

Loop 细分曲面的等距曲面的逼近

丁俊勇¹⁾ 胡事民¹⁾ 周登文²⁾

¹⁾(清华大学计算机科学与技术系 北京 100084)

²⁾(清华大学国家 CAD 工程中心 北京 100084)

摘 要 提出了一种生成 Loop 细分曲面的等距曲面的逼近方法, 通过让生成的等距曲面在对应于其控制网格顶点的地方是精确的等距, 作者将等距曲面的逼近问题转化为求解一个线性方程组, 文中提出了一个用于快速求解该线性方程组的改进的 Gauss-Jacobi 迭代法, 另外, 该文还考虑了逼近的误差控制和等距曲面的自交问题.

关键词 Loop 细分; 等距; 逼近

中图法分类号 TP391

Offset Approximation of Loop Subdivision Surfaces

DING Jun-Yong¹⁾ HU Shi-Min¹⁾ ZHOU Deng-Wen²⁾

¹⁾(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

²⁾(National CAD Engineering Center, Tsinghua University, Beijing 100084)

Abstract This paper presents an offset approximation algorithm for Loop subdivision surfaces. The main idea of this paper is to convert offset problem into solving linear equation system by making the limit positions corresponding to the new control vertices of offset surfaces be at their exact offset positions. To solve the linear equation system efficiently, an improved Gauss-Jacobi iteration method is presented. Error control for offset approximation and self-intersection of subdivision surface is considered.

Keywords loop subdivision; offset; approximation

1 引 言

等距是 CAD/CAM 系统中最重要的一操作之一^[1~3], 在众多的领域有着应用, 例如 NC 加工^[4]、机器人技术中机械手可及范围的表示^[5]等等.

对于给定的参数曲面 $s(u, v)$, 其等距曲面 $r(u, v)$ 定义为

$$r(u, v) = s(u, v) + dis \cdot n(u, v),$$

其中 $n(u, v)$ 是原参数曲面在 (u, v) 处的单位法向量, dis 是距离.

不管是等距曲线或曲面的逼近, 还是用 Bézier/B 样条精确表示等距曲线或曲面, 传统参数曲线或曲面的等距已经被广泛地研究. 然而, 作为一种新的建模工具, 有关细分曲面的等距曲面的生成尚有待于研究.

细分曲面的研究始于 20 多年前 Catmull 和 Clark^[6] 以及 Doo 和 Sabin^[7] 的两篇论文. 细分曲面被定义为对一个三维控制网格按照一定规则反复细分所得到的极限曲面. 与传统的参数曲面不同的是, 细分曲面可以作为三维模型的一种方便的任意拓扑形式的表示. 因此细分曲面被广泛地应用于电影制

作^[8]以及游戏引擎^[9]等领域.然而,由于缺少对求交以及等距这些必须的几何计算的算法,细分曲面在工程设计领域的应用一直受到限制.同时在游戏、动画制作过程中等距也有重要应用.

参考文献[10]提出了一种用于 Catmull-Clark 细分曲面的 NC 路径的生成方法,该算法中用到 Catmull-Clark 细分曲面的等距.它直接利用 Catmull-Clark 曲面的顶点计算公式建立方程,保证顶点处的等距是精确的,但并没有给出方程组的严格推导以及求解方法,也没有误差的控制及曲面的自交分析.而对于工程和游戏电影等工业中常用的 Loop 细分曲面,还没有类似的工作.

本文提出了一种用于生成 Loop 细分曲面的等距曲面的逼近算法.该算法生成的等距曲面是精确的等距曲面的一个逼近,以 Loop 细分曲面的形式表示,并且与原 Loop 细分曲面的拓扑结构相同.在第 2 节,我们通过让原控制网络的顶点对应的位置产生精确的等距,把等距曲面的生成转化为一个线性方程组的求解,并为此提出了一种改进的 Gauss-Jacobi 迭代法.方程组的建立借鉴了文献[10]的一些思想,但进行了严格的数学推导.第 3 节研究了逼近的误差控制;第 4 节考虑了生成的等距曲面的自交情况;第 5 节是本文的总结.

2 等距曲面的逼近计算

2.1 线性方程组的生成

令 M_S 表示原细分曲面 S 的控制网格, S_i^0 是 M_S 上编号为 i 的顶点,我们的目的是寻找一个新的细分曲面 F , F 是 S 的等距曲面并且和 S 的拓扑相同. M_F 是 F 的控制网格, F_i^0 是控制网格上的顶点.

把 S 和 F 上对应于 S_i^0 和 F_i^0 的极限位置记作 S_i^∞ 和 F_i^∞ . 根据 Jean 和 Schweitzer^[11] 提供的公式,这些极限可以用如下方法计算.

如图 1 所示, $S_{i_0}^0$ 是原控制网格 M_S 上一个度为

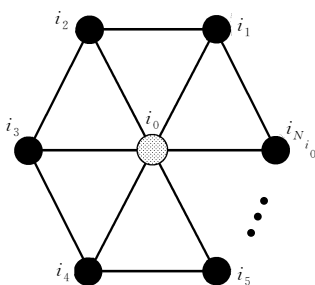


图 1 计算极限位置的示意图

N_{i_0} 的顶点,它周围的 N_{i_0} 个顶点为 $S_{i_k}^0, k=1, 2, \dots, N_{i_0}$. 那么 $S_{i_0}^0$ 的极限位置由下式给出

$$S_{i_0}^\infty = \sum_{j=0}^{N_{i_0}} a_{i_0, i_j} S_{i_j}^0 \tag{1}$$

其中当 $i_0 = i_j$ 时, $a_{i_0, i_j} = \frac{\omega}{\omega + N_{i_0}}$; 当 $S_{i_0}^0$ 和 $S_{i_j}^0$ 在同一条边上时, $a_{i_0, i_j} = \frac{1}{\omega + N_{i_0}}$, 其它情况下 $a_{i_0, i_j} = 0$. 另外 $\omega = 3/b, b = (1/N_{i_0})(5 - (3 + 2\cos(\frac{2\pi}{N_{i_0}}))^2/8)$.

同理,对于 M_F 可以得到类似的结果

$$F_{i_0}^\infty = \sum_{j=0}^{N_{i_0}} a_{i_0, i_j} F_{i_j}^0 \tag{2}$$

因为 M_S 和 M_F 具有相同的拓扑,所以可定义

$$d_i = F_i^0 - S_i^0 \tag{3}$$

式(2)-式(1)可得,

$$F_{i_0}^\infty - S_{i_0}^\infty = \sum_{j=0}^{N_{i_0}} a_{i_0, i_j} (F_{i_j}^0 - S_{i_j}^0) = \sum_{j=0}^{N_{i_0}} a_{i_0, i_j} d_{i_j} \tag{4}$$

我们的逼近策略是让生成的等距曲面 F 在控制顶点对应的位置处(例如 $F_{i_0}^\infty$)的等距是精确的. 根据等距的定义,原始表面上的 $S_{i_0}^\infty$ 和结果表面上的 $F_{i_0}^\infty$ 的距离向量 $F_{i_0}^\infty - S_{i_0}^\infty$ 应该沿着原曲面 S 上 $S_{i_0}^\infty$ 处的法线方向. 进一步的,曲面 S 上 $S_{i_0}^\infty$ 处的单位法向可以用参考文献[11]中的方法计算得到.

$$n_{i_0}^\infty = v_1 \times v_2 / \|v_1 \times v_2\| \tag{5}$$

其中 $v_1 = \sum_{j=1}^{N_{i_0}} \cos\left(\frac{2\pi(j-1)}{N_{i_0}}\right) S_{i_j}^0,$

$v_2 = \sum_{j=1}^{N_{i_0}} \sin\left(\frac{2\pi(j-1)}{N_{i_0}}\right) S_{i_j}^0.$ 根据式(4), (5)可得

$$F_{i_0}^\infty - S_{i_0}^\infty = \sum_{j=0}^{N_{i_0}} a_{i_0, i_j} d_{i_j} = dis \cdot n_{i_0}^\infty \tag{6}$$

对于一张没有折缝的封闭的细分曲面(即没有边界),它的控制网格 M_S 上的每一个顶点都可以推导出一个类似于式(6)的方程,这些方程组成线性方程组

$$Md = n \tag{7}$$

其中 M 是一个 $m \times m$ 的矩阵并且 $m_j = a_{i, j}, m$ 是网格 M_S 的顶点的个数, $d = [d_0, d_1, \dots, d_{m-1}]^T, n = [dis \cdot n_0^\infty, dis \cdot n_1^\infty, \dots, dis \cdot n_{m-1}^\infty]^T.$

一旦从方程组(7)中解出 d , 根据式(3)立刻可以计算得到等距曲面的控制网格 M_F . 这样的控制

网格满足我们的逼近策略,即控制顶点对应的位置是精确的等距.

2.2 改进的 Gauss-Jacobi 迭代法

对于线性方程组(7),传统的 Gauss-Jacobi 或者 Gauss-Seidel 迭代法并不能保证收敛. 为了有效求解方程组(7),我们提出了一种称之为边迭代的改进的 Gauss-Jacobi 迭代法. 边迭代的收敛性证明在附录中给出.

在传统的 Gauss-Jacobi 迭代中,方程(6)被改写为

$$a_{i_0, i_0} d_{i_0}^{(k+1)} + \sum_{j=1}^{N_{i_0}} a_{i_0, i_j} d_{i_j}^{(k)} = dis \cdot n_{i_0}^{\infty} \quad (8)$$

和传统 Gauss-Jacobi 迭代法不同的是,在边迭代中, $d_{i_0}^{(k+1)}$ 不只是从方程(6)改写生成的式(8)中得到,其它的一些方程也加入进来,和方程(6)一起联立求解 $d_{i_0}^{(k+1)}$. 我们以方程(6)对应的控制顶点 $S_{i_0}^0$ 的度数分度 = 3, 4 和 >4 三种情况介绍边迭代算法.

首先,我们假设点 $S_{i_0}^0$ 周围的点都是规则点(即度为 6). 这可以通过对原始控制网格进行一次细分的预处理得到保证.

(1) $S_{i_0}^0$ 的度 > 4. 见图 2.

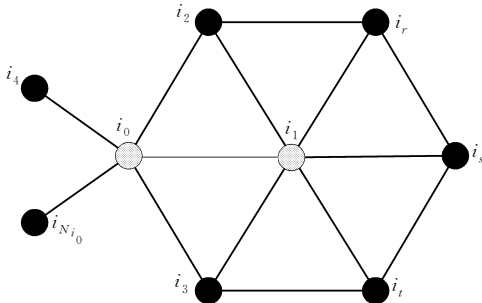


图 2 边迭代(度>4)

在方程组(7)中的每一个方程都对应于控制网格上的一个顶点. 我们选取和点 $S_{i_0}^0$ 相连的一个规则的顶点 $S_{i_1}^0$ (即度为 6), 这个点在方程组(7)中也对应着一个方程. 这 2 个点对应的方程组成 2×2 的线性方程组, 改写成迭代的形式如式(9)所示, 其中 d'_{i_0} 和 d'_{i_1} 为变量.

$$\begin{cases} a_{i_0, i_0} d'_{i_0} + a_{i_0, i_1} d'_{i_1} + \sum_{j=2}^{N_{i_0}} a_{i_0, i_j} d_{i_j}^{(k)} = dis \cdot n_{i_0}^{\infty} \\ a_{i_1, i_0} d'_{i_0} + a_{i_1, i_1} d'_{i_1} + \sum_{j=2, 3, r, s, t} a_{i_1, i_j} d_{i_j}^{(k)} = dis \cdot n_{i_1}^{\infty} \end{cases} \quad (9)$$

求解这个小方程组, 令 $d_{i_0}^{(k+1)} = d'_{i_0}$ 以用于下次迭代, d'_{i_1} 舍弃.

(2) $S_{i_0}^0$ 的度 = 3, 见图 3.

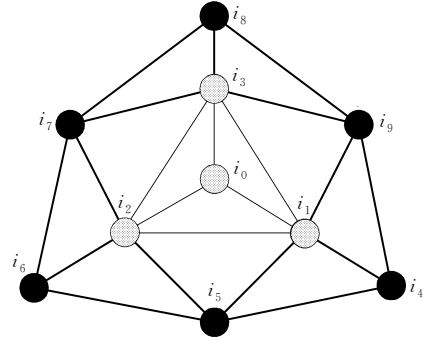


图 3 边迭代(度=3)

这种情况下, $S_{i_0}^0$ 有 3 个相邻顶点, 它们在式(7)中对应的方程和方程(6)一起组成类似于方程组(9)的小方程组, 从中求得 d'_{i_0} 并令 $d_{i_0}^{(k+1)} = d'_{i_0}$ 以用于下次迭代, $d'_{i_1}, d'_{i_2}, d'_{i_3}$ 舍弃.

(3) $S_{i_0}^0$ 的度 = 4, 见图 4.

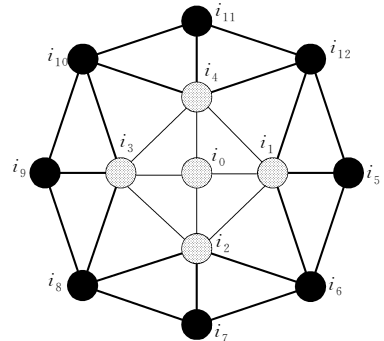


图 4 边迭代(度=4)

和度 = 3 相似, 对应于 $S_{i_0}^0$ 以及和 $S_{i_0}^0$ 相邻的 4 个顶点的一个 5 个方程组成类似于式(9)的 5×5 方程组, 求解得到 $d_{i_0}^{(k+1)}$ 用于下一次迭代.

实验过程中, 直接把控制网格的等距(法向取原始细分曲面上的对应处的法向)作为初始状态, 一般迭代 5 至 20 次后能收敛到满意的结果.

下面是采用我们的算法得到的等距曲面的逼近结果. 在图 5~7 中, (a) 是原始的控制网格, (b) 是细分过一两次的控制网格, 通过细分, 保证了每一个顶点都至少有一个规则点和它在同一条边上. (c) 是原控制网格对应的细分曲面, (d) 是我们算法生成的等距曲面的控制网格, (e) 是 (d) 中控制网格对应的细分曲面, (f) 是误差的情况, (g) 是 (f) 中误差对应的颜色索引.

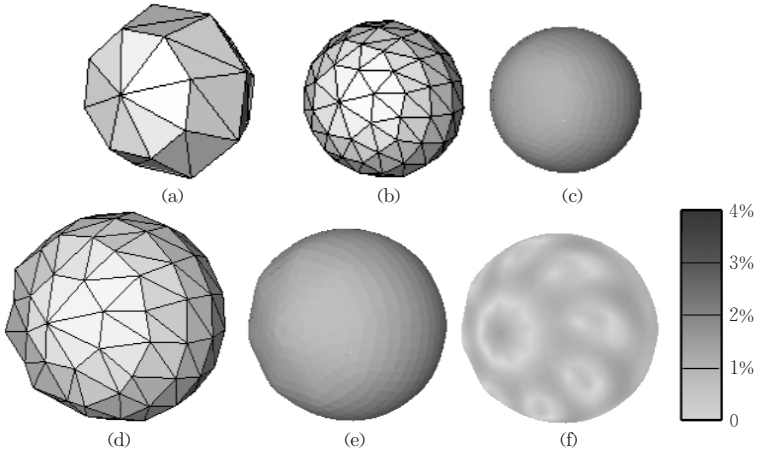


图 5 一个近似的球面的等距曲面的逼近结果

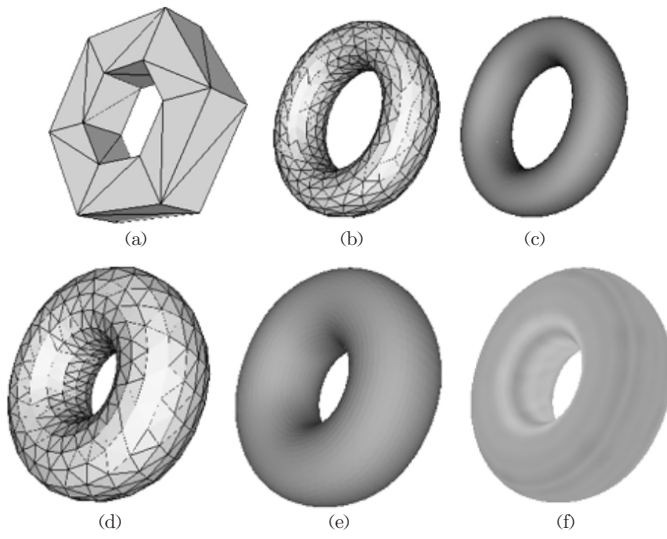


图 6 一个近似的环的等距曲面的逼近结果

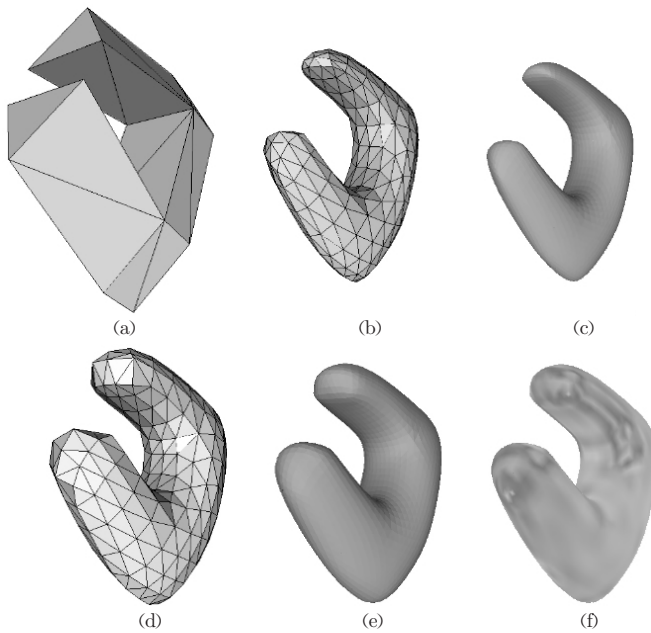


图 7 一个马蹄形的等距曲面的逼近结果

3 误差控制

我们通过对原始网格进行细分来提高逼近的精确度. 求得一个等距曲面以后, 我们提供下面的算法来估计这个等距曲面的最大误差, 如果误差大于容许范围, 就对原始网格进行一次细分, 然后求等距曲面, 如此直到误差符合要求. 下面给出我们提出的估计最大误差的算法.

根据 Joe Stam^[12] 的算法, 我们可以精确地确定 Loop 细分曲面上任意一点. S 和 F 可以改写成参数的形式 $S(v, w)$ 及 $F(v, w)$. 令

$$\delta(v, w) = S(v, w) - F(v, w) \quad (10)$$

如果等距曲面是精确的, $\|\delta(v, w)\| = dis$ 应该在任意参数处成立. 所以我们用

$$\epsilon(v, w) = \|\delta(v, w)\| - dis \quad (11)$$

来分析误差.

根据参考文献[12], 对于图 8 中对应于控制网格上的三角形 (478) 的面片可以如下计算

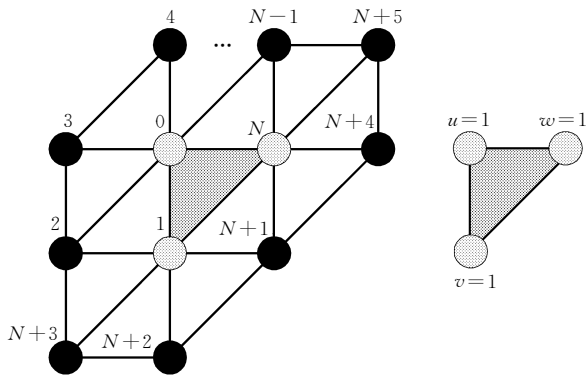


图 8 Loop 细分曲面的计算

$$\begin{cases} S(v, w) = \sum_{i=0}^{N+5} \varphi_i(v, w) S_i^0 \\ F(v, w) = \sum_{i=0}^{N+5} \varphi_i(v, w) F_i^0 = \sum_{i=0}^{N+5} \varphi_i(v, w) (S_i^0 + d_i) \\ \delta(v, w) = F(v, w) - S(v, w) = \sum_{i=0}^{N+5} \varphi_i(v, w) d_i \end{cases} \quad (12)$$

这里 N 是顶点 4 的度数.

从上面的表示可以看出, $\delta(v, w)$ 是和原细分曲面有着相同拓扑的细分曲面, $\delta(v, w)$ 的控制网格的顶点是 d_i . 所以误差的最大值估计转化为 Loop 细分曲面 $\delta(v, w)$ 的极值问题.

有多种方法可以求得细分曲面的极值. 从快速方便的角度, 我们采用常用的包围盒技术 AABBs (Axis-Aligned Bounding Boxes) 来估计 $\delta(v, w)$ 的极值. 根据细分曲面的凸包性, 一个面片的包围盒可以通过计算这块面片对应的控制网格的包围盒得到. 为了得到更加紧密的包围盒, 我们通过细分把这块面片分成一系列子面片, 然后再计算这些子面片的包围盒的并集.

在为细分曲面 $\delta(v, w)$ 的每一个面片都求得了包围盒(AABBs)以后, 很容易就得到 $\|\delta(v, w)\|$ 的极值. 从而根据式(11), 估计出误差的最大值以用于误差控制.

我们在此只是给出了一种可行的误差控制策略, 进一步的改进已经列在未来的工作计划中.

4 自交检测

对于等距操作, 自交检测的操作必不可少. 等距曲面存在两种类型的自交^[13]. 局部自交是指因为一个面片的曲率过高而导致的相交; 全局自交是指那些过于相近的面片产生的自交.

由于我们生成的等距曲面是 Loop 细分曲面, 所以可以利用已有的用于检测 Loop 细分曲面自交的算法检测等距曲面的自交情况. Elber 和 Cohen^[13] 提出了一个用于检测细分曲面自交情况的有效算法. 本文直接采用了这个算法来检测等距曲面的自交情况(具体细节见参考文献[13]). 算法的主要思想是对细分曲面上的面片求出法线的包围盒. 根据这些包围盒, 我们可以判断一个面片是否足够平坦到可以近似为一个平面来计算全局自交(不够平坦的话可以先细分, 然后再考虑它的子面片); 我们还可以利用这些包围盒得到 Gauss 映射^[13] 来判断一个面片是否局部自交.

图 9 展示了在我们计算的等距曲面上应用这个

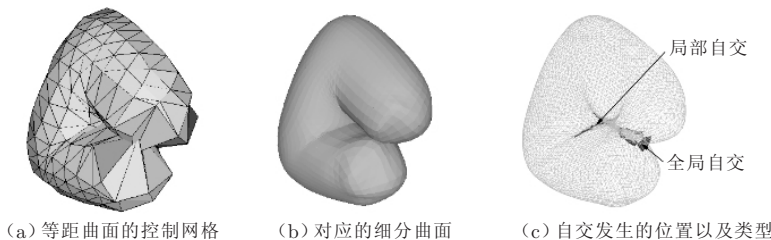


图 9 等距曲面的自交检测结果

自交检测算法的结果.

5 结论与展望

本文通过让控制网格上顶点对应的位置处的等距是精确的,把 Loop 细分曲面的等距曲面的逼近问题转化为线性方程组的求解问题并为之提供了有效的迭代解法.我们的算法生成的等距曲面是和原曲面(或者原曲面细分几次后的曲面)有着相同拓扑结构的 Loop 细分曲面,并且每一个面片的顶点位置处都是精确的等距.同时,因为是逼近算法以及等距曲面的原因,我们提供了误差的控制以及自交检测的算法.

未来有意义的工作包括:(1)误差控制策略的进一步改进.一般而言,整体细分会导致迭代过程的重新进行,从而影响整个算法的速度.未来工作将会针对细分曲面以及本算法的特点提出新的有效的误差控制策略.(2)应用 Stam^[12]的根据参数计算细分曲面上点的算法,使用类似传统参数曲面的等距曲面的方法,生成更加精确的逼近曲面.(3)研究哪类细分曲面的等距曲面可以用细分曲面精确地表示.

参 考 文 献

- 1 Farouki R T, Sakkalis T. Pythagorean hodographs. IBM Journal of Research and Development, 1990, 34(5): 736~752
- 2 Maekawa T. An overview of offset curves and surfaces. Computer-Aided Design, 1999, 31(2): 165~173
- 3 Pottmann H. Rational curves and surfaces with rational offsets. Computer Aided Geometric Design, 1995, 12: 175~192
- 4 Kuragano T, Sasaki N, Kikuchi A. The FRES DAM system

- for designing and manufacturing freeform objects. In: Matrin R ed. USA Japan Cross Bridge. Flexible Automation, 1988, 2: 931~938
- 5 Lozano-Perez T, Wesley MA. An algorithm for planning collision free paths amongst polyhedral obstacles. Communications of the ACM, 1979, 25(9): 560~570
- 6 Catmull E, Clark J. Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-Aided Design, 1978, 10(6): 350~355
- 7 Doo D, Sabin M. Analysis of the behaviour of recursive division surfaces near extraordinary points. Computer-Aided Design, 1978, 10(6): 356~360
- 8 DeRose T, Kass M, Truong T. Subdivision surfaces in character animation. In: Proceedings of SIGGRAPH'98, Orlando, FL, USA, 1998. 85~94
- 9 Zorin D, Schröder P. Subdivision for modeling and animation. In: SIGGRAPH'2000 Course Notes. Included on Course Notes CD-ROM, 2000
- 10 Joe Kurgano, Hiromasa Suzuki, Fumihiko Kimura. Generation of NC tool path for subdivision surfaces. In: Proceedings of CAD/Graphics'2001, 2001, Kunming, China, 2001. 676~682
- 11 Schweitzer J E. Analysis and application of subdivision surface [Ph D dissertation]. University of Washington, Seattle, 1996
- 12 Stam J. Evaluation of loop subdivision surfaces. In: SIGGRAPH'98. Included on course notes CD-ROM, 1998
- 13 Elber G, Cohen E. Error bounded variable distance offset operator for free form curves and surfaces. International Journal of Computational Geometry and Applications, 1991, 1(1): 67~78
- 14 Eitan G, Schröder P. Normal bounds for subdivision—Surface interference detection. In: Proceedings of IEEE Scientific Visualization, San Diego, CA, USA, 2001. 333~340
- 15 Jennings A, McKeown J J. Matrix Computation, 2nd edition. Hoboken, NJ, USA: John Wiley & Sons Ltd, 1992

附录. 边迭代收敛性的证明.

边迭代的收敛性的证明等价于证明边迭代可以写成如下形式

$$d^{(k+1)} = Cd^{(k)} + f \tag{I}$$

并且矩阵 C 的每一个特征值 λ_i 的绝对值均小于 1. 根据 Gerschgorin's 定理^[15], 对于每一个特征值 λ_i, C 中必有一行使得下式成立

$$|\lambda_i - c_{kk}| < \sum_{j \neq k} |c_{kj}| \tag{II}$$

所以, 下面我们只要证明对于 C 的每一行, 都有 $\sum_{j \neq k} |c_{kj}| < 1$

并且 c_{kk} = 0 即可.

根据边迭代的算法, c_{kk} = 0 显然成立. 对于 C 的每一行, 都有一个顶点和它对应, 我们根据该点的度数分 3 种情况 (>4, = 3 和 =4) 来证明 $\sum_{j \neq k} |c_{kj}| < 1$.

(1) 度 > 4.

如图 2 所示, 点 S_{i₀}⁰ 的度大于 4, 边迭代中的每一步有

$d_{i_0}^{(k+1)} = d'_{i_0}$, 其中 d'_{i_0} 通过求解 2×2 的方程组 (9) 得到.

因此

$$d_{i_0}^{(k+1)} = \frac{\begin{vmatrix} dis \cdot n_{i_0}^\infty - \sum_{j=2}^{N_{i_0}} a_{i_0, i_j} d_{i_j}^{(k)} & a_{i_0, i_1} \\ dis \cdot n_{i_1}^\infty - \sum_{j=2,3,r,s,t} a_{i_1, i_j} d_{i_j}^{(k)} & a_{i_1, i_1} \end{vmatrix}}{\begin{vmatrix} a_{i_0, i_0} & a_{i_0, i_1} \\ a_{i_1, i_0} & a_{i_1, i_1} \end{vmatrix}}$$

$$\triangleq [c_{i_0, 0}, c_{i_0, 1}, \dots, c_{i_0, m-1}] [d_0^{(k)}, d_1^{(k)}, \dots, d_{m-1}^{(k)}]^T + f_{i_0} \tag{III}$$

其中 $c_{i_2} = (-a_{i_0, i_2} a_{i_1, i_1} + a_{i_1, i_2} a_{i_0, i_1}) / g$, $c_{i_3} = (-a_{i_0, i_3} a_{i_1, i_1} + a_{i_1, i_3} a_{i_0, i_1}) / g$, 当 $j = 4, \dots, N_{i_0}$ 时 $c_{i_j} = -a_{i_0, i_j} a_{i_1, i_1} / g$, 当 $j = r, s, t$ 时 $c_{i_j} = a_{i_1, i_j} a_{i_0, i_1} / g$. 其它情况下, $c_j = 0$, $f = (n_{i_0}^\infty a_{i_1, i_1}$

$$-n_{i_1}^{\infty} a_{i_0, i_1}) \cdot dis/g, g = \begin{vmatrix} a_{i_0, i_0} & a_{i_0, i_1} \\ a_{i_1, i_0} & a_{i_1, i_1} \end{vmatrix}.$$

令

$$\begin{aligned} \Delta_{i_0} &= \sum_{\substack{j=0 \\ j \neq i_0}}^{m-1} |c_j| \\ &= (a_{i_1, i_1} \sum_{j=1}^{N_{i_0}} a_{i_0, i_j} + a_{i_0, i_1} \sum_{j=r, s, t} a_{i_1, i_j} + \\ &\quad |a_{i_0, i_2} a_{i_1, i_1} - a_{i_1, i_2} a_{i_0, i_1}| + |a_{i_0, i_3} a_{i_1, i_1} - a_{i_1, i_3} a_{i_0, i_1}|) / |g|, \end{aligned}$$

$$\Delta_{i_0} = \sum_{\substack{j=0 \\ j \neq i_0}}^{m-1} |c_j| = \sum_{i=1}^9 |c_j| = \frac{3 \left(\left| \begin{vmatrix} 0 & 0.2 & 0.2 & 0.2 \\ -1/12 & 1/2 & 1/12 & 1/12 \\ 0 & 1/12 & 1/2 & 1/12 \\ 0 & 1/12 & 1/12 & 1/2 \end{vmatrix} \right| + \left| \begin{vmatrix} 0 & 0.2 & 0.2 & 0.2 \\ -1/12 & 1/2 & 1/12 & 1/12 \\ 0 & 1/12 & 1/2 & 1/12 \\ -1/12 & 1/12 & 1/12 & 1/2 \end{vmatrix} \right| \right)}{\begin{vmatrix} 0.4 & 0.2 & 0.2 & 0.2 \\ 1/12 & 1/2 & 1/12 & 1/12 \\ 1/12 & 1/12 & 1/2 & 1/12 \\ 1/12 & 1/12 & 1/12 & 1/2 \end{vmatrix}} = 0.6923 < 1$$

根据 S_1^0 是规则点(度为 6)可得 $a_{i_1, i_1} = 0.5$, 当 $j = 0, 2, 3, r, s, t$ 时 $a_{i_1, i_j} = 1/12$; $a_{i_0, i_0} = \frac{\omega}{\omega + N_{i_0}}$, 当 $j = 1, 2, 3, \dots, N_{i_0}$ 时 $a_{i_0, i_j} = \frac{1}{\omega + N_{i_0}}$, 所以 $\Delta_{i_0} = \frac{6N_{i_0} - 5}{6\omega - 1}$.

显然, $\Delta_{i_0} < 1$.

(2) 度 = 3

我们需要求解一个 4×4 的方程组来计算 $d_{i_0}^{(k+1)}$. 通过和上面类似的讨论可得

(3) 度 = 4

$d_{i_0}^{(k+1)}$ 的获得需要求解一个 5×5 的方程组, 同理可得

$$\Delta_{i_0} = 0.5026 < 1.$$

证毕.



DING Jun-Yong, born in 1978, master. His research interests include subdivision surfaces and digital geometry processing.

HU Shi-Min, born in 1968, Ph. D., associate professor. His research interests include computer aided geometric design, computer graphics and human computer interaction.

ZHOU Deng-Wen, born in 1965, associate professor. His research interests include computer graphics and image processing.