

B 样条乘积计算的区间跳跃算法及应用

车翔玖¹, 高占恒¹, 车娜^{1,2}

CHE Xiang-jiu¹, GAO Zhan-heng¹, CHE Na^{1,2}

1. 吉林大学 计算机科学与技术学院, 长春 130012

2. 长春大学 艺术学院 计算机动画系, 长春 130024

1. College of Computer Science and Technology, Jilin University, Changchun 130012, China

2. Department of Computer Animation, College of Fine-Art, Changchun University, Changchun 130024, China

E-mail: gzh@jlu.edu.cn

CHE Xiang-jiu, GAO Zhan-heng, CHE Na. Knot interval skipping method for multiplication of B-spline functions and the application. Computer Engineering and Applications, 2009, 45(35): 157-160.

Abstract: By utilizing the relationship between the knot intervals, a method for calculating the transform matrix between different B-spline basis functions is presented. Furthermore, an interval skipping algorithm is given to calculate B-spline functions multiplication. This algorithm is so-called just as only part of knot intervals need to be considered to calculate rapidly B-spline functions multiplication. This algorithm partially solves the multiplication of piecewise polynomial and B-spline curve; and it can be applied into smooth B-spline surfaces blending and elevation of B-spline. Some examples are given to illustrate the efficiency of the presented algorithm.

Key words: B-spline multiplication; transform matrix; interval skipping method

摘要: 利用 B 样条基函数节点区间的对应关系, 首先给出了 B 样条基函数间的转换矩阵的计算方法, 进而给出了计算 B 样条乘积的区间跳跃算法。该算法仅需计算部分节点区间上的转换矩阵, 因此称其为区间跳跃算法。这一方法解决了分段多项式与 B 样条曲线乘积的计算问题, 可应用到 B 样条曲线的升阶、曲面间光滑拼接等问题中。通过算例验证了该方法计算简捷、易于实现。

关键词: B 样条乘积; 转换矩阵; 区间跳跃算法

DOI: 10.3778/j.issn.1002-8331.2009.35.048 **文章编号:** 1002-8331(2009)35-0157-04 **文献标识码:** A **中图分类号:** TP391

1 引言

在计算机辅助设计、逆向工程等诸多领域中, B 样条作为一种重要的曲线曲面造型方法得到广泛的应用。

而在 B 样条曲面造型, 具体地在 B 样条曲面间的光滑拼接中, 需要处理标量的拼接函数(分段多项式)与 B 样条切矢曲线的乘积问题^[1-3]。对该问题, 较早的是采用间接转化的方法。Farin 等^[4]和 Cohen 等^[5]所采用的间接方法即先通过节点插入将 B 样条转化为分段 Bézier, 然后进行 Bézier 的乘积计算和升阶, 再通过节点消去将分段 Bézier 转化为 B 样条。此过程繁杂、计算量大。因此, 学者们转向讨论 B 样条的升阶和不同 B 样条基函数间的转换矩阵来解决该问题。

针对一般形式节点向量的 B 样条曲线, 潘日晶^[6]给出了一种基于 Cohen 递推公式的快速升阶算法。孙景楠, 王仁宏^[7]改进了 Pigel 和 Tiller 关于 B 样条曲线升阶算法, 使之能够适用于端点插值型的均匀与非均匀 B 样条曲线的升阶问题。Wang G. Z., Deng C. Y.^[8]讨论了 B 样条曲线升阶与控制多边形割角之

的关系, 并指出 B 样条曲线进行升阶本质上就是对其控制多边形进行割角。Huang Q.X., Hu S.M. 等^[9]给出了 B 样条曲线一次升多阶的快速算法。算法是基于计算曲线的导数和对节点向量进行重采样, 从而计算出升阶后曲线的控制顶点。

利用 B 样条的基本运算, Pan, Weng^[10]得到了 B 样条基函数间的转换矩阵, 该方法可实现 B 样条曲线或曲面间的相互转化, 亦可用于 B 样条曲线的升阶。但 B 样条基函数间的转换矩阵是基于 B 样条基的整个支撑区间的, 因此, 转换矩阵的阶数较高。

另一代表性工作即 Chen, Riesenfeld 等^[11], 是基于 Blossoming 算法而提出的。该方法对于一般的 B 样条乘积计算具有通用性, 但算法中涉及许多搜索和判断过程, 计算量大, 不易于操作和实现。

针对 B 样条乘积的计算问题, 提出了一种直接、简捷的计算方法即矩阵方法。该方法能够快速实现分段多项式与 B 样条曲线的乘积计算及 B 样条曲线的升阶, 从而得到标量 B 样条与 B 样条曲线乘积的计算结果。

基金项目: 国家自然科学基金(the National Natural Science Foundation of China under Grant No.60773098, 60673021); 教育部博士学科点专项基金项目(the Doctoral Fund of Ministry of Education of China under Grant No.20060183041)。

作者简介: 车翔玖(1969-), 男, 博士, 教授, 中国计算机学会高级会员, 主要研究领域为计算机图形学, 计算机辅助几何设计; 高占恒(1979-), 男, 博士, 讲师, 中国计算机学会会员, 通讯作者, 主要研究领域为计算机辅助几何设计; 车娜(1982-), 女, 博士研究生, 讲师, 主要研究领域为计算机图形学。

收稿日期: 2009-10-15 **修回日期:** 2009-11-15

曲面间光滑拼接是 CAGD 中的基本问题,对于 B 样条曲面间光滑拼接问题的研究,使用的方法通常是用节点插入的技术将 B 样条曲面转化为分片 Bézier 曲面,利用 Bézier 曲面间已有的关于几何连续问题的结果进行研究^[12-15]。这类方法的缺点是难以直接得到曲面的 B 样条控制顶点之间的关系,并且节点插入和节点消去算法复杂。基于 B 样条基函数与其幂基表示之间的转换矩阵,研究了拼接函数是线性函数和二次函数的情况下邻接 B 样条曲面间的 G^1 拼接的问题。采用幂基表达式的优点是符号计算简单,可以直接得到 B 样条控制顶点之间的关系。

2 标量 B 样条函数乘积算法的总体思想

标量 B 样条函数与矢量 B 样条曲线乘积的问题可以归结为两个标量 B 样条函数乘积的问题。计算标量 B 样条函数乘积需要解决以下两个问题:乘积函数节点向量的确定和乘积函数控制系数的确定。现分别给出确定它们的方法。

设有 d 次标量 B 样条函数 $B(t)$ 和 \hat{d} 次标量 B 样条函数 $\hat{B}(t)$, 分别由节点向量 T, \hat{T} 和控制系数 b_i, \hat{b}_i 所定义。其中,

$$T = \{ \underbrace{t_0, \dots, t_0}_{q_0}, \underbrace{t_1, \dots, t_1}_{q_1}, \dots, \underbrace{t_s, \dots, t_s}_{q_s} \}$$

$$\hat{T} = \{ \underbrace{t_0, \dots, t_0}_{\hat{q}_0}, \underbrace{t_1, \dots, t_1}_{\hat{q}_1}, \dots, \underbrace{t_s, \dots, t_s}_{\hat{q}_s} \}$$

考虑端节点满重复度的情况,即 $t_0=0, t_s=1, q_0=q_s=d+1$ 。

2.1 乘积函数节点向量的确定

设 $A(t)=B(t)\hat{B}(t)$, 则 $A(t)$ 是 D 次 B 样条函数, $D=d+\hat{d}$ 。根据 $B(t)$ 和 $\hat{B}(t)$ 在每个节点处连续阶可以得到 $A(t)$ 的节点向量:

$$\bar{T} = \{ \underbrace{t_0, \dots, t_0}_{Q_0}, \underbrace{t_1, \dots, t_1}_{Q_1}, \dots, \underbrace{t_s, \dots, t_s}_{Q_s} \}$$

其中 $Q_0=Q_s=D+1, Q_i=D-\min\{d-q_i, \hat{d}-\hat{q}_i\}$ 。

2.2 乘积函数控制系数的确定

设 $A(t)$ 的控制系数是 a_i , 则在任意一个非退化节点区间 $[t_j, t_{j+1}]$ 上 ($j=0, 1, \dots, s-1$), $A(t), B(t)$ 和 $\hat{B}(t)$ 恰好分别有 $D+1, d+1, \hat{d}+1$ 个控制系数起作用, 它们分别是 $a_{R_j-D-1}, \dots, a_{R_j-1}, b_{r_j-d-1}, \dots, b_{r_j-1}, \hat{b}_{\hat{r}_j-\hat{d}-1}, \dots, \hat{b}_{\hat{r}_j-1}$ 。其中 $R_j = \sum_{i=0}^j Q_i, r_j = \sum_{i=0}^j q_i, \hat{r}_j = \sum_{i=0}^j \hat{q}_i$ 。

因此, $A(t)$ 的控制系数 $\{a_i\}_{R_j-D-1}^{R_j-1}$, 由 $B(t)$ 的控制系数 $\{b_i\}_{r_j-d-1}^{r_j-1}$ 和 $\hat{B}(t)$ 的控制系数 $\{\hat{b}_i\}_{\hat{r}_j-\hat{d}-1}^{\hat{r}_j-1}$ 所决定。

为了给出 b_j, \hat{b}_k, a_i 之间的关系, 首先需要给出在节点区间 $[t_j, t_{j+1}]$ 上非零的 B 样条基函数 $\{N_{i,d}(t)\}_{r_j-d-1}^{r_j-1}, \{\hat{N}_{i,\hat{d}}(t)\}_{\hat{r}_j-\hat{d}-1}^{\hat{r}_j-1}, \{\bar{N}_{i,D}(t)\}_{R_j-D-1}^{R_j-1}$ 与幂基之间的关系, 它们分别定义在节点向量 T, \hat{T}, \bar{T} 上。

设 $\{N_{i,d}(t)\}_{r_j-d-1}^{r_j-1}, \{\hat{N}_{i,\hat{d}}(t)\}_{\hat{r}_j-\hat{d}-1}^{\hat{r}_j-1}, \{\bar{N}_{i,D}(t)\}_{R_j-D-1}^{R_j-1}$ 与幂基之间的关系如下:

$$[N_{r_j-d-1,d}(t), \dots, N_{r_j-1,d}(t)] = [1, t, \dots, t^d] \mathbf{M}_J \quad (1)$$

$$[\hat{N}_{\hat{r}_j-\hat{d}-1,\hat{d}}(t), \dots, \hat{N}_{\hat{r}_j-1,\hat{d}}(t)] = [1, t, \dots, t^{\hat{d}}] \hat{\mathbf{M}}_J \quad (2)$$

$$[\bar{N}_{R_j-D-1,D}(t), \dots, \bar{N}_{R_j-1,D}(t)] = [1, t, \dots, t^D] \bar{\mathbf{M}}_J \quad (3)$$

其中, $\mathbf{M}_J, \hat{\mathbf{M}}_J, \bar{\mathbf{M}}_J$ 分别是 $d+1, \hat{d}+1, D+1$ 阶非奇异矩阵, 称它们分别为 $\mathbf{B}(t), \hat{\mathbf{B}}(t), \mathbf{A}(t)$ 在节点区间 $[t_j, t_{j+1}]$ 上与幂基之间的转换矩阵。

记

$$\begin{bmatrix} \bar{b}_{r_j-d-1} \\ \vdots \\ \bar{b}_{r_j-1} \end{bmatrix} = \mathbf{M}_J \begin{bmatrix} b_{r_j-d-1} \\ \vdots \\ b_{r_j-1} \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} \bar{\hat{b}}_{\hat{r}_j-\hat{d}-1} \\ \vdots \\ \bar{\hat{b}}_{\hat{r}_j-1} \end{bmatrix} = \hat{\mathbf{M}}_J \begin{bmatrix} \hat{b}_{\hat{r}_j-\hat{d}-1} \\ \vdots \\ \hat{b}_{\hat{r}_j-1} \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} \bar{a}_{R_j-D-1} \\ \vdots \\ \bar{a}_{R_j-1} \end{bmatrix} = \bar{\mathbf{M}}_J \begin{bmatrix} a_{R_j-D-1} \\ \vdots \\ a_{R_j-1} \end{bmatrix} \quad (6)$$

则由式(1),(2),(3)得

$$\bar{a}_i = \sum_{j+k=i} \bar{b}_j \bar{\hat{b}}_k, i=R_j-D-1, \dots, R_j-1$$

从而, 由(6)有

$$\begin{bmatrix} a_{R_j-D-1} \\ a_{R_j-D} \\ \vdots \\ a_{R_j-1} \end{bmatrix} = \bar{\mathbf{M}}_J^{-1} \begin{bmatrix} \bar{a}_{R_j-D-1} \\ \bar{a}_{R_j-D} \\ \vdots \\ \bar{a}_{R_j-1} \end{bmatrix} \quad (7)$$

3 转换矩阵的计算

B 样条基函数在本质上是分段的多项式函数, 因此, 只要计算出 B 样条基函数在每一个分段上的具体表达式就可以得到转换矩阵。借助符号计算软件 Maple 强大的计算能力给出计算转换矩阵的 Maple 程序。

首先, 给出能够计算任意次数、任意类型节点向量 B 样条基函数在每个节点区间上的表达式的 Maple 程序:

```
N := proc (i :: integer, k :: integer, j :: integer, T, u)
```

该程序实现的功能是: 计算 B 样条基函数 $N_{i,k}(u)$ 在节点区间 $[T_j, T_{j+1}]$ 上的表达式。其中, T 是节点向量。

程序 1 B 样条基函数 Maple 程序

```
restart;
N := proc (i :: integer, k :: integer, j :: integer, T, u) local N1, N2, N3,
m1, m2;
if k=0 then
if i=j then
1
else
0
end if
else
N1 := N(i, k-1, j, T, u);
N2 := N(i+1, k-1, j, T, u);
if Tik=Ti then
m1 := 0
else
m1 := (u-Ti)/(Tik-Ti)
end if;
if Tik+1=Ti+1 then
```

```

m2:=0
else
    m2:=(Tink+1-u)/(Tink+1-Tin1)
end if;
N3:=m1*N1+m2*N2
end if
end proc;

```

其次,在程序 1 的基础上,可以按如下步骤计算 $[T_j, T_{j+1}]$ 上的转换矩阵:

- (1) 计算 $N(i, k, j, T, u)$ 的解析表达式, 其中 $i=j-k, j-k+1, \dots, j$ 。
- (2) 将每一个 $N(i, k, j, T, u)$ 的系数按 u 的升幂顺序排成一列。
- (3) 将所有 $N(i, k, j, T, u)$ 的系数列按 i 从小到大的顺序组成矩阵, 此矩阵即是转换矩阵。

4 区间跳跃算法及其在曲面拼接中的应用

由前面叙述, 有如下在节点区间 $[t_j, t_{j+1}]$ 上计算 $\{a_i\}_{R_j-D-1}^{R_j-1}$ 的算法。

算法 1 在节点区间 $[t_j, t_{j+1}]$ 上计算乘积函数的控制系数

Input:

$$\{b_j\}_{r_j-d-1}^{r_j-1}, \{\hat{b}_k\}_{\hat{r}_j-d-1}^{\hat{r}_j-1}, T, \hat{T}, \bar{T}$$

Output: $\{a_i\}_{R_j-D-1}^{R_j-1}$

Begin:

(1) 根据 T, \hat{T}, \bar{T} 计算转换矩阵 $M_j, \hat{M}_j, \bar{M}_j$, 根据式(4), (5)

计算 $\{\bar{b}\}_{r_j-d-1}^{r_j-1}, \{\hat{\bar{b}}\}_{\hat{r}_j-d-1}^{\hat{r}_j-1}$

(2) 计算 \bar{a}_i

If $d \leq \hat{d}$

For $i=0, 1, \dots, D$ Do

$$\bar{a}_{R_j-D-1+i} = \sum_{j=\max\{0, i-d\}}^{\min\{i, d\}} \bar{b}_{r_j-d-1+j} \bar{\hat{b}}_{\hat{r}_j-d-1+i-j}$$

Else

For $i=0, 1, \dots, D$ Do

$$\bar{a}_{R_j-D-1+i} = \sum_{j=\max\{0, i-d\}}^{\min\{i, \hat{d}\}} \bar{\hat{b}}_{\hat{r}_j-d-1+j} \bar{b}_{r_j-d-1+i-j}$$

(3) 由式(7)计算 a_i

End

在算法 1 的基础上, 可以计算全部节点区间上的乘积函数的控制系数。注意到, 从顺次三个节点区间 $[t_j, t_{j+1}], [t_{j+1}, t_{j+2}], [t_{j+2}, t_{j+3}]$ 可以分别得到 $\{a_i\}_{R_j-D-1}^{R_j-1}, \{a_i\}_{R_{j+1}-D-1}^{R_{j+1}-1}, \{a_i\}_{R_{j+2}-D-1}^{R_{j+2}-1}$, 因此, 如果 $R_{j+2}-1 \leq R_j-D-1$, 则不必在节点区间 $[t_{j+1}, t_{j+2}]$ 上进行计算, 而可以直接从区间 $[t_j, t_{j+1}]$ 跳跃到 $[t_{j+2}, t_{j+3}]$ 上去。

算法 2 在整个区间上计算乘积函数的控制系数

Input: $B(t)$ 和 $C(t)$ 的所有控制系数 $\{b_j\}, \{\hat{b}_k\}$ 以及节点向量

T, \hat{T}, \bar{T}

Output: 乘积函数的所有控制系数 $\{a_i\}$

Begin:

1. $J:=0$

2. 由算法 1 计算 $\{a_i\}_{R_j-D-1}^{R_j-1}$

3. While($J \leq s-1$)

If $R_{j+2}-D-1 \leq R_j-1$

$J:=J+2$

Else

$J:=J+1$

计算 $\{a_i\}_{R_j-D-1}^{R_j-1}$

End If

End While

在对 B 样条曲面进行光滑拼接的科研实践中发现, 有大量的计算耗费在了拼接函数和空间曲线的乘积计算上。

例如, 设有两邻接 $r_L \times s$ 次张量积 B 样条曲面 $P_L(u_L, v)$ 和 $r_R \times s$ 次张量积 B 样条曲面 $P_R(u_R, v)$, 其公共边界是 $\Phi(v) = P_L(0, v) = P_R(0, v), v \in [0, 1]$, 则两曲面 G^1 连续的充分必要条件是:

$$\alpha(v) \frac{\partial P_L}{\partial u_L} \Big|_{u_L=0} + \beta(v) \frac{\partial P_R}{\partial u_R} \Big|_{u_R=0} + \gamma(v) \frac{d\Phi(v)}{dv} = 0 \quad (8)$$

式(8)中的 $\alpha(v), \beta(v), \gamma(v)$ 称为拼接函数, 而 $\frac{\partial P_L}{\partial u_L} \Big|_{u_L=0}$,

$\frac{\partial P_R}{\partial u_R} \Big|_{u_R=0}$ 和 $\frac{d\Phi(v)}{dv} = 0$ 分别是空间中的 B 样条曲线。

在实际拼接中经常将 $\alpha(v), \beta(v)$ 取为常数, 将 $\gamma(v)$ 取为线性多项式函数, 但这无法充分利用 B 样条曲面分段连续的灵活性, 因此也有学者讨论了用其他类型拼接函数对曲面进行拼接的问题, 例如取 $\alpha(v), \beta(v)$ 为线性多项式函数, $\gamma(v)$ 为二次多项式函数等。但用高次拼接函数进行拼接计算的时候, 需要计算式(8)中拼接函数与 B 样条空间曲线的乘积之后的新的 B 样条空间曲线的控制顶点, 而目前尚无解决此问题的快速算法, 因此, 将提出的区间跳跃算法应用到曲面进行拼接问题上可以达到既简单又快速的效果。

5 实例

在 Dell-630 笔记本电脑上分别用[11]中的算法和文中提出的算法计算了线性函数与三次函数的乘积以及二次函数与三次函数乘积, 二者在计算时间上的对比如表 1 所示。由表 1 可见, 该算法的速度明显优于文献[11]中的算法。

表 1 该文算法与 Blossoming 算法的比较

类型	Blossoming 算法的时间/s	该文算法的时间/s
线性函数与三次函数的乘积	0.501 0	0.109 0
二次函数与三次函数的乘积	0.531 0	0.109 0

将线性 B 样条函数和二次 B 样条函数分别与三次 B 样条函数进行乘积计算, 其结果如图 1 和图 2 所示。两图中的三次 B 样条函数是相同的, 内部节点都是单节点。两图中的红色曲线分别代表线性 B 样条函数和二次 B 样条函数, 蓝色曲线是一个三次 B 样条函数, 绿色曲线是红色曲线和蓝色曲线的乘积。其中, 图 1 中红色曲线的节点向量为 $T=[0, 0, 1/3, 1/2, 1, 1]$, 其控制顶点(一维)为 $[0.8, 0.5, 0.9, 1.0]$, 图 2 中红色曲线的节点向量为 $T=[0, 0, 0, 1/3, 1/2, 1, 1, 1]$, 其控制顶点为 $[1.0, 0.7, 0.6, 0.7, 0.8]$ 。两图中蓝色曲线的节点向量都为 $\hat{T}=[0, 0, 0, 0, 1/3, 1/2, 1, 1, 1, 1]$, 控制顶点为 $[0.2, 0.4, 0.5, 0.3, 0.2, 0.4]$

上面两例中, 针对两标量 B 样条函数的乘积给出了例证。若其一为 B 样条曲线, 只需对 B 样条曲线的每一坐标分量分

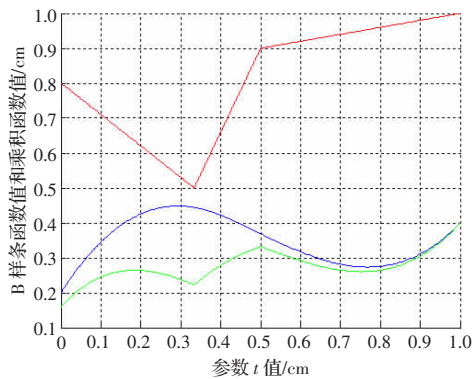


图1 线性 B 样条与三次 B 样条的乘积

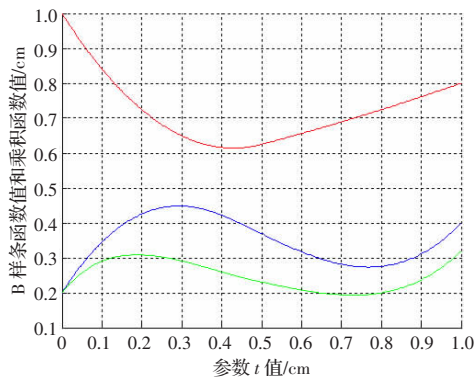


图2 二次 B 样条与三次 B 样条的乘积

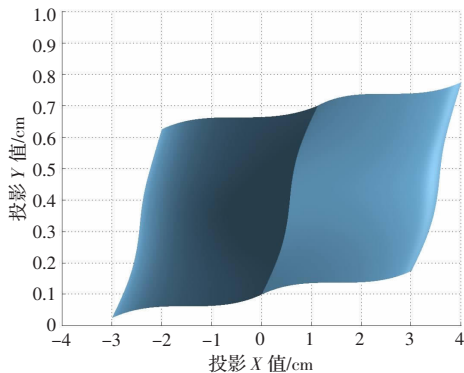


图3 两 C^0 连续的双三次 B 样条曲面

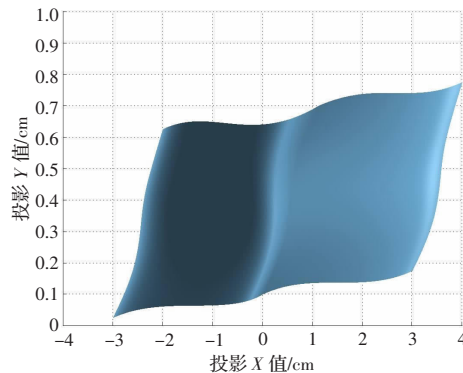


图4 对图3中两曲面进行 G^1 拼接的结果

别执行上面的算法即可得到乘积曲线的结果。

给出了在 $\alpha(v), \beta(v)$ 是线性多项式函数, $\gamma(v)$ 是二次多项式函数的情况下, 对两邻接双三次 B 样条曲面进行 G^1 光滑拼接的例子。图 3 中是两邻接 C^0 连续的双三次 B 样条曲面, 图 4 是利用如下拼接函数对其进行 G^1 拼接后的结果:

$$\begin{aligned} \alpha(v) &= 0.015 + 0.01v \\ \beta(v) &= 0.01 + 0.015v \\ \gamma(v) &= 0.001 + 0.002v + 0.003v^2 \end{aligned}$$

6 结论与展望

利用 B 样条基函数与幂基间的转化关系, 提出了计算多项式与 B 样条曲线乘积的矩阵方法。

通过实验, 该方法直接, 且易于实现。在计算时间上, 该算法明显少于文献[11]中的算法。

参考文献:

[1] Che X, Liang X, Li Q. G^1 continuity conditions of adjacent NURBS surfaces[J]. Computer-Aided Geometric Design, 2005, 22(4): 285-298.

[2] Shi X, Yu P, Wang T. G^1 Continuous conditions of biquartic B-spline surfaces[J]. Journal of Computational and Applied Mathematics, 2002, 144(1-2): 251-262.

[3] Shi X, Wang T, Wu P, et al. Reconstruction of convergent G^1 smooth B-spline surfaces[J]. Computer-Aided Geometric Design, 2004, 21(9): 893-913.

[4] Farin, G. Curves and surfaces for CAGD: A practical guide[M]. 5th ed. Hardbound: Morgan Kaufman, 2001.

[5] Cohen E, Riesenfeld R F, Elber G. Geometric modeling with splines: An introduction[M]. Natick: AK Peters, 2001.

[6] 潘日晶. B 样条曲线递推升阶方法的快速算法[J]. 计算机辅助设计与图形学学报, 2001, 13(8): 1-7.

[7] 孙景楠, 王仁宏. B 样条曲线升阶算法中的问题及其解决办法[J]. 大连理工大学学报, 2003, 43(4): 397-406.

[8] Wang G Z, Deng C Y. On the degree elevation of B-spline curves and corner cutting[J]. Computer Aided Geometric Design, 2007, 24(2): 90-98.

[9] Huang Q X, Hu S M, Matin R R. Fast degree elevation and knot insertion for B-spline curves[J]. Computer Aided Geometric Design, 2005, 22(2): 183-197.

[10] Pan R, Weng B. Recursive representation and application of transformation matrices of B-spline bases[J]. Computer-Aided Geometric Design, 2009, 26(1): 82-93.

[11] Chen X, Riesenfeld R F, Cohen E. Sliding windows algorithm for B-spline multiplication[C]//Lévy B, Manocha D. Proceedings of ACM Solid and Physical Modeling 2007, 2007: 256-276.

[12] Wen-hui D, Francis J M S. On the G^1 continuity of piecewise Bézier surfaces: a review with new results[J]. Computer-Aided Design, 1990, 22(9): 556-573.

[13] Piqiang Y, Xiquan S. G^1 continuity conditions for a kind of $K \times K$ B-spline surfaces[J]. Applied Mathematics, 2002, 15(1): 97-102.

[14] Xiquan S, Tianjun W, Peiru W, et al. Reconstruction of convergent G^1 smooth B-spline surfaces[J]. Computer Aided Geometric Design, 2004, 21(9): 893-913.

[15] Xiangjiu C, Xuezhang L, Qiang L. G^1 continuity conditions of adjacent NURBS surfaces[J]. Computer Aided Geometric Design, 2005, 22(4): 285-298.