

蝶形细分面片的光顺

于行洲 查红彬 石青云

(北京大学视觉与听觉信息处理国家重点实验室, 北京 100871)

摘要 使用蝶形细分法细分一般的初始控制网格得到的细分面片光滑而不光顺, 面片的视觉效果很差, 而运用现有的光顺技术, 又只能直接光顺细分以后的结果, 其需要保存的数据不仅量大, 而且会引入误差. 针对这一问题, 提出了一种新的光顺方法, 即通过调整初始网格顶点位置来光顺细分以后的结果. 在添加合适的约束后, 该方法不仅可以在光顺细分面片的同时, 降低细分面片和三维真实物体表面之间的逼近误差, 而且由于最终输出的是初始控制网格, 故需要保存的数据量小.

关键词 计算机图形学(520.6030) 细分 网格光顺 控制网格

中图分类号: TP391.41 **文献标识码**: A **文章编号**: 1006-8961(2003)03-0299-07

Mesh Fairing of Butterfly Subdivision Surfaces

YU Xing-zhou, ZHA Hongbin, SHI Qing-yun

(National Laboratory on Machine Perception, Peking University, Beijing 100871)

Abstract Subdivision surfaces generated by subdividing general control meshes with the butterfly subdivision method are theoretically smooth. However, the surfaces are not visually smooth, because they are not fairing. Fairing the subdivision surfaces directly with traditional fairing methods has to store the huge volume data of subdivision results with approximation error introduced. In this paper, a new mesh fairing method is presented to address such problem. The subdivision result is fairied through moving the vertices of the initial control mesh other than the vertices of the subdivision surface. With simple constraints, the method can fair the subdivision surface while decreasing the approximation error between the subdivision surface and the real 3D geometrical model. Moreover, this method can reduce the volume of data to be stored since it only needs to store the updated initial control mesh. To generate fairing subdivision surface, one just needs to subdivide the initial control mesh with simple subdivision rules.

Keywords Subdivision, Mesh fairing, Control meshes

0 引言

细分技术可以通过细分任意拓扑结构的初始控制网格来得到光滑的细分面片. 由于其简洁、高效等特性, 细分技术已经在计算机辅助设计、三维几何建模、计算机动画等领域得到了广泛的应用. 市场上销售的三维动画制作软件, 如 3D Studio Max, Maya 等, 已经把细分方法作为一个标准的插件提供给用户.

按照下一层细分网格是否保存上一层细分网格

顶点的原则, 细分方法可以分为插值型细分法^[1~3]和逼近型细分法^[4,5]两类. 其中逼近型细分法得到的细分面片的光滑性比较好, 而插值型细分法更便于保留初始网格上的几何信息, 但是无论是逼近型细分法, 还是插值型细分法, 在用于真实物体几何建模时, 对于一般的初始网格, 其细分的面片上都会出现褶皱. 由于这些由细分过程引入的高频信息, 严重影响了细分面片的视觉质量^[3,6], 因此必须对细分面片进行光顺.

现有光顺技术的基本思想大多是构造线性滤波

器^[7]或者是将光顺问题转化成求解泛函极小值问题^[8]。虽然运用这些光顺算法光顺细分以后,一般都可以得到高质量的面片,但是现有的光顺技术应用到蝶形细分面片时,有以下两个问题:

(1) 面片在光顺的同时,会因收缩效应^[7,8]而引入误差。

(2) 由于只能直接光顺细分以后的面片,因此为了保存光顺结果就必须保存细分后的面片数据,这样需要保存的数据量太大。

为了克服现有方法的缺点,对于蝶形插值细分面片,提出了一种新的光顺算法。新算法选取原来真实三维物体的细密网格作为比较对象,并控制最终细分面片因光顺而引入的误差。由于该光顺算法的直接作用对象是初始控制网格,因此只需要储存初始控制网格,而不是细分面片。也就是说,要得到最终的细分面片,只需要重新细分控制网格即可。由于随着细分过程的进展,细分面片的数据量呈等比级数增长,因此该光顺算法需要保存的数据量相对要小很多。同时,该算法不仅适用于蝶形细分法,同样可以应用到其他类型的插值型细分法中。

1 网格简化与蝶形细分法

1.1 网格的简化

由于真实物体的三角形细密网格往往是均匀采样的,故数据量极大^[9],但可以采用现有的网格简化算法^[10,11]来简化细密的三维网格模型,并将简化后模型作为控制网格优化的起点。简化后网格的三角形个数取决于所要求保留的三维模型几何信息的多少,可以进行交互控制。因为采用蝶形细分法只能细分流形,所以对于简化后的三角形网格要求其必须是流形。所谓流形,即要求网格顶点周围的三角形集合拓扑等价于一个圆盘或者半圆盘。对于流形来说,三角形的边只能被两个三角形共享,或只属于一个三角形,当其中一条边只属于一个三角形时,称为边缘边。通过数据实验发现,若简化网格的顶点分布越均匀,则光顺算法的效率越高,效果越好。

1.2 蝶形细分法

Dyn, Gregory 和 Levin 首先提出了蝶形插值细分法^[1,2]。该算法所采用的细分模板如图 1(a)所示。该细分算法,在规则的区域可以得到 C^1 阶的光滑性,但是在奇异点附近,光滑性降低(与一个点相连的边的条数称为一个点的入度。如果一个内点的

入度不为 6 或者一个边缘点的入度不为 4,则称这样的点为奇异点,否则称为规则点)。Zorin 修改了 Dyn 等人的细分规则^[3],他在奇异点周围定义了新的细分规则(如图 1(b)所示)。这种新的细分规则可以保证细分面片在奇异点周围是光滑的。同时,Zorin 在边缘处又定义了新的细分规则^[12],以保证细分面片在边缘处有良好的形态。虽然 Zorin 解决了传统的蝶形细分法在奇异点附近光滑性很差的问题,但是他的算法如果用于一般的三维网格,则细分面片的质量会很差,其原因在于,由于细分面片的光顺性很差,因而严重影响了细分面片的视觉效果。本文的工作正是通过调整初始网格的顶点位置,在不引入误差的情况下,提高细分面片的光顺质量。

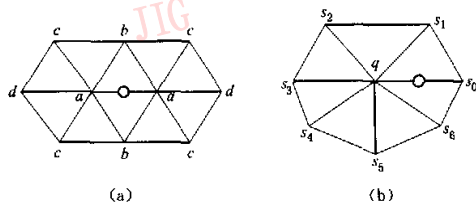


图 1 细分模板

细分面片一般是指对初始网格进行无限次细分后得到的极限结果。在实际运用中,这样做不可能,也不现实。为了得到细分面片,一般是细分数次,当细分网格的大小已经小于屏幕的分辨率时便停止。在本文的算法中,对于初始网格只进行两次细分,并将通过检测两次细分后,面片的逼近误差和光顺效果作为调整初始控制网格的依据。尽管细分的次数越多,效果越好,但是将导致计算量增大,光顺效率降低。因此细分两次是效果与效率的折中选择,实验证明,所得到的结果,通常可以接受。

2 光顺算法

2.1 算法的总体框架

首先全面描述一下光顺算法,然后详细介绍子算法。其总体算法如下:

输入真实物体的细密网格。

网格简化。

蝶形细分。

细密网格分割,误差、光顺能量初始化。

将所有的控制网格上的顶点压入堆栈,进入主循环:

do {

(1) 从堆栈中取出一个顶点。

- (2) 进行点的扰动。
 - (3) 如果满足误差约束条件,并且光顺能量减小,则进行局部细分网格更新、局部误差数据更新、局部光顺能量更新,并转向步骤 2。否则转向步骤 4。
 - (4) 如果控制网格的顶点进行了位置更新,则将该点周围受到影响的点压入堆栈,转向步骤 1。
 - (5) 如果控制网格的顶点始终无法进行位置更新,则将点出栈,转向步骤 1。
- }while(堆栈不为空)

在该算法中,每次叠代都尝试去更新一个初始控制网格顶点位置,如果顶点位置不可以更新,就把这个点推出堆栈;否则进行相应信息更新,并将所有受到影响的点压入堆栈,进入下一次循环。当所有的点都不可再移动时,光顺过程即结束。这样整个计算方法就将细分面片的光顺问题转化为在误差控制下的最优化问题。

2.2 误差的度量

为了监测细分面片和真实物体三维网格模型之间的逼近误差,可定义如下误差度量

$$e(S) = \sum_v d^2(v, \varphi(v, S)) \tag{1}$$

其中, v 是细密网格上的顶点; S 是控制网格细分两次以后的细分面片; $d(\cdot, \cdot)$ 表示空间中两个点的欧氏距离; $\varphi(v, S)$ 是真实物体的三维网格顶点 v 在细分面片上的最近投影点。其几何意义见图 2。

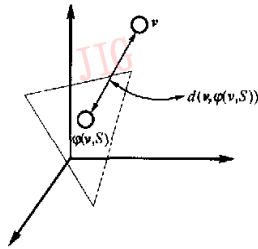


图 2 细密网格顶点在细分面片上的投影

2.3 光顺能量

为了监测细分后网格的光顺质量,因而在细分面片上定义了离散光顺能量 E_f

$$E_f(S) = \sum_v E_f(v) \tag{2}$$

这里

$$E_f(v) = \left| \frac{\sum_{j=1}^{P_v} (v - v_j) \times \bar{L}_j}{\sum_{j=1}^{P_v} \bar{L}_j} \right| \tag{3}$$

其中, P_v 是顶点 v 的入度, P_j 是与顶点 v 相连的第 j

个顶点 v_j 的入度, \bar{L}_j 为与细分面片上的顶点 v 相连接的顶点 v_j 所有人边的平均长度,其定义为

$$\bar{L}_j = \frac{\sum_{i=1}^{P_j} L_i}{P_j} \tag{4}$$

离散光顺能量 E_f 主要用于评价细分网格顶点分布的均匀程度,而光顺能量的减少,则意味着网格顶点分布得比较均匀,由于其能够使凝聚在一起的顶点分散开,故消除了细分面片表面的褶皱和提高了面片的光顺效果。

2.4 细密网格的分割和局部误差更新

每一次控制网格的点扰动后,都需要重新更新细分面片,进而需要更新逼近误差估计。细分面片更新后,为了计算细密网格顶点到细分面片上的投影点,就要对细分面片上的小三角形进行全局搜索。如果设定控制网格的个数为 m ,细密网格的顶点个数为 n ,则所需要的计算量为 $O(mn)$ 。为减小计算量,根据细密网格顶点到细分面片的距离来对原来的三维细密网格进行分割(如图 3 所示),即首先寻找每一个细密网格顶点在细分面片上的最近投影点,然后根据细分面片和初始控制网格的三角形在细分过程中建立的一一对应关系来将细密网格顶点分配到对应的初始控制网格三角形中去。经过这样的处理,就建立了控制网格的三角形与细密网格细分面片之间的对应关系。每一次控制网格顶点扰动后,只需要对相邻的 3 层之内,与控制网格三角形对应的细密网格顶点进行误差数据更新,而且仅需在与相邻的三角形对应的细分面片上找对应的投影点即可。之所以选 3 层,是因为细分的局部性要求所致,这一点将在后面的局部细分部分得到更深的认识。通过细密网格的分割,可以将细分面片逼近误差估计的计算量减少到 $O(1)$ 。

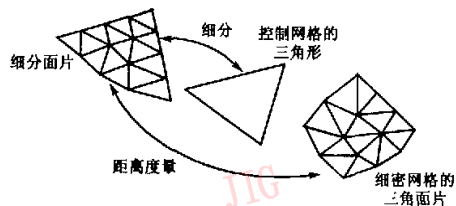


图 3 细密网格的分割

2.5 控制网格顶点的扰动

光顺算法的每一次循环,都要从堆栈里面取出控制网格的一个顶点,通过移动控制网格的顶点来

光顺细分面片. 本文采用的控制网格扰动方法, 不同于一般的光顺算法, 因为一般的光顺算法, 根据光顺法就可以在三维空间中的任意方向上移动顶点, 而本文的控制网格顶点只能在细密网格顶点上跳动, 并且只能从细密网格的一个顶点跳到另一个相邻的顶点. 这样做的原因是可以时刻监测误差的改变, 并且可通过细密网格的不断重新划分来使得误差更新的计算局部化.

在实验中, 光顺能量和逼近误差的约束条件如下:

$$\begin{cases} \omega \times \frac{(E_{li}^b - E_{li}^e)}{E_{li}^b} + (1 - \omega) \times \frac{(e_i^b - e_i^e)}{e_i^b} > 0 \\ E_{li}^b - E_{li}^e \end{cases} \quad (5)$$

其中, E_{li}, e_i 分别表示受影响的局部光顺能量和逼近误差, 上标 b 和 e 分别表示控制网格顶点扰动的前后. $\omega \in [0, 1]$ 是局部光顺能量和局部误差变化量的一个加权因子, 在实验中, 取值在 0.75 左右, 可以得到很好的效果.

光顺算法具体实现时, 在循环开始前, 可以建立控制网格的顶点和部分细密网格顶点的——对应关系. 如果有的控制网格的顶点不属于细密网格的顶点集合, 就用最近的细密网格的顶点来代替控制网格的顶点. 经过这样的预处理后, 则每一个控制网格的顶点就都与细密网格的一个顶点重合了. 在控制网格顶点扰动时, 可以在对应的细密网格邻接顶点范围内, 寻找可能的移动位置和方向(如图 4 所示).

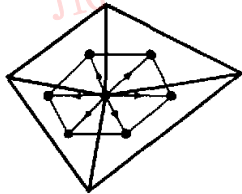


图 4 控制网格顶点的扰动

如果这样的移动可以同时满足误差约束条件和减小光顺能量, 那么移动就被接受, 并且就建立了控制网格顶点与新细密网格顶点的对应关系. 每次扰动结束后, 如果将

- (1) 曾经试探过, 但是不能接受的移动方向;
- (2) 控制网格顶点原来的位置.

两种情况标记下来, 那么在以后的扰动中就不再向这些方向试探, 以避免重复的计算. 如果对所有可能的方向进行移动都不能减小光顺能量, 并满足误差约束条件, 则顶点的扰动就结束了.

如果经过顶点的扰动, 始终没有减小光顺能量, 就将该控制网格顶点出栈, 否则将该顶点附近 3 层以内的控制网格顶点压入堆栈. 之所以只取附近 3 层, 是因为细分的局部性所致. 堆栈管理总的原则是先进先出, 且顶点扰动在堆栈为空时结束.

2.6 局部细分和局部光顺能量更新

因为蝶形细分法的细分模板是有限的(图 1), 所以当改变一个控制网格顶点的几何坐标时, 虽然受影响的区域会随着细分次数的增加而不断扩大, 但是受到影响的区域大小是有上界的. 假设以 0, 1, 2, 3, ... 来表示初始控制网格被扰动顶点周围顶点的不同层数(如图 5(a)所示); 而以 $\frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, \dots$ 来表示经过细分一次后的细分面片上得到的新控制顶点层数(如图 5(b)所示); 若以 $\frac{1}{4}, \frac{3}{4}, 1, \frac{5}{4}, \frac{3}{2}, \dots$ 来表示经过两次细分后得到的细分面片上产生的新的控制网格顶点的层数(如图 5(c)所示), 则受到影响的最外层控制网格顶点的层数和细分次数之间遵循着如下的关系:

$$\begin{cases} l_0 = 1 \\ l_1 = 2 \\ l_2 = 2 \frac{1}{2} \\ \vdots \end{cases} \quad (6)$$

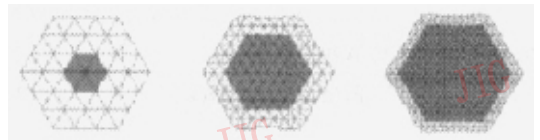
式中, l_n 表示受到影响的最外层顶点的层数, 其角标表示细分的次数.

运用不完全归纳法可以得到

$$l_n = \sum_{i=1}^n \frac{3}{2^i} + \frac{1}{2^n} \quad (7)$$

这样, 受影响顶点最大层数的计算就变成了等比数列求和问题, 所以

$$l_n = 3 - \frac{1}{2^{n-1}} \rightarrow 3, n \rightarrow \infty \quad (8)$$



(a) 没有细分 (b) 细分 1 次 (c) 细分 2 次

图 5 受到点扰动影响的细分区域(深色)

利用细分法的这种局部性, 可以提高相关的运算效率. 为了减小计算量, 提出局部细分的概念. 这

样每次控制网格顶点扰动以后,只需要根据受到影响的控制网格顶点向外传播的顺序来逐次更新相应的细分面片的几何数据和光顺能量数据,并局部更新对应细密网格分片的误差估计即可.如果用 m 表示两次细分后控制网格顶点的个数,那么细分的局部化使得相应计算量由 $O(m)$ 降到了 $O(1)$.

3 算法实现

3.1 数据结构

使用邻接表来表示各级细分网格,其头节点不但有指向表节点的指针,还有相关的光顺能量信息和顶点的空间坐标指针信息;其表节点的相应信息区域则含有头节点到邻接点的距离信息.这样的数据结构就可以很好地描述细分网格顶点的空间位置和光顺能量分布,从而便于局部细分和局部光顺能量更新.

细密网格描述同样采用邻接表的方法,其头节点中有一个符号位,表示该顶点是否与初始控制网格的顶点重合.如果细密网格顶点没有和初始控制网格的顶点重合,就要给出它的误差和它在细密网格分割时建立的与初始网格三角形的对应关系.这样的数据结构,既可以满足控制网格顶点扰动的需要,又可以根绝误差的局部更新进行细密网格的局部新划分.

为了更加便于局部的误差更新,用邻接表建立了控制网格的三角形和细密网格顶点之间的归属关

系.这种邻接表中的头节点是控制网格的三角形序号和指向表节点的指针;表节点是划入该三角形中的对应细分面片的细密网格顶点.由于这样的数据结构可以在局部误差更新和细密网格重新划分时进行批处理,因此可以提高计算效率.

3.2 收敛性

整个算法可以认为是一个约束最优化问题,因为每一次循环时,从堆栈中只取出一个顶点对其进行试探性的扰动,而只有光顺能量下降时,顶点的新位置才会被接受,由式(2)、(3)、(4)可知,由于本文定义的光顺能量是非负的,而单调递减数列若有下界,则必有下确界,所以算法是收敛的.

4 实验结果

为了验证算法的效果,使用 Stanford 大学计算机图形实验室提供的三维网格模型和北京大学视觉与听觉信息处理国家重点实验室采集的三维网格数据进行了实验,实验结果如下:

(1)对于 Stanford 大学的兔子模型,使用了 Garland 的网格简化算法^[10],将原来的模型,由 69 451 个三角形简化到 2 000 个三角形.如果将简化后的模型直接细分就会得到图 6(a)、(b)的结果,由图 6(a)、(b)可以看到,三维网格的顶点簇拥到一起,并在细分面片的表面形成了一些褶皱.这些褶皱可以通过使用本文算法优化初始网格来消除,优化后的初

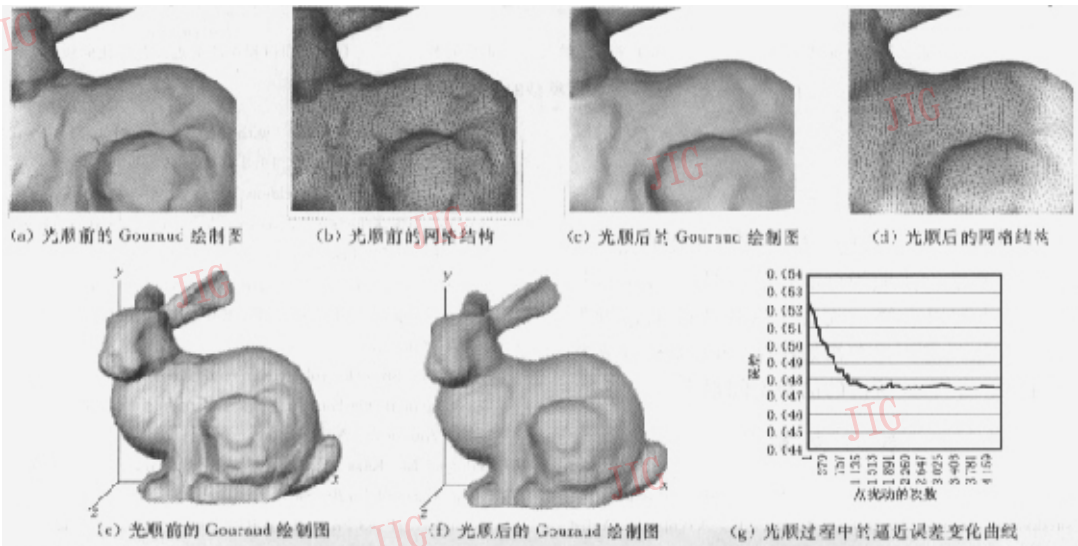


图 6 兔子模型的实验结果

始网格细分结果如图 6(c)、(d)所示. 由图 6(c)、(d) 可以看到, 面片表面的褶皱消除了, 且细分网格的顶点分布得很规则. 图 6(e)和(f)分别是光顺前后的细分面片总视图, 由图 6(e)、(f)可见, 光顺后的细分面片的视觉质量得到了显著的提高. 本文算法在光顺细分面片的同时, 还可以控制细分面片和原来三维模型之间的误差. 图 6(g)是优化过程中的误差变化图, 从该图可以看出, 误差总体上显著下降.

(2)图 7 是用北京大学实验室的人脚数据模型进行实验的结果. 实验使用三维扫描仪 FastScan 采集了真实石膏模型的三维深度数据, 并用三维建模软件 Polyworks 建立了细密的网格模型. 模型最初有 307 255 个三角形, 而利用 Garland 简化算法则可

以简化到 500 个三角形. 图 7 中各图表示的内容与图 6 中各图表示的内容相同. 采用本文算法, 同样可以在细分面片的光顺质量得到提高的同时, 降低三维细分面片与原来三维模型之间的逼近误差.

每一次细分后, 由于网格的三角形个数将变成细分前网格三角形个数的 4 倍, 因此如果采用现有的光顺算法来保存光顺的结果, 那么所需要储存的数据量就会很大. 相对来说, 本文的光顺算法只需要保存初始网格, 而不是保存细分后的面片, 其所需要保存的数据量比常规方法小得多. 例如兔子模型, 只需要保存最初的 2 000 个三角形, 而不是两次细分后的 16 000 个三角形.

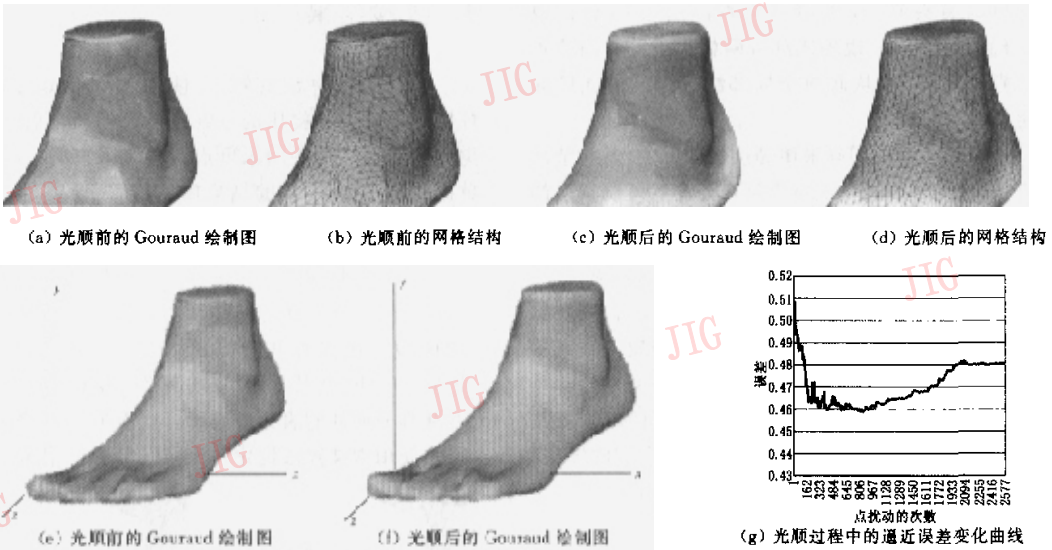


图 7 人脚模型的实验结果

5 总 结

本文设计了一种完整的蝶形细分面片的光顺算法. 由于该算法在光顺细分面片的同时, 不仅可以控制细分面片与原来的三维真实物体之间的逼近误差, 而且算法只需要保存初始网格, 而不是保存光顺之后的细分面片, 故所占的储存空间很小.

参 考 文 献

- 1 Dyn N, Levin D, Gregory J A. A4-point interpolatory subdivision scheme for curve design [J]. Computer Aided Geometric Design, 1987, 4(3):257~268.
- 2 Dyn N, Levin D, Gregory J A. A butterfly subdivision scheme

for surface interpolation with tension control [J]. ACM Transactions on Graphics, 1990, 9(2):160~169.

- 3 Zorin D, Schröder P, Sweldens W. Interpolating subdivision for meshes with arbitrary topology [A]. In: Computer Graphics (SIGGRAPH'96 Proceedings)[C], ACM, 1996:189~192.
- 4 Catmull E, Clark J. Recursive generated B-spline surfaces on arbitrary topological meshes[J]. Computer Aided Design, 1978, 10(6): 350~355.
- 5 Loop C. Smooth subdivision surfaces based on triangle. Department of mathematics[D]. Master's thesis, University of Utah, American, August 1987.
- 6 Halstead M, Kass M, DeRose T. Efficient, Fair Interpolation using Catmull-Clark Surfaces [A]. In: Computer Graphics (SIGGRAPH '93 Proceedings)[C], ACM, 1993:35~44.
- 7 Taubin G. A signal processing approach to fair surface design [A]. In: Computer Graphics (SIGGRAPH'95 Proceedings)[C],

Los Angeles CA, USA, ACM, 1995: 351~358.

- 8 Kobbelt L. Discrete fairing[A]. In Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces[C], ACM, 1996: 101~131.
- 9 Levoy M, Pulli K, Curless B *et al.* The digital michelangelo michelangelo project: 3D scanning of large statues[A]. In: Computer Graphics (SIGGRAPH'00 Proceedings)[C], ACM, 2000: 131~144.
- 10 Garland M, Heckbert P S. Surface simplification using quadric error metrics[A]. In: Computer Graphics (SIGGRAPH '97 Proceedings)[C], ACM, 1997: 209~216.
- 11 Hoppe H. Progressive meshes[A]. In: Computer Graphics (SIGGRAPH '96 Proceedings)[C], ACM, 1996: 99~108.
- 12 Zorin D, Deroose T, Schroder P, etc. Subdivision for modeling and animation[A]. In: SIGGRAPH 2000 Course Notes[C], ACM, 2000:.



查红彬 1962年生,工学博士,北京大学视觉与听觉信息处理国家重点实验室教授、博士生导师。主要研究方向为计算机视觉、三维几何建模和智能机器人。



石青云 中科院院士,1957年毕业于北京大学数学力学系,现任北京大学数学科学学院教授、应用数学专业博士生导师。主要的研究方向是模式识别、图象数据库、图象数据压缩等。



于行洲 1973年生,北京大学数学科学学院应用数学专业99级博士生。主要研究方向是计算机视觉和模式识别、三维几何建模。