

---

---

# 细分曲面拟合的局部渐进插值方法

赵宇, 蔺宏伟<sup>+</sup>, 鲍虎军

(浙江大学 CAD&CG 国家重点实验室 浙江 杭州 310058)

## Local Progressive Interpolation for Subdivision Surface Fitting

Yu Zhao, Hongwei Lin<sup>+</sup>, Hujun Bao

(State Key Lab of CAD&CG, Zhejiang University, 310058, China)

+ Corresponding author: E-mail: hwlin@cad.zju.edu.cn

Received 2004-00-00; Accepted 2004-00-00

**Abstract:** The quality of subdivision surface generated by the approximating scheme is usually better than that by the interpolating scheme, while the approximating subdivision surface is unable to interpolate the vertices of the initial control mesh. Traditional methods that make the approximating subdivision surface interpolate the initial mesh need to solve a global linear system. It is computation intensive, and hard to deal with dense meshes. Without solving a linear system, the progressive interpolation calculates the approximating subdivision surface that interpolates the initial mesh by adjusting the vertices of the control mesh iteratively. It can handle control meshes of any size and any topology while generating smooth subdivision surfaces that faithfully resemble the shape of the initial meshes. In this paper, we show the local property of the progressive interpolation for approximating subdivision schemes. That is, if only a subset of the vertices of the control mesh are adjusted, and others remain unchanged, the limit of the subdivision surface generated in the progressive interpolation procedure still interpolates the corresponding subset of the vertices in the initial mesh. The local property of the progressive interpolation brings more flexibility for shape controlling, and makes the adaptive fitting possible. Lots of experimental examples presented in this paper illustrate the shape controlling and adaptive fitting capabilities of the local progressive interpolation.

**Key words:** progressive interpolation, approximating subdivision scheme, local property, shape controlling, adaptive fitting, geometric design.

**摘要:** 逼近型细分方法生成的细分曲面, 其品质要优于插值型细分方法生成的细分曲面. 然而, 逼近型细分方法生成的细分曲面不能插值于初始控制网格顶点. 为使逼近型细分曲面具有插值能力, 一般通过求解全局线性方程组, 使其插值于网格顶点. 当网格顶点较多时, 求解线性方程组的计算量很大, 因此, 难以处理稠密网格. 与此不同, 在不直接求解线性方程组的情况下, 渐进插值方法通过迭代调整控制网格顶点, 最终达到插值的效果. 渐进插值方法可以处理稠密的任意拓扑网格, 生成插值于初始网格顶点的光滑细分曲面. 本文证明了逼近型细分曲面渐进插值的局部性质, 也就是, 迭代调整初始网格的若干控制顶点, 且保持剩余顶点不变, 最终生成的极限细分曲面仍插值于初始网格中被调整的那些顶点. 这种局部渐进插值性质给形状控制带来了更多的灵活性, 并且使得自适应拟合成为可能. 文中给出的结果验证了局部渐进插值的形状控制以及自适应拟合能力.

**关键词:** 渐进插值, 逼近型细分格式, 局部性质, 形状控制, 自适应拟合, 几何设计.

中图法分类号: TP391 文献标识码: A

## 1 引言

细分曲面在许多领域中有着诸多应用, 例如计算机图形学, 三维曲面建模, 以及电子游戏产品等. 传统意义下, 细分格式可以分为两类, 即插值型细分与逼近型细分. 插值型细分主要通过局部范围内对邻近点进行仿射变换而产生新顶点. 较为流行的细分格式如 Butterfly 细分格式<sup>[1]</sup>, 改进的 Butterfly 细分格式<sup>[2]</sup>, 以及 Kobbelt 细分格式<sup>[3]</sup>. 上述方法较为简单, 易于实现, 且生成的细分曲面插值于初始网格. 然而, 插值型细分格式对初始网格中的不规则性区域较敏感, 这使得插值型细分曲面的质量要差于逼近型细分曲面.

另一方面, 逼近型细分格式包括 Loop 细分<sup>[4]</sup>, Doo-Sabin 细分<sup>[5]</sup>, 以及 Catmull-Clark 细分<sup>[6]</sup>. 虽然逼近型细分可以产生较为光滑的曲面, 但是其极限曲面通常会生成不同程度的收缩. 这种收缩在初始网格点较为稀疏的情况下尤为明显. 为克服逼近型细分的这种收缩现象, 可以设法使生成的极限曲面插值于初始网格. 为达到插值的目的, 全局优化方法通过求解带有若干光顺项的线性方程组, 计算控制网格的顶点, 以使得其最终所得的细分曲面插值于初始网格<sup>[7,8]</sup>. 但是, 全局优化方法计算量较大, 难以处理具有大量顶点的稠密网格.

为节省计算量, 人们提出了若干避免求解全局线性方程组的方法. 利用准插值(quasi-interpolation)技术, Litke et. al.<sup>[9]</sup>通过计算逼近型细分曲面的控制网格, 使其准插值于给定的初始网格. Zheng 和 Cai<sup>[10]</sup>提出了一种 two-phase 细分方法, 适用于 Catmull-Clark 细分格式的任意大小的控制网格. Lai 和 Cheng<sup>[11]</sup>提出了一种利用构造过程相似性来解决插值问题的方法, 同样应用于 Catmull-Clark 细分格式.

近来, 一种名为渐进插值(progressive interpolation, 简记为 PI)的技术被应用于细分曲面拟合中. 这种技术通过迭代调整初始网格的顶点, 使得极限细分曲面插值于初始网格顶点. 已经证明, 渐进插值的迭代过程对于 Loop 细分格式<sup>[12,13]</sup>, Doo-Sabin 细分格式<sup>[14]</sup>, 以及 Catmull-Clark 细分格式<sup>[15]</sup>都是收敛的. 渐进插值的极限曲面插值于初始网格的所有顶点, 但是极限曲面的每一个控制顶点只与它邻近控制顶点有关, 可以通过局部方法求得<sup>[13]</sup>, 从而极大地降低了计算复杂度.

事实上, 细分曲面的渐进插值(PI)方法是全正基(positive basis)混合曲线曲面(blending curves and surfaces)的迭代逼近(PIA: progressive-iterative approximation)方法的拓展. PIA 方法是由 Qi et. al.<sup>[16]</sup>以及 de Boor<sup>[17]</sup>分别发现的, 在文献<sup>[18,19,20]</sup>中有进一步的推广. 近来, Lin 证明了 PIA 方法的局部性质<sup>[21]</sup>. 也就是, 仅迭代调整控制顶点集合的子集, 最终得到的极限曲线或曲面仍插值于该初始控制顶点集合的子集. 这种局部 PIA 方法在形状控制上带来了更多的灵活性, 并且使得自适应拟合成为可能. 另一种类似的数据拟合技术是 Maekawa 等人提出的 Interpolation by geometric 方法<sup>[22]</sup>.

在本文中, 我们证明了 Loop, Doo-Sabin, 以及 Catmull-Clark 等细分格式的渐进插值均具有这种局部性质. 仅调整初始网格顶点集合的某一子集, 保持其余顶点不变, 最终生成的极限细分曲面仍然插值于初始网格中这些被调整顶点. 利用局部渐进插值方法, 可以对给定的数据点进行自适应拟合, 并且细分曲面的形状控制也更加灵活.

本文的结构如下. 第二节证明了 Loop, Doo-Sabin, 以及 Catmull-Clark 细分格式的局部渐进插值算法的收敛性. 在第三节中, 我们展示了若干实验结果, 验证了局部渐进插值算法的形状控制和自适应拟合能力. 第四节总结了全文.

## 2 局部渐进插值及其收敛性

本节首先对逼近型细分方法的全局渐进插值技术做一简短回顾, 详细内容可参考文献<sup>[12,13,14,15]</sup>. 所谓全局渐进插值, 即对给定网格的全部顶点均作调整. 接着, 我们证明了逼近型细分曲面的局部渐进插值技术的收敛性. 所谓局部渐进插值技术, 即仅移动控制网格顶点集合中的某一子集, 并保持其余顶点不动. 此后, 如无特殊说明, 细分曲面均指 Loop, Doo-Sabin, 以及 Catmull-Clark 细分曲面.

## 2.1 细分曲面的全局渐进插值迭代格式

给定初始控制网格  $M_0$ , 对于  $M_0$  中的任意顶点  $V_i^0$ , 设其在所得细分曲面的极限位置为  $V_{i,\infty}^0$ . 为阐述细分曲面的全局渐进插值格式, 我们首先计算顶点  $V_i^0$  与其极限位置  $V_{i,\infty}^0$  的差向量,  $D_i^0 = V_i^0 - V_{i,\infty}^0$ , 然后, 将上述差向量与顶点  $V_i^0$  相加得到新的顶点  $V_i^1$ , 即  $V_i^1 = V_i^0 + D_i^0$ . 通过这种方法, 我们得到一个顶点为  $V_i^1$  的新网格  $M_1$ , 它与原网格  $M_0$  具有相同的拓扑结构.

上述操作迭代进行. 设在第  $k$  次迭代之后产生的新网格为  $M_k$ , 其上的顶点记为  $V_i^k$ . 类似地, 首先计算顶点  $V_i^k$  在相应细分曲面上对应的极限位置, 记为  $V_{i,\infty}^k$ , 然后计算差向量,

$$D_i^k = V_i^k - V_{i,\infty}^k. \quad (1)$$

进而, 将  $D_i^k$  与  $V_i^k$  相加, 得到新网格  $M_{k+1}$  的顶点  $V_i^{k+1} = V_i^k + D_i^k$ . 以此类推, 通过上述迭代过程, 可以得到序列  $D^k = [D_1^k, D_2^k, \dots, D_n^k], k = 0, 1, \dots$ .  $D^k$  的迭代格式用矩阵形式可表示为,

$$[D_1^{k+1}, D_2^{k+1}, \dots, D_n^{k+1}]^T = (I - B) \begin{bmatrix} D_1^k \\ D_2^k \\ \vdots \\ D_n^k \end{bmatrix} = (I - DS) \begin{bmatrix} D_1^k \\ D_2^k \\ \vdots \\ D_n^k \end{bmatrix}, \quad (2)$$

其中, 矩阵  $I$  是单位矩阵;  $B = DS$ , 矩阵  $D$  为对角元素均为正值的对角矩阵, 矩阵  $S$  是对称正定矩阵. 在文献 [12,13,14,15] 中证明了, 如果矩阵  $B$  的特征值满足,  $0 < \lambda(B) \leq 1$ , 那么矩阵  $I - B$  的谱半径均小于 1, 即  $\rho(I - B) < 1$ , 因此, 序列  $\{D^k, k = 0, 1, \dots\}$  收敛. 也就是说,

$$\lim_{k \rightarrow \infty} D^k = 0. \quad (3)$$

于是, 对于 Loop, Doo-Sabin, 以及 Catmull-Clark 细分曲面, 渐进插值生成的极限细分曲面插值于初始网格的顶点  $V_i^0$ .

值得注意的是, 在迭代格式(2)中, 网格中的所有顶点均被移动, 我们称其为**全局渐进插值**. 在下一节中, 我们将证明对应于 Loop, Doo-Sabin, 以及 Catmull-Clark 细分格式的局部渐进插值(即仅有部分顶点移动)的收敛性.

## 2.2 细分曲面的局部渐进插值迭代格式的收敛性

给定初始网格  $M_0$ , 假设仅有指标属于集合  $I = \{i_0, i_1, \dots, i_l\}$  中的顶点被移动, 称该顶点集合为**移动顶点子集**. 剩余顶点的指标集合记为  $J = \{j_0, j_1, \dots, j_j\}$ , 指标属于该集合的顶点保持位置不变, 称为**不变顶点子集**. 这种仅有部分顶点被移动的迭代格式, 称为**局部渐进插值迭代格式**. 接下来, 我们首先证明对于 Loop 细分格式而言, 局部渐进插值得到的极限细分曲面插值于初始网格  $M_0$  中的移动顶点  $V_{i_0}^0, V_{i_1}^0, \dots, V_{i_l}^0$ . 对于 Doo-Sabin 以及 Catmull-Clark 细分曲面而言具有相同的结果. 进一步, 对应于不同的细分曲面, 我们推导了不变顶点  $V_{j_0}^0, V_{j_1}^0, \dots, V_{j_j}^0$  在相应极限细分曲面上的对应位置.

为了阐述局部渐进插值格式, 需要初始控制网格  $M_0$  上顶点在细分曲面上对应极限位置的表达式. 设  $V_i^0$  为  $M_0$  上任一顶点,  $V_i^0$  在 Loop 细分曲面上对应的极限位置可表示为如下形式:

$$V_{i,\infty}^0 = \beta_n V_i^0 + (1 - \beta_n) Q_i^0, \quad (4)$$

其中,

$$\beta_n = \frac{3}{11 - 8 \times (\frac{3}{8} + (\frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{n})^2)}, \quad Q_i^0 = \frac{1}{n} \sum_{j=1}^n Q_{i,j}^0.$$

这里,  $n$  表示顶点  $V_i^0$  的度,  $Q_{i,j}^0, j = 1, 2, \dots, n$  表示与  $V_i^0$  邻接的顶点.

假设已生成控制网格序列  $M_j$ , 其上顶点记为  $V_l^j, j = 0, 1, \dots, k, l = 0, 1, \dots, m$ , 其中指标在集合  $I = \{i_0, i_1, \dots, i_l\}$  中的顶点为移动顶点. 为生成新网格  $M_{k+1}$ , 首先计算对应于每个移动顶点  $V_l^{k+1}, l \in I$  的差向量  $D_l^{k+1}$ ,

$$D_l^{k+1} = V_l^0 - V_{l,\infty}^{k+1} = V_l^0 - (\beta_n V_l^{k+1} + (1 - \beta_n) Q_l^{k+1}), \quad (5)$$

其中  $Q_l^{k+1} = \frac{1}{n} \sum_{i=1}^n Q_{l,i}^{k+1}$ ,  $Q_{l,i}^{k+1}, i = 1, 2, \dots, n$  是邻接于顶点  $V_l^{k+1}$  的顶点.

对于移动顶点对应的差向量, 有如下定理.

**定理 1** 对于 Loop 细分曲面而言, 局部渐进插值格式是收敛的. 即, 当  $k \rightarrow \infty$  时, 移动顶点对应的差向量  $D_l^k, l \in I$  收敛于 0. 这意味着极限 Loop 细分曲面插值于相应的初始网格移动顶点.

证明: 因为,

$$V_l^{k+1} = V_l^k + D_l^k, \quad Q_i^{k+1} = Q_i^k + D_{Q_i^k}^k,$$

其中,  $Q_i^k$  是邻接于顶点  $V_l^k$  的移动顶点, 它们的指标集合记为  $I_l$ . 由公式(5), 我们有,

$$\begin{aligned} D_l^{k+1} &= V_l^0 - (\beta_n V_l^k + (1 - \beta_n) Q_l^k) - (\beta_n D_l^k + \frac{1 - \beta_n}{n} \sum_{i \in I_l} D_{Q_{l,i}^k}^k) \\ &= D_l^k - (\beta_n D_l^k + \frac{1 - \beta_n}{n} \sum_{i \in I_l} D_{Q_{l,i}^k}^k). \end{aligned} \quad (6)$$

定义函数,

$$H(Q, V) = \begin{cases} 1, & \text{顶点 } Q \text{ 是移动顶点, 且与顶点 } V \text{ 邻接,} \\ 0, & \text{其他,} \end{cases}$$

并且记

$$D_l^k = [D_{i_0}^k, D_{i_1}^k, \dots, D_{i_l}^k]^T. \quad (7)$$

根据等式(6), 可以得到移动向量的迭代格式的矩阵表示形式,

$$D_l^{k+1} = (I - \bar{B}) D_l^k, \quad (8)$$

其中, 矩阵  $\bar{B}$  为,

$$\bar{B} = [b_{ij}]_{(I+1) \times (I+1)} = \begin{bmatrix} \beta_{n_{i_0}} & \dots & \frac{1 - \beta_{n_{i_0}}}{n_{i_0}} H(V_{n_{i_0}}, V_{n_{i_k}}) & \dots \\ \vdots & \ddots & & \\ \frac{1 - \beta_{n_{i_k}}}{n_{i_k}} H(V_{n_{i_0}}, V_{n_{i_k}}) & \dots & \beta_{n_{i_k}} & \dots \\ \vdots & & & \beta_{n_{i_l}} \end{bmatrix}.$$

并且矩阵  $\bar{B}$  具有如下性质:

1.  $b_{ij} \geq 0$ , 并且  $\|\bar{B}\|_\infty < 1$ ;
2. 如果  $b_{ij} = 0$ , 那么  $b_{ji} = 0$ .

此外, 还有  $\bar{B} = \bar{D}\bar{S}$ . 矩阵  $\bar{D}$  和  $\bar{S}$  分别是对称正定矩阵  $D$  和  $S(2)^{[13]}$  的主子矩阵, 所以, 矩阵  $\bar{D}$  与矩阵  $\bar{S}$  同样均为正定矩阵. 并且由于  $\|\bar{D}\| < 1$ ,  $\|\bar{S}\| < 1$ , 上述两个矩阵的特征值均满足  $0 < \lambda(\bar{D}) < 1$ ,  $0 < \lambda(\bar{S}) < 1$ . 因此, 矩阵  $\bar{B}$  的特征值满足,  $0 < \lambda(\bar{B}) < 1$ . 由此可得,  $0 < \lambda(I - \bar{B}) < 1$ . 因此, 迭代格式(8)收敛于 0. 这意味着极限 Loop 细分曲面插值于初始网格中相应的移动顶点  $V_i^0, i \in \{i_0, i_1, \dots, i_l\}$ .  $\square$

另一方面, 对应于不变顶点  $V_l^0, l \in J = \{j_0, j_1, \dots, j_J\}$  的差向量可表示为,

$$\begin{aligned} D_l^{k+1} &= V_l^0 - V_{l,\infty}^{k+1} = V_l^0 - (\beta_n V_l^{k+1} + (1 - \beta_n) Q_l^{k+1}) \\ &= V_l^0 - (\beta_n V_l^k + (1 - \beta_n) Q_l^k) - \left( \frac{1 - \beta_n}{n} \sum_{i \in I_l} D_{Q_{l,i}^k}^k \right), \\ &= D_l^k - \left( \frac{1 - \beta_n}{n} \sum_{i \in I_l} D_{Q_{l,i}^k}^k \right), \end{aligned}$$

(9)

其中,  $I_l$ 表示邻接于顶点 $V_l^k$ 的移动顶点指标集.

记  $D_J^k = [D_{j_0}^k, D_{j_1}^k, \dots, D_{j_J}^k]^T$ , 有关于不变顶点对应的差向量序列 $D_J^k$ 极限的如下定理.

**定理 2** 差向量 $D_J^k, l \in J$ 的极限为,

$$D_J^\infty = D_J^0 + C\bar{B}^{-1}D_I^0,$$

其中,  $D_I^0$ 如(7)式所示,  $\bar{B}$ 如(8)式所示; 记 $H_{j,i} = H(V_j, V_i)$ , 则,

$$C = \begin{bmatrix} -\frac{1-\beta_{n_{j_0}}}{n_{j_0}} H_{j_0, i_0} & -\frac{1-\beta_{n_{j_0}}}{n_{j_0}} H_{j_0, i_1} & \dots & -\frac{1-\beta_{n_{j_0}}}{n_{j_0}} H_{j_0, i_I} \\ -\frac{1-\beta_{n_{j_1}}}{n_{j_1}} H_{j_1, i_0} & -\frac{1-\beta_{n_{j_1}}}{n_{j_1}} H_{j_1, i_1} & \dots & -\frac{1-\beta_{n_{j_1}}}{n_{j_1}} H_{j_1, i_I} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1-\beta_{n_{j_J}}}{n_{j_J}} H_{j_J, i_0} & -\frac{1-\beta_{n_{j_J}}}{n_{j_J}} H_{j_J, i_1} & \dots & -\frac{1-\beta_{n_{j_J}}}{n_{j_J}} H_{j_J, i_I} \end{bmatrix}.$$

证明: 基于等式(9), 差向量 $D_J^k$ 的迭代格式用矩阵形式可表示为,

$$D_J^{k+1} = D_J^k + CD_I^k = D_J^0 + C \sum_{i=0}^k D_I^i. \quad (10)$$

结合公式(8),  $D_I^k = (I - \bar{B})D_I^{k-1}$ , 以及矩阵 $I - \bar{B}$ 的谱半径小于1, 即,  $\rho(I - \bar{B}) < 1$ , 有,

$$\begin{aligned} D_J^\infty &= \lim_{k \rightarrow \infty} D_J^{k+1} = D_J^0 + \lim_{k \rightarrow \infty} C \sum_{i=0}^k D_I^i \\ &= D_J^0 + \lim_{k \rightarrow \infty} C \sum_{i=0}^k (I - \bar{B})^i D_I^0 \\ &= D_J^0 + \lim_{k \rightarrow \infty} C(I - (I - \bar{B})^{k+1})\bar{B}^{-1}D_I^0 \\ &= D_J^0 + C\bar{B}^{-1}D_I^0. \end{aligned}$$

这就是不变顶点的极限位置到其自身的差向量的极限形式.  $\square$

此外, 对于 Doo-Sabin 以及 Catmull-Clark 细分曲面而言, 有类似的结果.

**定理 3** 对于 Doo-Sabin 以及 Catmull-Clark 细分曲面, 局部渐进插值格式是收敛的, 也就是,

3. 当 $k \rightarrow \infty$ 时, 移动顶点对应差向量 $D_J^k$ 趋向于0, 意味着 Doo-Sabin 以及 Catmull-Clark 细分曲面的极限插值于初始网格中的移动顶点 $V_l^0, l \in I$ ;

2. 对于 Doo-Sabin 细分曲面, 差向量 $D_J^k$ 趋向于,

$$D_J^\infty = D_J^0 + C_d \bar{B}_d^{-1} D_I^0; \quad (11)$$

3. 对于 Catmull-Clark 细分曲面而言, 差向量 $D_J^k$ 趋向于,

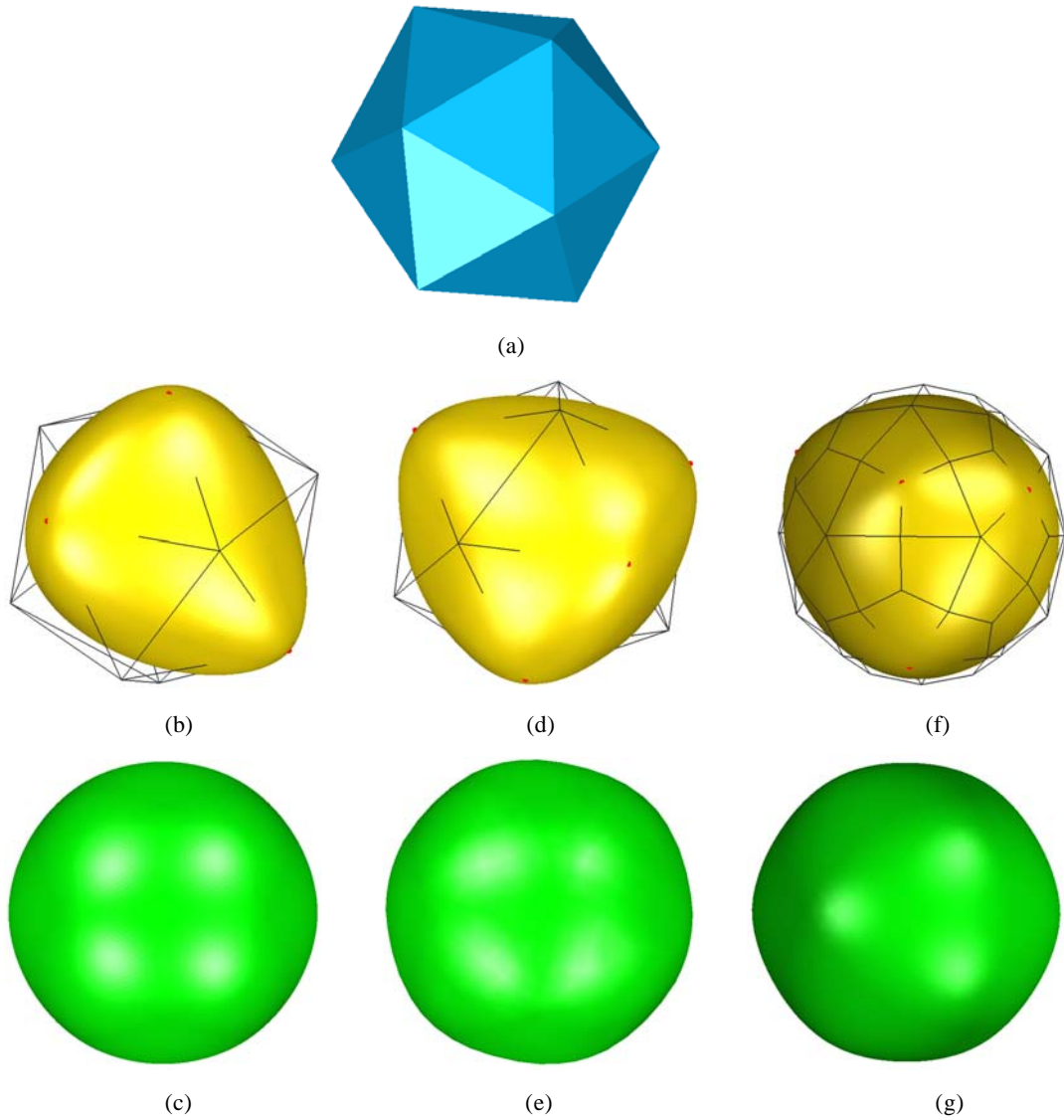
$$D_J^\infty = D_J^0 + C_a \bar{B}_a^{-1} D_I^0. \quad (12)$$

其中, 矩阵 $C_d, \bar{B}_d$ , 以及 $C_a, \bar{B}_a$ 的具体表达形式列在附录中.

定理 3 的证明过程与定理 1, 2 相似.

### 3 结果与讨论

本节给出了几个实验结果. 我们在 visual studio 2008编程环境中实现了局部渐进插值算法, 程序运行环境为, CPU: Intel Core2 Quad, 频率为: 2.66Ghz; 内存: 4G.



**图 1:** 对同一初始控制网格分别进行局部渐进插值, 和全局渐进插值的效果比较. (a) 初始控制网格—二十面体(icosagon); (b, d, f)分别为 Loop, Doo-Sabin, 以及 Catmull-Clark 细分曲面的局部渐进插值效果, 其中红色顶点表示移动顶点; (c, e, g)分别为 Loop, Doo-Sabin, 以及 Catmull-Clark 细分曲面的全局渐进插值效果.

图 1 演示了局部渐进插值在形状控制方面的能力. 图 1(a)是形状为二十面体(icosagon)的初始网格. 图 1(b), 1(d), 以及 1(f)是上述二十面体(icosagon)在采用局部渐进插值方法后的 Loop, Doo-Sabin, 以及 Catmull-Clark 细分曲面. 其中做 Catmull-Clark 细分曲面的渐进插值时采用的初始控制网格, 是由原二十面体(icosagon)(图 1(a))经一步 Catmull-Clark 细分后得到的网格(图 1(f)). 图中红色顶点表示移动顶点, 其余顶点位置保持不变. 观察图 1(b), 1(d), 以及 1(f), 可以发现极限细分曲面均插值于相应的移动顶点(红色). 作为对比, 图 1(c), 1(e), 以及 1(g)分别列出了初始网格使用三种细分格式的全局渐进插值得到的效果图. 图 1 中所有示例均为 10 次迭代后得到的结果.

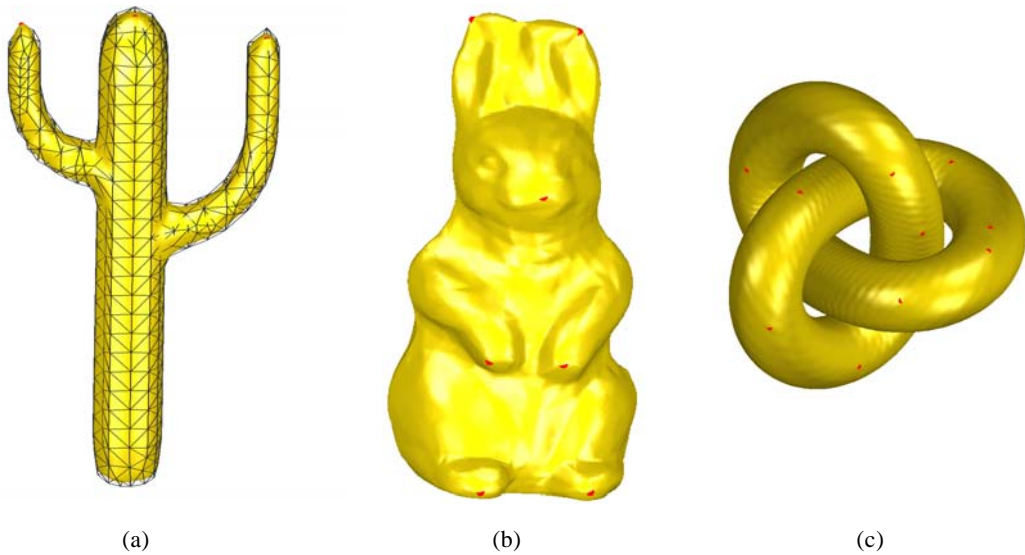
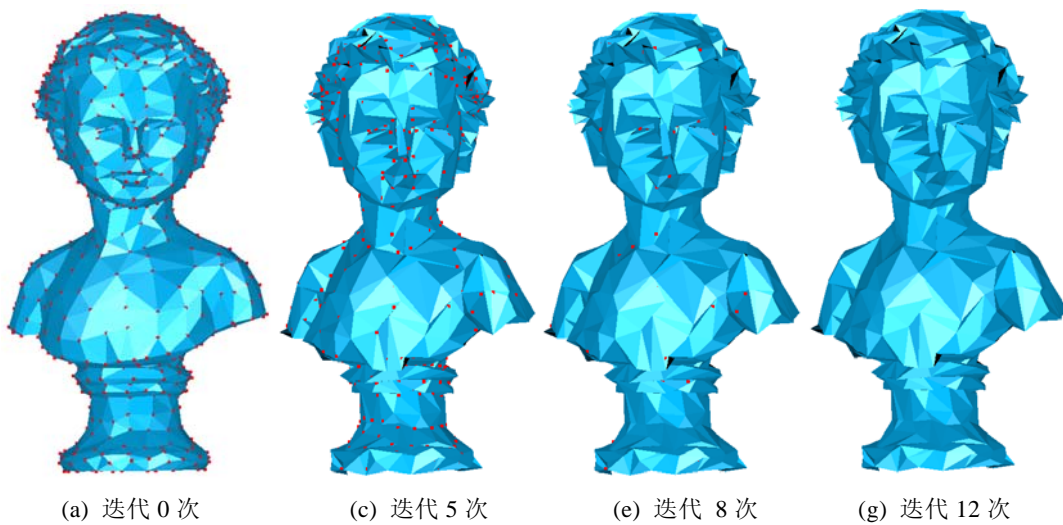


图 2: 三个不同模型在三种不同细分格式下: Loop(a.), Doo-Sabin(b.), Catmull-Clark(c.), 应用局部渐进插值算法的得到的效果图, 图中红色顶点表示移动顶点.

在图 2 中, 我们展示了具有复杂形状的网络模型分别采用三种细分格式的局部渐进插值算法后生成的效果图. 其中, 红色顶点表示移动顶点, 其余顶点保持不变. 经 10 次迭代后, 相对于初始移动顶点的拟合精度分别达到  $2.5300 \times 10^{-5}$  (图 2(a)),  $4.7847 \times 10^{-7}$  (图 2(b)), 以及  $2.1227 \times 10^{-5}$  (图 2(c)), 其中图中所示三个模型的包围盒分别为  $[-0.07, 0.07] \times [-0.3, 0.3] \times [-0.5, 0.5]$  (图 2(a)),  $[-0.4, 0.4] \times [-0.4, 0.4] \times [-0.8, 0.8]$  (图 2(b)), 以及  $[-0.6, 0.6] \times [-0.6, 0.6] \times [-0.5, 0.5]$  (图 2(c)). 局部渐进插值所采用的拟合精度用移动顶点对应差向量的范数的最大值表示.



(a) 迭代 0 次

(c) 迭代 5 次

(e) 迭代 8 次

(g) 迭代 12 次

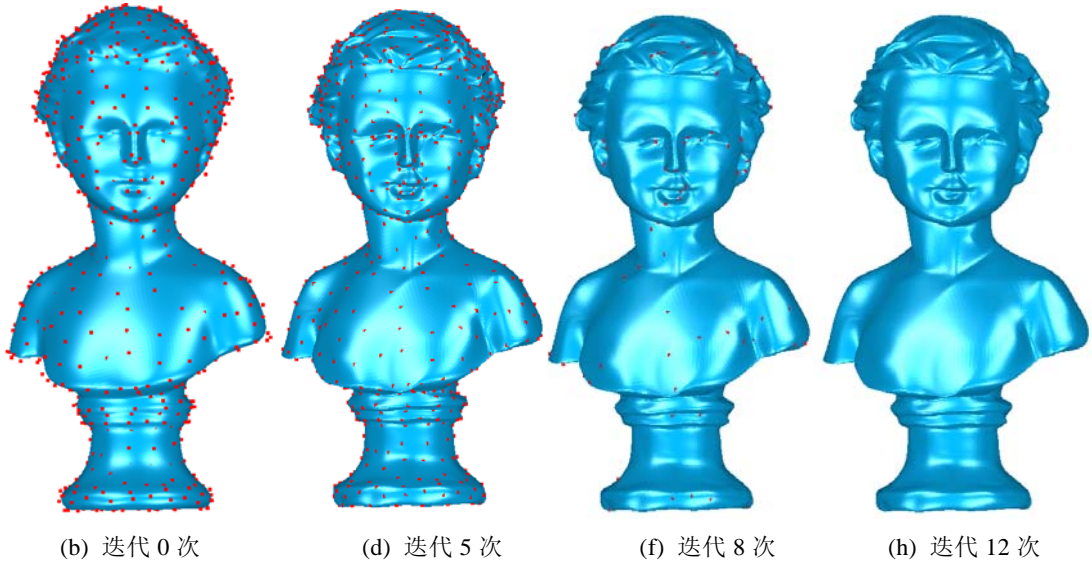


图 3: 基于 Loop 细分格式局部渐进插值算法的自适应拟合结果示意图. 图(a,c,e,g)分别为控制网格; 图(b,d,f,h)分别为对应于上述控制网格的细分曲面, 未达到阈值要求的顶点用红色标示.

渐进插值的局部性质使得自适应拟合成为可能. 在若干次迭代之后, 如果某些顶点的拟合精度达到给定阈值要求, 就可以保持其现有位置不变, 从而达到节省计算量的目的. 图 3 演示了采用 Loop 细分格式的自适应拟合过程, 其中红色顶点表示该顶点拟合精度未达到给定阈值要求  $5 \times 10^{-4}$ . 初始时, 所有顶点拟合精度均超出给定阈值(图 3(a), 3(b)). 随着迭代次数增加, 未达到阈值要求的顶点数量逐渐减少(图 3(c), 3(d), 3(e), 3(f) 所示). 经过 12 次迭代之后, 所有顶点的拟合精度均达到阈值范围内(图 3(g), 3(h)). 图 3(a), 3(c), 3(e), 以及 3(g) 为迭代中的控制网格; 相应地, 图 3(b), 3(d), 3(f), 以及 3(h) 为细分曲面. 图 3 中模型的包围盒为  $[-0.8, 0.8] \times [-0.4, 0.4] \times [-0.3, 0.3]$ .

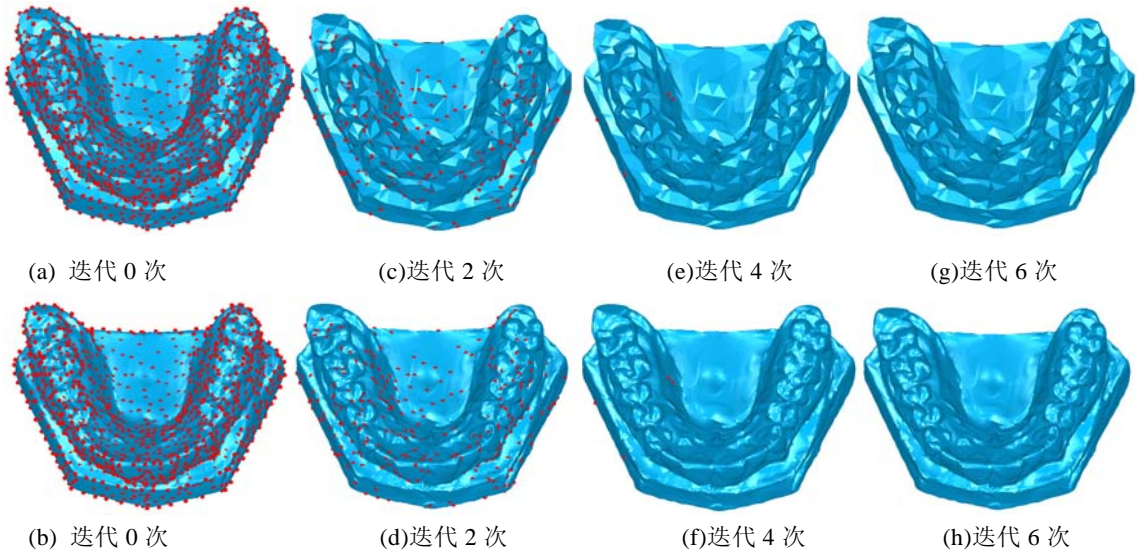


图 4: 基于 Doo-Sabin 细分格式局部渐进插值算法的自适应拟合结果示意图. 图(a,c,e,g)分别为控制网格; 图(b,d,f,h)为对应于上述控制网格的细分曲面. 拟合精度未达到阈值要求的顶点用红色标示.



图 4 为在给定阈值为  $5 \times 10^{-4}$  的条件下, 采用 Doo-Sabin 细分格式局部渐进插值的自适应拟合结果. 在经过 6 次迭代之后, 每个顶点的拟合精度均达到阈值要求. 图 4(a), 4(c), 4(e), 以及 4(g) 为自适应拟合过程中的控制网格; 图 4(b), 4(d), 4(f), 以及 4(h) 为上述控制网格对应的细分曲面. 图 4 中模型的包围盒为  $[-0.8, 0.8] \times [-0.3, 0.3] \times [-0.6, 0.6]$ .

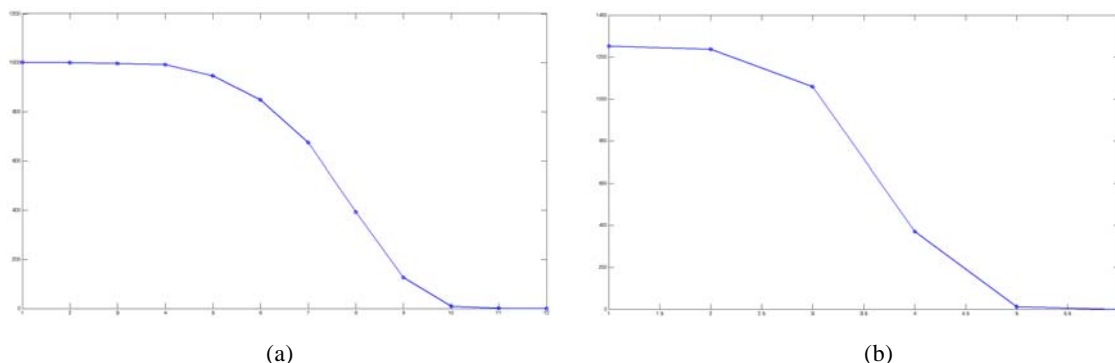


图 5: 应用局部渐进插值算法的自适应拟合过程中, 拟合精度未达到阈值要求的顶点数目的折线图. (a.) 图 3 中模型的数据; (b.) 图 4 中模型的数据.

此外, 为了直观演示采用局部渐进插值技术的自适应拟合算法节省计算量的能力, 我们在图 5 中绘制出每次迭代过程中拟合精度未达到阈值要求的顶点数目折线图. 如图 5 所示, 横坐标代表迭代次数, 纵坐标表示拟合精度未达要求的顶点数目. 显然, 随着迭代次数的增加, 未达到阈值要求的顶点数目逐步减少, 这也意味着不变顶点数目增加, 因此可以起到减少计算量的效果.

## 4 结论

文本提出了逼近型细分格式的渐进插值的局部性质. 如果初始网格顶点中仅有部分顶点做调整, 同时保持剩余顶点位置不变, 通过渐进插值算法得到的细分曲面序列的极限仍插值于初始网格中相应的移动顶点. 渐进插值算法的局部特性使得形状控制更具灵活性. 此外, 渐进插值的局部特性使得细分曲面拟合可以自适应进行, 从而减少拟合过程的计算量. 本文列举的大量实例, 验证了局部渐进插值算法在形状控制, 以及自适应拟合方面的能力.

**致谢:** 本文得到国家自然科学基金(Nos. 60970150, 60933008), 浙江省自然科学基金(No. Y1090416)资助.

## 附录

本附录给出了等式(11)中矩阵  $\bar{B}_d$ ,  $C_d$  以及等式(12)中矩阵  $\bar{B}_a$ ,  $C_a$  的具体形式. 如上所述, 对于初始控制网格  $M_0$ , 局部渐进插值方法生成的网格序列记为  $\{M_k, k = 0, 1, \dots\}$ , 其上顶点记为  $V_i^k, i = 0, 1, \dots, n$ , 其中, 移动顶点记为  $V_i^k, i \in \{i_0, i_1, \dots, i_l\}$ , 不变顶点记为  $V_j^k, j \in \{j_0, j_1, \dots, j_r\}$ .

对于 Doo-Sabin 细分格式而言, 矩阵  $\bar{B}_d$  以及  $C_d$ (11) 表示如下, 记  $\bar{B}_d = [b_{pq}]$ , 那么,

$$b_{pq} = \begin{cases} \lambda_k + \sum_{k=1}^{n_p} \frac{1-\lambda_k}{n_{p,f_k}}, & p = q, \\ \frac{1-\lambda_k}{n_{p,f_k}} + \frac{1-\lambda_{k-1}}{n_{p,f_{k-1}}}, & \text{如果 } H(V_{i_p}, V_{i_q}) = 1, \text{ 且 } V_{i_q} \text{ 是 } V_{i_p} \text{ 的第 } k \text{ 个边点,} \\ \frac{1-\lambda_k}{n_{p,f_k}}, & \text{如果 } H(V_{i_p}, V_{i_q}) = 1, \text{ 且 } V_{i_q} \text{ 是 } V_{i_p} \text{ 的第 } k \text{ 个面点,} \\ 0, & \text{否则.} \end{cases}$$

同时, 记  $C_d = [c_{pq}]$ , 则,

$$c_{pq} = \begin{cases} -\left(\frac{1-\lambda_k}{n_{p,f_k}} + \frac{1-\lambda_{k-1}}{n_{p,f_{k-1}}}\right), & \text{如果 } H(V_{i_p}, V_{j_q}) = 1, \text{ 且 } V_{j_q} \text{ 是 } V_{i_p} \text{ 的第 } k \text{ 个边点,} \\ -\frac{1-\lambda_k}{n_{p,f_k}}, & \text{如果 } H(V_{i_p}, V_{j_q}) = 1, \text{ 且 } V_{j_q} \text{ 是 } V_{i_p} \text{ 的第 } k \text{ 个面点,} \\ 0, & \text{否则.} \end{cases}$$

这里使用的是修订的 Doo-Sabin 细分格式[FL08], 构造矩阵  $\bar{B}_d$  及  $C_d$  的具体细节可参考文献<sup>[12]</sup>.

另一方面, 对于 Catmull-Clark 细分格式而言, 矩阵  $\bar{B}_a$  以及  $C_a$ (12) 的形式如下,

$$\bar{B}_a = \begin{bmatrix} \frac{n_{i_0}^2}{n_{i_0}(n_{i_0}+5)} & \cdots & \frac{4H_{i_0,i_k}}{n_{i_0}(n_{i_0}+5)} & \cdots & \frac{H_{i_0,i_l}}{n_{i_0}(n_{i_0}+5)} & \cdots \\ \vdots & \ddots & & & & \\ \frac{4H_{i_0,i_k}}{n_{i_k}(n_{i_k}+5)} & \cdots & \frac{n_{i_k}^2}{n_{i_k}(n_{i_k}+5)} & & & \\ \vdots & & & \ddots & & \\ \cdots & & & & \cdots & \frac{n_{i_l}^2}{n_{i_l}(n_{i_l}+5)} \end{bmatrix},$$

及,

$$C_a = \begin{bmatrix} -\frac{4H_{i_0,j_0}}{n_{j_0}(n_{j_0}+5)} & \cdots & -\frac{H_{i_k,j_0}}{n_{j_0}(n_{j_0}+5)} & \cdots \\ \vdots & & \vdots & \\ -\frac{4H_{i_0,j_l}}{n_{j_l}(n_{j_l}+5)} & \cdots & -\frac{H_{i_k,j_l}}{n_{j_l}(n_{j_l}+5)} & \cdots \\ \vdots & & \vdots & \\ -\frac{4H_{i_0,j_J}}{n_{j_J}(n_{j_J}+5)} & \cdots & -\frac{H_{i_k,j_J}}{n_{j_J}(n_{j_J}+5)} & \cdots \end{bmatrix},$$

其中,  $H_{i,j} = H(V_i, V_j)$ . 构造矩阵  $\bar{B}_a$  及  $C_a$  的细节可参考文献<sup>[15]</sup>.

#### 参考文献:

- [1] Dyn N., Levin D., Gregory J. A.: A butterfly subdivision scheme for surface interpolation with tension control. ACM Trans. Graphics, 1990, 9(2): 160-169.
- [2] Zorin D., Schröder P., Sweldens W.: Interpolating subdivision for meshes with arbitrary topology. Computer Graphics, Ann. Conf. Series, 1996, 30: 189-192.
- [3] Kobbelt L.: Interpolatory subdivision on open quadrilateral nets with arbitrary topology. Comput. Graph. Forum, 1996, 5(3): 409-420.
- [4] Loop C. Smooth subdivision surfaces based on triangles. Master' Thesis. Dept. Math., Univ. Utah, 1978.
- [5] Doo D., Sabin M.: Behaviour of recursively division surfaces near extraordinary points. Computer-Aided Design, 1978, 10(6): 356-360.

- 
- [6] Catmull E., Clark J.: Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 1978, 10(6): 350-355.
- [7] Nasri A. H.: Surface interpolation on irregular networks with normal conditions. *Computer Aided Geometric Design*, 1991, 8: 89-96.
- [8] Halstead M., Kass M., DeRose T.: Efficient, fair interpolation using Catmull-Clark surfaces. In *Proc. SIGGRAPH 1993*, Anaheim, USA, August 1-6, 1993, pp.47-61.
- [9] Litke N., Levin A., Schröder P.: Fitting subdivision surfaces. In *Proc. Visualization 2001*, San Diego, USA, Oct. 21-26, 2001, pp.319-324.
- [10] Zheng J., Cai Y.: Interpolation over arbitrary topology meshes using a two-phase subdivision scheme. *IEEE Trans. Visualization and Computer Graphics*, 2006, 12(3): 301-310.
- [11] Lai S., Cheng F.: Similarity based interpolation using Catmull-Clark subdivision surfaces. *The Visual Computer*, 2006, 22(9): 865-873.
- [12] Cheng F., Fan F., Lai S., Huang C., Wang J., Yong J.: Progressive interpolation using loop subdivision surfaces. *GMP* 2008, 526-533.
- [13] Cheng F., Fan F., Lai S., Huang C., Wang J., Yong J.: Loop Subdivision Surface Based Progressive Interpolation. *Journal of Computer Science and Technology*, 24(1), 39-46, 2009.
- [14] Fan F., Lai S.: Subdivision based Interpolation with Shape Control. *Computer Aided Design & Applications*, 5, 1-4, 539-547, 2008.
- [15] Chen Z., Luo X., Tan L., Ye B., Chen J.: Progressive Interpolation based on Catmull-Clark Subdivision Surfaces. *Computer Graphics Forum*, 27(7), 1823-1827, 2008.
- [16] Qi D., Tian, Z., Zhang, Y., and Feng J.B.(1975): The method of numeric polish in curve fitting. *ACTA MATHEMATICA SINICA*, 18(3), 173-184.
- [17] de Boor C. (1979): How does Agee's smoothing method work? In *Proceedings of the 1979 Army Numerical Analysis and Computers Conference*, ARO Report 79-3, Army Research Office, 299-302.
- [18] Lin H., Wang G., and Dong C. (2004): Constructing Iterative Non-Uniform B-spline Curve and Surface to Fit Data Points. *SCIENCE IN CHINA, Series F*, 47(3), 315-331.
- [19] Lin H., Bao H., and Wang G. (2005): Totally Positive Bases and Progressive Iteration Approximation. *Computers and Mathematics with Applications*, 50(3-4), 575-586, 2005.4.
- [20] Shi L., Wang R. (2006): An Iterative Algorithm of NURBS Interpolation and Approximation. *Journal of Mathematical Research and Exposition*. 26(4), 735-743.
- [21] Lin H. (2010) Local progressive-iterative approximation format for blending curves and patches. *Computer Aided Geometric Design*, 27, 322-339, 2010.
- [22] Maekawa T., Matsumoto Y., Namiki K. Interpolation by geometric algorithm. *Computer-Aided Design*, 39, 313-323, 2007.