软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(5):1097−1105 [doi: 10.3724/SP.J.1001.2011.03750]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于变分网格的曲面简化高效算法*

金 勇, 吴庆标+, 刘利刚

(浙江大学 数学系,浙江 杭州 310027)

Efficient Algorithm for Surface Simplification Based on Variational Mesh

JIN Yong, WU Qing-Biao+, LIU Li-Gang

(Department of Mathematics, Zhejiang University, Hangzhou 310027, China)

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

Jin Y, Wu QL, Liu LG. Efficient algorithm for surface simplification based on variational mesh. *Journal of Software*, 2011,22(5):1097−1105. http://www.jos.org.cn/1000-9825/3750.htm

Abstract: The paper presents a local greedy algorithm that minimizes the energy defined by a variational mesh approximation. The algorithm simplifies the mesh by controlling the number of target polygons, while attempting to gain ideal effect from adaptively selected seed triangles. The algorithm has an intuitive geometric meaning. The algorithm is efficient enough to be efficiently adopted in the geometric modeling system.

Key words: polygon mesh simplification; variational mesh approximation; greedy algorithm; geometric modeling

摘 要: 根据变分网格逼近表示所定义的全局误差能量,提出一种局部贪心优化算法.该算法通过控制目标网格分片数来简化网格,通过种子的自适应选取来达到理想的简化效果,具有直观的几何意义.该方法计算量较小,效率较高,能够有效地应用于几何造型系统中.

关键词: 多边形网格简化;变分网格逼近;贪心算法;几何造型

中图法分类号: TP391 **文献标识码**: A

三维多边形网格模型包括三角形网格、四边形网格等,在计算机辅助几何设计、计算机动画、虚拟现实、计算机游戏和医学影像等领域有着大量的应用[1].随着三维扫描技术的发展,顶点数为数万的模型已经非常常见,有的模型顶点数达到数十万甚至更多.所以,在网格的多分辨率显示、加速网格着色、网格压缩等过程中需要快速的并且能够保持网格模型细节的网格简化算法.

已有的多边形网格简化技术主要分为以下几种:

(1) 删减(decimation):Garland 等人[2]提出一种基于边缩减的网格简化算法.该方法采用网格的顶点到其相关平面的二次距离作为度量,进行边缩减的迭代,直到达到目标网格边数为止.类似地,Hoppe 等人[3]、Klein 等人[4]、Garland 等人[5]对网格元素提出一种误差度量,其误差度量可以基于顶点坐标、颜色或是纹理坐标来建立,然后对网格元素进行迭代缩减以简化网格.删减的方法虽然可以得到有效的网格简化效果,但是该类方法得到的网格往往存在边度数非常高的顶点,并且由于操作网格的拓扑结构而比较耗时.

(2) 网格精炼(mesh refinement):Eck 等人[6]、Delingette 等人[7]、Lee 等人[8]都提出以一个粗网格来逼近表

* 基金项目: 国家自然科学基金(10871178, 60776799); 浙江省重大科技创新项目(2008C01048-3)
收稿时间: 2009-06-23; 定稿时间: 2009-10-10

示原始网格模型,然后以各自的策略迭代地在局部进行精炼加密粗网格,以达到能够准确表示原始网格模型精度的目的.

(3) 网格重构(remeshing):Alliez 等人[9,10]、Gu 等人[11]提出了能够控制目标网格顶点数的网格重构方法,但是这些方法都受限制于网格的参数化,包括大量的计算和数值不稳定性.Valette 等人[12,13]通过局部的的贪心算法构造近似的 Centroidal Voronoi Diagrams(CVD),然后构造 CVD 的对偶图作为目标简化网格.由于 CVD 的对偶图是 Delaunay 三角剖分,所以该方法简化的网格有非常理想的质量,并且该方法不需要对网格进行参数化操作.

(4) 全局优化(global optimization):Hoppe 等人[14]提出将网格简化问题视作全局优化问题.以一个能量函数来度量原始网格,同时通过控制网格的顶点数、顶点坐标和拓扑连接关系以优化定义的能量函数,可以得到保持原始网格模型细节和曲率简化效果.这种方法被推广到同时使用与图像有关的度量来设计能量函数,能量函数不只是和模型的几何性质有关(Lindstorm[15]).

Cohen-Steiner 等人[16]提出了变分网格逼近方法.该方法将原始网格分为目标数量的近似平面簇集,使用与法向有关的能量来度量平面簇集,以 Lloyd 算法来得到优化的网格分簇.最后,每一片分簇以一个多边形表示以得到最终的简化网格.该方法直观、有效,简化后的网格模型能够有效保持原始网格的细节,且该算法由于不涉及网格拓扑的修改,使得算法简洁并且具有较好的稳定性.

Cohen-Steiner 等人[16]提出的变分网格逼近方法有比较理想的视觉效果,然而速度局限于 Lloyd 算法的迭代次数.Lloyd 算法也无法保证得到一个全局最优的结果,并且需要后期处理(合并法向几乎一致的平面簇集)以达到理想的效果.本文主要优化一个与各近似平面簇集的特征法向相关的全局能量,通过离散该全局能量,使得能够使用局部的贪心算法优化该能量;通过初值种子点的自适应选取,可以得到比 Lloyd 算法更好的效果.本文算法在时间上优于 Cohen-Steiner 等人[16]使用的 Lloyd 算法,并且不需要后期的近似平面簇集合并处理.

1 变分网格逼近问题

由于三角形网格的广泛应用性,本文基于三角形网格表述算法.该算法可以非常容易地推广到多边形网格.本节约定一些术语并且介绍变分网格逼近的概念.

三角形网格 M 可以表示为 $\{V,E,F\}$,其中,V 为网格顶点的集合,E 为网格边的集合,F 为网格面的集合:

$$\begin{cases} V=\{v_i=(x_i,y_i,z_i)\in R^3 \mid 1\leqslant i\leqslant N_V\} \\ E=\{e_i=\{v_{i_1},v_{i_2}\},i=1,...,N_E\} \\ F=\{f_i=\{v_{i_1},v_{i_2},v_{i_3}\},i=1,...,N_F\} \end{cases} \tag{1}$$

变分网格逼近理论处理将原始网格 M 的网格面片 F 划分为边连通的近似平面簇集 $C=\{C_i\subset F,i=1,\ldots,N_C\}$,$N_C<<N_F$ 的问题.其中,$\bigcup_{1,...,N_C} C_i=F$,$C_i\cap C_j=\varnothing,\forall i,j.N_C$ 用以控制网格的简化比例,每一簇 C_i 中的原始网格面 $f\in C_i$ 相互关于边连通.当分簇完成后,每一个簇集可以用一个多边形表示,进而可以得到一个逼近原始网格的多边形简化网格.图 1 表明了变分网格逼近的流程.

由上述流程可知,如何选择网格分簇的策略是变分网格逼近的关键.Cohen-Steiner 等人[16]提出,使用与法向相关的全局能量作为网格分簇策略中所优化的能量,使得简化后的网格能够更好地逼近原始网格.

定义每一片簇集 C_i 的特征法向量 n_{C_i} 为该簇集最近似的平面的法向,可如下计算:

$$n_{C_i}=Unif\left(\int_{v\in C_i} n(v)\mathrm{d}v\right) \tag{2}$$

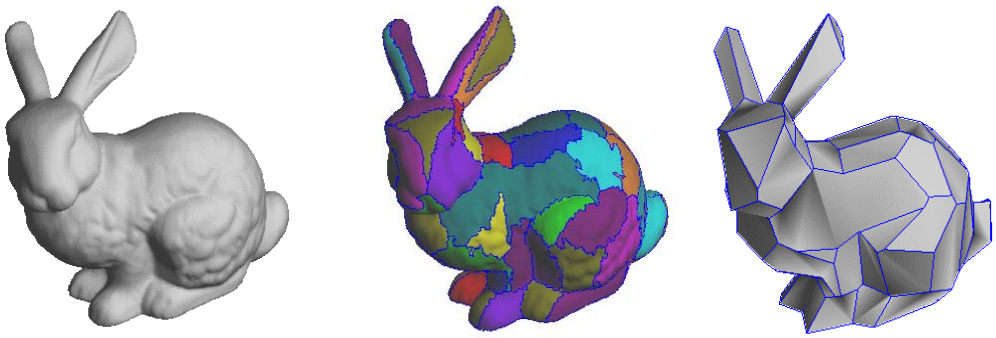
其中,$n(v)$为 v 点的单位法向量,$Unif(\cdot)$指对向量的单位化操作运算.

希望给出一种分簇策略,使得每一个簇集中的所有点与其所对应的簇集特征向量最接近.这样,各个簇集都各自最接近于平面.

给定网格 $M=\{V,E,F\}$、目标分簇数 N_C,在对网格的分簇过程中,希望最小化如下度量:

$$\varepsilon(F,C)=\int_{v\in F}\|n(v)-n_{C_i(v)}\|^2\,\mathrm{d}v \tag{3}$$

其中,$n(v)$为 v 点的单位法向量,$C_i(v)$为 v 所属于的簇集,$n_{C_i(v)}$ 为公式(2)定义的特征法向量.



(a) Original model of bunny, 69 451 triangles (b) Be partioned into 100 clusters (c) Simplified polygonal mesh
(a) 原始 bunny 模型,有 69 451 个三角片 (b) 模型分为 100 个簇集 (c) 简化后的多边形网格

Fig.1 Flow chart of variational mesh approximation
图 1 变分网格逼近流程图

观察公式(3),特征法向量 n_{C_i} 代表与该簇集最近似的平面的法向,将每一点的单位法向量与其所属的簇集的特征法向量的方差的积分作为误差量,误差量越小,说明每一个簇集中的所有点与其所对应的簇集特征向量越接近,亦即每一个簇集越将法向相近的网格三角片归在一起.从图形的绘制上来分析,同一个簇集中的网格片拥有相近法向则用有着相近的光照效果,有利于用一个多边形来表示一个簇集.

2 算 法

给定网格 $M=\{V,E,F\}$、目标分簇数 N_C、误差度量$\varepsilon(F,C)$,希望找到一个最优的划分 C_{opt},使得$\varepsilon(F,C_{opt})=\min\{\varepsilon(F,C)\}$,其中,$C$ 为该目标分簇数所对应的所有可能的划分.虽然公式(2)、公式(3)给出的簇集特征法向量和误差度量形式为连续积分形式,但是最终希望得到的还是对于离散的三角形网格 M 的划分.因此,首先给出特征向量 n_{C_i} 和误差度量$\varepsilon(F,C)$的离散形式,然后给出详细的分簇算法.

2.1 误差度量的离散和简化

注意到每个三角片 $f\in F$ 的法向量可以代表该三角片中所有点的法向量,所以各簇集的单位特征法向量 n_{C_i} 可以离散表示为

$$n_{C_i}=Unif\left(\int_{v\in C_i}n(v)\mathrm{d}v\right)=Unif\left(\sum_{T_j\in C_i}\rho_j n_j\right)=\frac{\sum_{T_j\in C_i}\rho_j n_j}{\|\sum_{T_j\in C_i}\rho_j n_j\|} \tag{4}$$

其中,T_j 为属于簇集 C_i 的所有三角片,n_j 为该三角片的单位法向量,ρ_j 为该三角片的面积.

由此,误差度量$\varepsilon(F,C)$可以离散为

$$\varepsilon(F,C)=\int_{v\in F}\|n(v)-n_{C_i(v)}\|^2\,\mathrm{d}v=\varepsilon_d(F,C)=\sum_{i=0}^{N_C-1}\left(\sum_{T_j\in C_i}\rho_j\left\|n_j-\frac{\sum_{T_j\in C_i}\rho_j n_j}{\|\sum_{T_j\in C_i}\rho_j n_j\|}\right\|^2\right) \tag{5}$$

其中,$\varepsilon_d(F,C)$为$\varepsilon(F,C)$的离散形式.我们希望设计一种基于迭代更新各个簇集边界以优化误差度量的算法(详见第 2.2 节),在每一步的迭代过程中,我们需要考虑簇控制集边界变化以减小误差度量$\varepsilon_d(F,C)$的可能性.为此,需要

知道误差度量$\varepsilon_d(F,C)$在簇集边界变化前后的值,如果以公式(5)计算$\varepsilon_d(F,C)$,其复杂度为 $O(M+N)$(M 和 N 为当前簇集边界相邻的两个簇集的三角片个数,这两个簇集对于误差度量的贡献由于特征法向量的变化而会发生变化,需要重新计算),显然时间开销巨大.为此,分析公式(5)得到:

$$\begin{aligned}\varepsilon_d(F,C)&=\sum_{i=0}^{N_C-1}\left(\sum_{T_j\in C_i}\rho_j\frac{\left\|\|\sum_{T_j\in C_i}\rho_j n_j\|\cdot n_j-\sum_{T_j\in C_i}\rho_j n_j\right\|^2}{\|\sum_{T_j\in C_i}\rho_j n_j\|^2}\right)\\&=\sum_{i=0}^{N_C-1}\left(\frac{\sum_{T_j\in C_i}\rho_j\|\sum_{T_j\in C_i}\rho_j n_j\|^2\|n_j\|^2-2\|\sum_{T_j\in C_i}\rho_j n_j\|\sum_{T_j\in C_i}\rho_j n_j\cdot\sum_{T_j\in C_i}\rho_j n_j+\sum_{T_j\in C_i}\rho_j\|\sum_{T_j\in C_i}\rho_j n_j\|^2}{\|\sum_{T_j\in C_i}\rho_j n_j\|^2}\right)\\&=\sum_{i=0}^{N_C-1}\left(\sum_{T_j\in C_i}\rho_j\|n_j\|^2-2\|\sum_{T_j\in C_i}\rho_j n_j\|+\sum_{T_j\in C_i}\rho_j\right)\end{aligned}\tag{6}$$

注意到,$\forall T_j,\|n_j\|=1$,则

$$\varepsilon_d(F,C)=\sum_{i=0}^{N_C-1}\left(2\sum_{T_j\in C_i}\rho_j-2\|\sum_{T_j\in C_i}\rho_j n_j\|\right)\tag{7}$$

我们将可以看到,公式(5)具有直观的几何意义;重要的是,对于每一个簇集 C_i,只需记录一个向量 $N_i=\sum_{T_j\in C_i}\rho_j n_j$ 和一个标量 $S_i=\sum_{T_j\in C_i}\rho_j$,就可以在迭代更新簇集边界时以 $O(1)$复杂度计算出误差能量$\varepsilon_d(F,C)$.

2.2 局部贪心算法

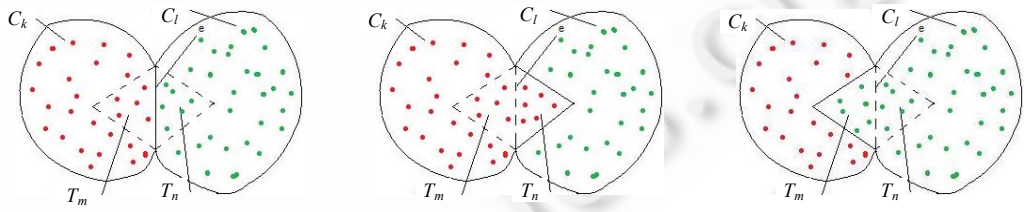
根据公式(7),我们提出一种迭代更新各个簇集 C_i 边界的贪心算法.假设网格边 $e\in E$,e 的两侧三角形为 T_m,T_n,如果 T_m,T_n 属于不同的簇集,那么称 e 是一条簇集边界;所有簇集边界的集合为 E_C,显然,$E_C\subset E$.

每一步迭代中,考虑每一条簇集边 $e\in E_C$ 变更以减小$\varepsilon(F,C)$的可能性:假设 e 的两侧三角形为 T_m,T_n,所属于的簇集分别为 C_k,C_l.如图 2 所示,考虑 3 种变更情况,可以得到 3 个不同的误差度量:

(a) $\varepsilon_d(F,C^{init})$:原始情况,$T_m\in C_k$,$T_n\in C_l$.

(b) $\varepsilon_d(F,C^1)$:将三角片 T_n 并入到簇集 C_k 中,$T_m\in C_k$,$T_n\in C_k$.

(c) $\varepsilon_d(F,C^2)$:将三角片 T_n 并入到簇集 C_k 中,$T_m\in C_l$,$T_n\in C_l$.



(a) Initial: $T_m\in C_k$, $T_n\in C_l$
(a) 初始:$T_m\in C_k$,$T_n\in C_l$

(b) C_k grows, $T_m\in C_k$, $T_n\in C_k$
(b) C_k 增长,$T_m\in C_k$,$T_n\in C_k$

(c) C_l grows, $T_m\in C_l$, $T_n\in C_l$
(c) C_l 增长,$T_m\in C_l$,$T_n\in C_l$

Fig.2 Analysis for modification of local cluster border

图 2 局部簇集边界变更分析

取三者之中误差度量ε_d最小的作为最后的变更结果,然后更新簇集边集合 E_C.对于簇集边进行迭代更新,可以迭代地减小误差度量ε,直到所有簇集边的变更判断结果为 C^{init} 时停止.由于每一步变更都缩小误差度量ε_d,所以该算法的收敛性能够得到保证.为了保证簇集的连通性,在实际迭代计算中,如果簇集边的变更使得破坏簇集

的连通性,则禁止此步变更.

2.3 进一步优化

在实际的计算中,对每一个簇集 C_i,只需记录 $N_i=\sum_{T_j\in C_i}\rho_j n_j$ 即可.

事实上,当考虑 $e\in E_C$ 的变更可能性时,只有其相关的簇集 C_k,C_l 所贡献的误差度量部分可能发生变化.由公式(7),

$$E_{C_k\cup C_l}=2\left(\sum_{T_j\in C_k\cup C_l}\rho_j-\|\sum_{T_j\in C_k}\rho_j n_j\|-\|\sum_{T_j\in C_l}\rho_j n_j\|\right) \tag{8}$$

在第 2.2 节提到的 3 种簇集边 e 的变更情况中,$\sum_{T_j\in C_k\cup C_l}\rho_j$ 为常量,所以只需考虑:

$$L_{C_k\cup C_l}=\|\sum_{T_j\in C_k}\rho_j n_j\|+\|\sum_{T_j\in C_l}\rho_j n_j\|=\frac{1}{2}\left(E_{C_k\cup C_l}-\sum_{T_j\in C_k\cup C_l}\rho_j\right) \tag{9}$$

分别计算 C^{init},C^1,C^2 所对应的 L_{init},L_1,L_2 的值,取 L 最大的变更情况(即对应误差度量最小)作为最后的变更结果.L 越大,对应公式(7)中所对应的误差度量$\varepsilon_d(F,C)$越小.观察公式(9),L 为各簇集三角片的以面积为权的法向量和的模.由向量和的性质:一簇相同模长的向量,方向越是相同,其向量和越大.所以,也可以由此得到优化公式(7)即公式(9)的几何意义:将具有相近法向量的三角片归于同一个簇集,从而同一簇集的三角片近似属于某一特征平面,这也正是分簇目的所在.

2.4 多边形化和三角化

在得到最终的优化簇集划分后,可以将 3 个簇集以上的交点视为多边形网格顶点,即可得到一个多边形网格模型.注意,此处多边形不一定是平面多边形.如上构造出的多边形边可能无法很好地逼近该簇集的边界,由此可以选择性地对多边形网格作一步后处理:各边迭代细分修正.如果 $\overline{AB}$ 为多边形网格的一条边,A,B 为相应多边形网格顶点,遍历边 $\overline{AB}$ 所对应簇集边界上的所有原始网格顶点,寻找出与边 $\overline{AB}$ 距离最远的顶点 P 和相应的距离 d,如果 $d/\|\overline{AB}\|>Th_{\max}$ ($Th_{\max}$ 为给定的容忍极限值),那么将边 $\overline{AB}$ 细分为边 $\overline{AP}$ 和边 $\overline{PB}$.

我们还可以将多边形网格分片为三角形网格,可以使用 Cohen-Steiner 等人[16]提出的多边形边校正方法优化和三角化该多边形网格,或是以经典的多边形三角化方法将简化的多边形网格转换为简化的三角形网格.

3 初始化

第 2.2 节给出了更新簇集边以优化误差度量$\varepsilon(F,C)$的局部贪心算法.要设计一种完整的分簇算法,需要一个初始分簇划分,然后实施第 2.2 节的贪心算法优化误差度量$\varepsilon(F,C)$,直到收敛为止.如果使用随机的簇集初值划分,则不利于算法收敛的速度以及最终的效果.为此,本文提出自适应的种子选取和簇集增长方式.

3.1 自适应的分簇初值

注意到以下几个事实:

(1) 网格曲率越高的部分应该用较多的多边形去逼近表示;相对地,曲率较低的部分应该用较少的多边形去逼近表示.

(2) 法向几乎相同的三角形应该属于同一个簇集.比如,一个平面应该只有一个簇集.

(3) 具有相近法向的三角形在最终分簇更有可能属于同一个簇集.给出一个将相近法向的三角形归在同簇的初始划分,可以加快第 2.2 节簇集边更新算法的收敛速度.

使用 Meyer 等人[17]对于二维流形三角网格的高斯曲率的估计,对于网格顶点,其高斯曲率为

$$\kappa_G(v)=\left(2\pi-\sum_{\theta_j\in Neighbour(v)}\theta_j\right)\Big/A_v \tag{10}$$

其中,θ_j为顶点 v 周围的顶角,A_v为顶点 v 的 Voronoi 区域的面积.三角片的高斯曲率$\kappa_G(T_i)$近似取为其三顶点的高斯曲率绝对值和的平均值,用以反映该三角片曲率的高低.

考虑网格分簇问题:网格 $M=\{V,E,F\}$、目标分簇数 N_C.为了拥有高曲率的三角片的簇集将归并较少的三角片,拥有低曲率的三角片的簇集将归并较多的三角片.我们预测理想的分簇情况下,每一个簇集的高斯曲率总值应该接近于网格的平均值:

$$D=\frac{1}{N_C}\sum_{T_i\in F}\rho_i\kappa_G(T_i) \tag{11}$$

3.2 分簇初值的算法

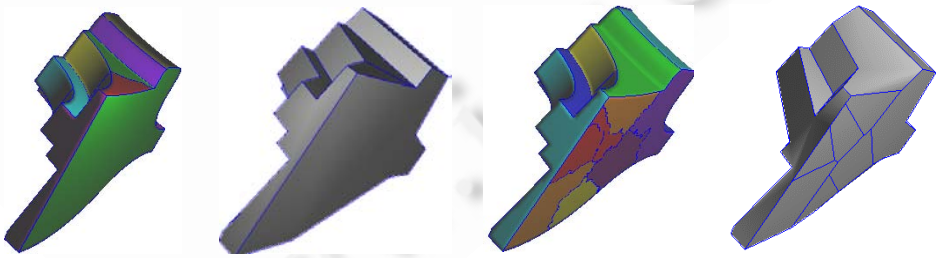
根据第 3.1 节的分析,本节给出分簇初值的算法:

建立每一个簇集 C_i 时:随机取一个还未归属于任何一个簇集的三角片 T_{seed} 作为当前簇集 C_i 的种子三角片,循环地更新该簇集边界,将还未归属于任何簇集并且与该簇集边相连的三角片以 $\left|n(T)-Unif\left(\sum_{T_j\in C_i}\rho_{T_j}n_{T_j}\right)\right|$ 为优先级并入到此簇集中,直到 $\sum_{T_j\in C_i}\rho_j\kappa_G(T_j)>D$ 停止并入三角片.为使几乎属于一个平面的三角片只归入一个簇集当中,当簇集停止并入三角形后,仍然可以将 $\left|n(T)-Unif\left(\sum_{T_j\in C_i}\rho_{T_j}n_{T_j}\right)\right|<dTh$ (dTh 很小)的三角片并入当前簇集中.循环上述建立簇集的方法,直到建立 N_C 个簇集为止.至此得到一个初始簇集划分.实际簇集数 N_{rC} 可能小于 N_C,那么直接将 N_C 改为 N_{rC}.

得到初始的分簇划分后,可以实现第 2.2 节的局部贪心算法:在每一步迭代中,遍历所有的簇集边界 $e\in E_C$,如果 e 相邻的两三角片之一还未归于任何簇集,那么将其归于另一个三角片所属于的簇集;如果两三角片归于不同的簇集,那么执行第 2.2 节中提出的变更判断,更新簇集边界 E_C,直到所有簇集边的变更判断结果为 C^{init} 时停止.

4 实 例

我们将本文算法实验于多种网格模型,达到了较好的视觉效果和较快的速度.与 Cohen-Steiner 等人[16]所提出的 Lloyd 算法相比,视觉效果相似或略好;同时,本文算法比 Lloyd 算法有 3~4 倍的速度优势,且速度优势随模型规模增加而愈加明显.本文同时给出本文算法的例子和一些 Cohen-Steiner 等人[16]算法的例子(Lloyd 算法).图 3 为将 Fandisk 模型以 30 个多边形逼近表示的例子,图 3(a)、图 3(b)为本文算法,图 3(c)、图 3(d)为 Lloyd 算法.



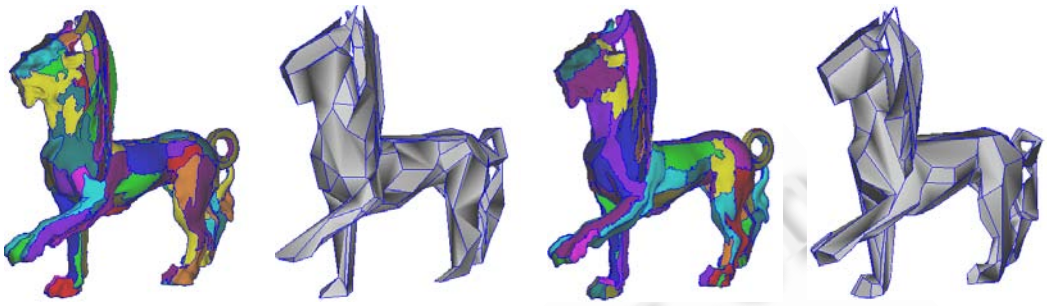
(a) (b) (c) (d)

Fig.3 Fandisk, which has 12 946 triangles, 6 475 vertices, is approximated by 30 polygons

图 3 Fandisk,12 946 个三角形,6 475 个顶点,以 30 个多边形逼近表示

本文算法自适应地将 Fandisk 底面用一个簇集表示,而 Lloyd 算法将底面分为多个簇集,需要后处理合并这

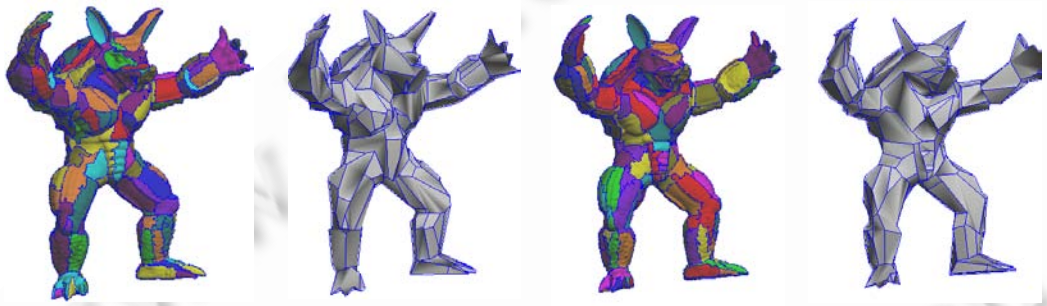
3 个簇集.图 4、图 5 给出了较大模型 Feline 和 Amodilo 的例子,同样地,图 4(a)、图 4(b)、图 5(a)、图 5(b)为本文算法,图 4(c)、图 4(d)、图 5(c)、图 5(d)为 Lloyd 算法.通过表 1,可以看到本文算法在大型模型上的速度与 Lloyd 算法相比具有较大的优势.



(a) (b) (c) (d)

Fig.4 Feline, which has 99 732 triangles, 49 864 vertices, is approximated by 200 polygons

图 4 Feline,99 732 个三角形,49 864 个顶点,以 200 个多边形逼近表示



(a) (b) (c) (d)

Fig.5 Amodilo, which has 331 904 triangles, 165 954 vertices, is approximated by 300 polygons

图 5 Amodilo,331 904 个三角形,165 954 个顶点,以 300 个多边形逼近表示

表 1 列出了所有示例的详细数据,包括收敛速度和$\varepsilon_d(F,C)$(模型为归一化模型).可以看到,本文的算法和度量 Lloyd 算法最终得到的度量误差非常接近;同时,本文算法有较快的速度(本文的计算在一台 CPU:Intel Core 2 Duo T7100,内存:2GB 的 PC 机上完成).

Table 1 Parameters of models and results of algorithm

表 1 模型参数和算法结果

Model	N_F	N_V	N_C	Algorithm	Figure index	Time (ms)	$\varepsilon_d(F,C)$
Bunny	69 451	35 947	100	Our algorithm	Fig.1	596	0.389 8
Fandisk	12 946	6 475	30	Our algorithm	Fig.3(a), Fig.3(b)	76	0.096 8
				Lloyd algorithm	Fig.3(c), Fig.3(d)	181	0.371 2
Feline	99 732	49 864	200	Our algorithm	Fig.4(a), Fig.4(b)	696	0.478 1
				Lloyd algorithm	Fig.4(c), Fig.4(d)	2 357	0.408 2
Amodilo	331 904	165 954	300	Our algorithm	Fig.5(a), Fig.5(b)	2 403	0.409 2
				Lloyd algorithm	Fig.5(c), Fig.5(d)	8 174	0.417 2

5 结束语

本文提出变分网格逼近的一种改进的高效算法,以分簇数作为控制模型简化比例的参数.Lloyd 算法由于受到迭代次数的限制,速度比较慢.本文提出完全不同于 Lloyd 算法的一种能量优化方式,通过离散基于法向的分簇能量,提出自适应的分簇初值选取和局部贪心优化算法,以得到理想的分簇结果,在视觉效果和时间上都要优于 Lloyd 算法.通过大量实验表明,本文的方法易于实现,并且具有直观的几何意义,可以有效应用于几何造型系统中.

在定义了全局误差度量、划分目标及簇集数后,本文的方法和 Lloyd 算法都不能保证得到的全局误差度量最小,其分簇结果取决于分簇初值三角形的选取.本文簇集种子的自适应选取可以有效地减少分簇初值对结果的影响.在实际实验中,不同的分簇初值得到的分簇结果在视觉和误差度量上都很接近.

未来的工作包括研究更有效的分簇度量,比如曲率、测地距离,或者以法向、曲率、欧氏距离的组合作为分簇度量等.

References:

[1] Sun JG, *et al*. Computer Graphics. 3rd ed., Beijing: Tsinghua University Press, 1998 (in Chinese).

[2] Garland M, Heckbert PS. Surface simplification using quadric error metrics. In: Whitted T, ed. Proc. of the ACM SIGGRAPH. Los Angeles: ACM Press, 1997. 209−216. [doi: 10.1145/258734.258849]

[3] Hoppe H. Progressive meshes. In: Rushmeier H, ed. Proc. of the ACM SIGGRAPH. New Orleans: Addison-Wesley Professional, 1996. 99−108. [doi: 10.1145/237170.237216]

[4] Klein R, Liebich G, Straßer W. Mesh reduction with error control. In: Yagel R, Nielson GM, eds. Proc. of the IEEE Visualization. San Francisco: IEEE Computer Society Press, 1996. 311−318. [doi: 10.1109/VISUAL.1996.568124]

[5] Garland M, Heckbert PS. Simplifying surfaces with color and texture using quadric error metrics. In: Ebert DS, Rushmeier H, Hagen H, eds. Proc. of the IEEE Visulaization. Washington: IEEE Computer Society Press, 1998. 263−269. [doi: 10.1109/VISUAL.1998.745312]

[6] Eck M, DeRose T, Duchamp T, Hoppe H, Lounsbery M, Stuetzle W. Multiresolution analysis of arbitrary meshes. In: Mair SG, Cook R, eds. Proc. of the ACM SIGGRAPH. Los Angeles: ACM Press, 1995. 173−182. [doi: 10.1145/218380.218440]

[7] Delingette H, Herbert M, Ikeuchi K. Shape representation and image segmentation using deformable surfaces. Image and Vision Computing, 1992,10(3):132−144. [doi: 10.1016/0262-8856(92)90065-B]

[8] Lee AWF, Sweldens W, Schröder P, Cowsar L, Dobkin D. Maps: Multiresolution adaptive parameterization of surfaces. In: Machover C, ed. Proc. of the ACM SIGGRAPH. Orlando: ACM Press, 1998. 95−104. [doi: 10.1145/280814.280828]

[9] Alliez P, Meyer M, Desbrun M. Interactive geometry remeshing. In: Appolloni T, ed. Proc. of the ACM SIGGRAPH. San Antonio: ACM Press, 2002. 355−361.

[10] Alliez P, Cohen-Steiner D, Devillers O, Levy B, Desbrun M. Anisotropic polygonal remeshing. In: Rockwood AP, ed. Proc. of the ACM SIGGRAPH. San Diego: ACM Press, 2003. 485−493. [doi: 10.1145/882262.882296]

[11] Gu X., Gortler S, Hoppe H. Geometry images. In: Appolloni T, ed. Proc. of the ACM SIGGRAPH. San Antonio: ACM Press, 2002. 363−374. [doi: 10.1145/566654.566589]

[12] Valette S, Chassery JM. Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening. Computer Graphics Forum, 2004,23(3):381−389. [doi: 10.1111/j.1467-8659.2004.00769.x]

[13] Valette S, Kompatsiaris I, Chassery JM. Adaptive polygonal mesh simplification with discrete centroidal voronoi diagrams. In: Lazzari G, Pianesi F, Crowley JL, Kenji M, Oviatt SL, eds. Proc. of the ICMI. Trento: ACM Press, 2005. 655−662.

[14] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Mesh optimization. In: James TK, ed. Proc. of the ACM SIGGRAPH. Anaheim: ACM Press, 1993. 19−26. [doi: 10.1145/166117.166119]

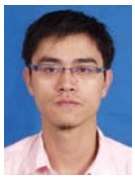
[15] Lindstorm P, Turk G. Image-Driven simplification. ACM Trans. on Graphics, 2000,19(3):204−241. [doi: 10.1145/353981.353995]

[16] Cohen-Steiner D, Alliez P, Desbrun M. Variational shape approximation. In: Marks J, ed. Proc. of the ACM SIGGRAPH. Los Angeles: ACM Press, 2004. 905−914. [doi: 10.1145/1015706.1015817]

[17] Meyer M, Desbrun M, Schröder P, Barr AH. Discrete differential-geometry operators for triangulated 2-manifolds. In: Hege HC, Pothier K, eds. Proc. of the Visualization and Mathematics. Berlin: Springer-Verlag, 2002. 34–57. [doi: 10.1016/j.cagd.2007.07.005]

附中文参考文献:

[1] 孙家广,等.计算机图形学.第 3 版,北京:清华大学出版社,1998.



金勇(1985－),男,上海人,博士生,主要研究领域为数字几何处理,计算机辅助几何设计.

吴庆标(1963－),男,博士,教授,博士生导师,主要研究领域为图形与图像处理,数值计算方法,高性能并行计算,计算机模拟.

刘利刚(1975－),男,博士,教授,博士生导师,主要研究领域为数字几何处理,计算机辅助几何设计,计算机图形学,图像处理.