_i| \leq \frac{N}{2}$, $\varphi(i)$ 为 i 的整系数多项式。设式(1)中各数已确定,把 $x_i = x_{i+1} - \Delta x_i$, $\varphi(i) = \varphi(i+1) - \Delta\varphi(i)$ 代入式(1),整理得

$$Nx_{i+1} = \varphi(i+1) - \Delta\varphi(i) + z_i + N\Delta x_i$$

由此式可按下面各种情况确定 x_{i+1} 和 z_{i+1} :

① 若 $-\Delta\varphi(i) + z_i \leq -\frac{N}{2}$, 则 $\Delta x_i = 1$ (即 $x_{i+1} = x_i + 1$), $z_{i+1} = -\Delta\varphi(i) + z_i + N$;

② 若 $-\Delta\varphi(i) + z_i > \frac{N}{2}$, 则 $\Delta x_i = -1$ (即 $x_{i+1} = x_i - 1$), $z_{i+1} = -\Delta\varphi(i) + z_i - N$;

③ 若 $-\frac{N}{2} < -\Delta\varphi(i) + z_i \leq \frac{N}{2}$, 则 $\Delta x_i = 0$ (即 $x_{i+1} = x_i$), $z_{i+1} = -\Delta\varphi(i) + z_i$ 。

由选取的 n 和 z_i ($n \geq \max_{0 \leq t \leq 1} |x'(t)|$, $-\frac{N}{2} < z_i \leq \frac{N}{2}$), 可保证 $-\frac{3N}{2} < -\Delta\varphi(i) + z_i \leq \frac{3N}{2}$,

从而 $-\frac{N}{2} < z_{i+1} \leq \frac{N}{2}$ 。

用上述方法可逐次得到 $x_i, i=1, 2, \dots, n$, 所用到的差分 $\Delta\varphi(i)$ 可由下面公式得出

$$\Delta^h \varphi(i) = \Delta^h \varphi(i-1) + \Delta^{h+1} \varphi(i-1)$$

由于 h 次多项式的 h 阶差分恒为常数,因此在求出 $i=0$ 时的各阶差分后,以后的各阶差分都可根据上式用一次加法得到。

2 参数多项式曲面片的逐点算法

设曲面片的参数方程为: $P = P(r, s)$, $0 \leq r, s \leq 1$, 即 $x = x(r, s)$, $y = y(r, s)$, $z = z(r, s)$, 其中 $x(r, s)$, $y(r, s)$, $z(r, s)$ 为 r, s 的多元多项式,并设其

系数为有理数。取 $r = \frac{i}{n}$, $s = \frac{j}{n}$, $0 \leq i, j \leq n$, 则

$$x(r, s) = x\left(\frac{i}{n}, \frac{j}{n}\right) = \frac{f(i, j)}{N} \quad (2)$$

其中, $f(i, j)$ 是 i, j 的整系数多项式, N 是一个正整数。当 $n \geq \max_{0 \leq r, s \leq 1} \left\{ \left| \frac{\partial x}{\partial r} \right|, \left| \frac{\partial x}{\partial s} \right| \right\}$ 时,由式(2)易证明,当 i 或 j 增加 1 时, x 增量的绝对值不大于 1, $f(i, j)$ 增量的绝对值不大于 N 。由于 x 取整数值,且曲面与曲线的逐点计算类似,因此可把式(2)改写为

$$Nx_{i,j} = f(i, j) + e_{i,j} \quad (3)$$

其中, $x_{i,j}$ 和 $e_{i,j}$ 为整数,且 $|e_{i,j}| \leq \frac{N}{2}$ 。对 $j=0$, 依次取 $i=0, 1, \dots, n$, 可按文献[5]中多项式曲线逐点算法由式(3)求得各 $x_{i,0}$ 和 $e_{i,0}$ 。对每个固定的 $x_{i,0}$ 和 $e_{i,0}$, 再依次取 $j=0, 1, \dots, n$, 即可求得各 $x_{i,j}$ 和 $e_{i,j}$ 。

对 y 和 z 也可作同样的处理。这样所得各点 $P_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$, $i, j=0, \dots, n$, 即组成所求的曲面片。为保证曲面的连续性, n 应取为 x, y, z 关于 r, s 的偏导数的绝对值的共同上界。

由于许多绘图软件(如 OpenGL)具有深度检测功能,可自行解决消隐问题,因此本算法中不再对消隐问题加以考虑。

3 逐点计算时,点数 n 的确定

由上节算法的推导知,参数 r, s 方向上的点数 n 应满足

$$n \geq \max_{0 \leq r, s \leq 1} \left\{ \left| \frac{\partial x}{\partial r} \right|, \left| \frac{\partial x}{\partial s} \right|, \left| \frac{\partial y}{\partial r} \right|, \left| \frac{\partial y}{\partial s} \right|, \left| \frac{\partial z}{\partial r} \right|, \left| \frac{\partial z}{\partial s} \right| \right\}$$

因此应求出 x, y, z 关于 r, s 的偏导数绝对值的上界。下面分几种情况讨论:

(1) m 次三角形 Bézier 曲面片计算公式为

$$P = \sum_{i=0}^m \sum_{j=0}^{m-i} P_{i,j} B_{i,j}^m(r,s) \quad 0 \leq r,s,r+s \leq 1 \quad (4)$$

其中, $P_{i,j}$ 为控制点, $B_{i,j}^m(r,s)$ 为二元 Bernstein 基函数

$$B_{i,j}^m(r,s) = \begin{cases} \frac{m!}{i!j!(m-i-j)!} r^i s^j (1-r-s)^{m-i-j} & 0 \leq i,j,i+j \leq m \\ 0 & \text{其他} \end{cases} \quad (5)$$

$B_{i,j}^m(r,s)$ 满足

$$\left. \begin{aligned} \frac{\partial B_{i,j}^m(r,s)}{\partial r} &= m[B_{i-1,j}^{m-1}(r,s) - B_{i,j}^{m-1}(r,s)] \\ \frac{\partial B_{i,j}^m(r,s)}{\partial s} &= m[B_{i,j-1}^{m-1}(r,s) - B_{i,j}^{m-1}(r,s)] \end{aligned} \right\} \quad (6)$$

这样即可得

$$\left. \begin{aligned} \frac{\partial P}{\partial r} &= m \sum_{i=0}^{m-1} \sum_{j=0}^{m-1-i} (P_{i+1,j} - P_{i,j}) B_{i,j}^{m-1}(r,s) \\ \frac{\partial P}{\partial s} &= m \sum_{i=0}^{m-1} \sum_{j=0}^{m-1-i} (P_{i,j+1} - P_{i,j}) B_{i,j}^{m-1}(r,s) \end{aligned} \right\} \quad (7)$$

由于 $\sum_{i+j \leq m} B_{i,j}^m(r,s) \equiv 1$, 所以

$$\left| \frac{\partial P}{\partial r} \right| \leq m \cdot \max_{i,j} |P_{i+1,j} - P_{i,j}|$$

$$\left| \frac{\partial P}{\partial s} \right| \leq m \cdot \max_{i,j} |P_{i,j+1} - P_{i,j}| \quad (8)$$

若以上两式右端的 P 分别换为 x, y, z , 则可求得 6 个值, 而不小于该 6 个值的整数, 即可作为 n 的值。

(2) 四边形 Bézier 曲面片可由如下二代数多项式表示

$$P = P(r,s), 0 \leq r,s \leq 1$$

设 $P(r,s)$ 对 r 是 m 次多项式, 对 s 是 h 次多项式, 则四边形 Bézier 曲面片方程可表示为

$$P = \sum_{i=0}^m \sum_{j=0}^h P_{i,j} B_{i,m}(r) B_{j,h}(s) \quad (9)$$

其中, $B_{i,m}(r)$ 和 $B_{j,h}(s)$ 为 Bernstein 基函数; 四边形 Bézier 曲面也可表示为

$$P = \sum_{i=0}^m \sum_{j=0}^h Q_{i,j} r^i s^j \quad (10)$$

若四边形 Bézier 曲面是后一种形式(式(10)), 则可通过下式(见文献[5]或文献[7])转化为式(9)

$$\begin{pmatrix} 1 \\ x \\ y \\ z \end{pmatrix} = \begin{bmatrix} C_0^m & C_1^m & \dots & C_m^m \\ 0 & C_{-1}^{m-1} & \dots & C_{-1}^0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & C_0^0 \end{bmatrix} \begin{pmatrix} \frac{1}{C_0^m} \\ \vdots \\ \frac{1}{C_1^m} \\ \vdots \\ \frac{1}{C_m^m} \end{pmatrix} \begin{pmatrix} B_{i,m}(r) \\ B_{j,h}(s) \\ \vdots \\ B_{i,m}(r) \end{pmatrix} \quad (11)$$

由 $B_{i,k}^k(x) = k [B_{i-1,k-1}(x) - B_{i,k-1}(x)]$ 和

$$\sum_{i=0}^k B_{i,k}^k(x) \equiv 1, \text{ 易得}$$

$$\left| \frac{\partial P}{\partial r} \right| \leq m \cdot \max_{i,j} |P_{i+1,j} - P_{i,j}|$$

$$\left| \frac{\partial P}{\partial s} \right| \leq h \cdot \max_{i,j} |P_{i,j+1} - P_{i,j}| \quad (12)$$

同样, 如以上两式右端的 P 分别换为 x, y, z , 则可求得 6 个值, 而不小于该 6 个值的整数, 即可作为 n 的值。

4 曲面绘制实例

两个三次三角形 Bézier 曲面片 P, Q 的各控制点分别为 $p_{i,k}$ 和 $q_{i,k}$ ($i, j, k \geq 0, i+j+k=3$), 各控制点坐标数据如下(为使两曲面光滑连接, P 附近的控制点 q 需适当选取, 以使两曲面相邻的控制三角形构成平行四边形):

- $p_{300} = (180, 20, 150),$
- $p_{210} = (100, 60, 100), p_{201} = (200, 180, 380),$
- $p_{120} = (100, 120, 160), p_{111} = (160, 140, 400),$
- $p_{102} = (300, 120, 100),$
- $p_{030} = (40, 200, 60), p_{021} = (200, 200, 360),$
- $p_{012} = (100, 220, 60), p_{003} = (300, 200, 300);$
- $q_{300} = p_{003},$
- $q_{210} = p_{012}, q_{201} = p_{012} + p_{003} - p_{002},$
- $q_{120} = p_{021}, q_{111} = p_{021} + p_{012} - p_{111},$
- $q_{102} = (240, 200, 200),$
- $q_{030} = p_{030}, q_{021} = p_{030} + p_{021} - p_{120},$
- $q_{012} = (50, 300, 220), q_{003} = (180, 360, 140).$

图 1 为用 VC++4.2 编制的本文算法程序所生成的曲面(已用简单的光照模型进行了处理), 曲面由 P, Q 两曲面片光滑拼接而成, P 在上, Q 在下。左边的是曲面在 XY 坐标面上的投影, 右边的是曲面在 YZ 坐标面上的投影。

5 结论

本文给出了参数多项式曲面的逐点生成算法, 此算法可逐点计算曲面上点的空间坐标, 由于只用到整数加法, 故计算速度很快。曲面生成中的消隐问题可通过适当选取节点的排列方向或利用深度缓冲区(Z 缓冲区)来解决。本算法还可在如下几方面进行改进或推广:

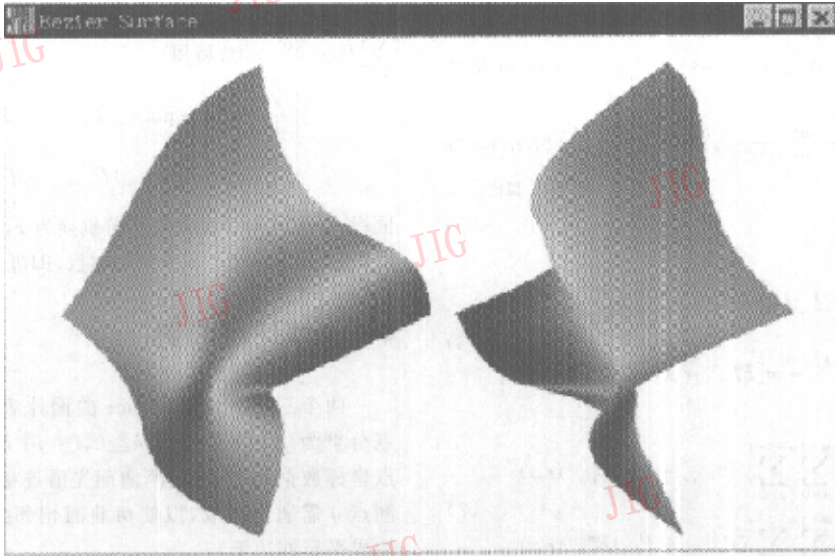


图1 用本文算法生成的曲面

(1)有理曲线、曲面的逐点生成算法;

(2)研究曲面光照模型的逐点生成计算等.由于光照模型中,涉及的三角函数可由有理函数表示,故以上两个问题实为同一个问题,即均为有理函数的逐点计算问题.另外,有理函数比多项式复杂得多,这将在其他文章中进行讨论;

(3)用离散值 i/n 来代替连续变量 t 时, n 的取值取决于曲线(面)函数导数绝对值的最大值,但在导数绝对值较小处,算出的点可能重叠,将造成计算量的浪费,但若分段选取 n 或固定 n 而使迭代步长可变,则又使算法大大复杂化,这些方法未必可取.因此如何进一步节省计算量也是值得深入探讨的问题.

参考文献

- 1 Bresenham J E. Algorithms for computer control of a digital plotter[J]. IBM Systems Journal, 1965,4(1):25~33.
- 2 Bresenham J E. A linear algorithm for incremental digital display of circular arcs[J]. Communications of the ACM, 1977,20(2): 100~106.
- 3 Huynh N P, Natlawit D. Efficient algorithm for Bézier curves [J]. Computer Aided Geometric Design, 2000, 17 (3): 247~250.

- 4 刘勇全,石教英.曲线的整数型生成算法[J].计算机学报,1998, 21(3):270~280.
- 5 黄有度,朱功勤.参数多项式曲线的快速逐点生成算法[J].计算机学报,2000,23(4):393~397.
- 6 贾志刚.精通OpenGL[M].北京:电子工业出版社,1998.
- 7 施法中.计算机辅助几何设计与非均匀有理B样条(CAGD&NURBS)[M].北京:北京航空航天大学出版社,1994.



黄有度 1949年生,硕士,现为合肥工业大学数学与信息科学系教授,研究方向为计算数学和计算机图形学.



朱功勤 1938年生,现为合肥工业大学数学与信息科学系教授兼中国科技大学数学系博士生导师,研究方向为计算数学.