

基于超包络的三角形网格简化算法*

张明敏¹ 周 昆¹ 潘志庚²

¹(浙江大学 CAD&CG 国家重点实验室 杭州 310027)

²(香港理工大学电子计算学系 香港)

摘要 文章提出了一种基于超包络的三角形网格简化算法. 该算法不仅适用于任意拓扑结构的网格, 而且能定量控制简化的全局误差, 具有速度快、效果好的优点. 在此算法的基础上, 文章提出了一种连续细节层次模型的生成方法, 并给出一组实例, 说明了算法的有效性. 另外, 文章还将此算法与其他具有全局误差控制的简化算法进行了比较.

关键词 三角形网格, 包络, 网格简化, 累进网格.

中图法分类号 TP391

在用三角形网格模型描述复杂物体时, 需要成千上万个三角形才能刻画出复杂物体的细节, 结果导致了庞大的物体模型的出现. 由于绘制时间和存储量与三角形的数目成正比, 因此, 过于庞大的物体模型通常是不实用的. 采用三角形网格简化方法, 能在保持物体基本形状的前提下大大减少三角形的数目, 从而得到简化的物体模型.

Schroeder^[1]提出了基于顶点删除的三角形网格模型简化方法; Turk^[2]给出了基于重新划分的多边形网格模型简化方法; Hoppe^[3]提出了一种整体网格优化算法, 此算法建立了比较复杂的全局能量优化方程; Rossignac^[4]提出了一种基于顶点聚类的网格简化算法; Cohen^[5]提出了包络网格的概念, 用包络网格来约束基于顶点删除的网格简化过程; Garland^[6]使用二次误差测度来控制表面网格的简化, 此算法使用顶点对折叠这一基本简化操作, 可得到高质量的近似模型; Hoppe^[7]在原有算法的基础上提出了基于边折叠方法的、与视点有关的累进网格的生成方法; Guéziec^[8]使用基于容差体 (tolerance volume)、具有全局误差控制的网格简化方法; Bajaj^[9]也提出了一种全局近似误差控制方法; Cohen^[10]在 1998 年的 SIGGRAPH 大会上, 提出了能保持模型外观特性的简化算法. 国内在这方面的研究结果包括对现有算法的改进和把网格简化算法用于多细节层次模型的构造等^[11-12].

一般来说, 网格简化算法可以分为两大类: 一类是最小误差简化算法, 另一类是最小顶点数目算法. 最小误差简化算法其实很难达到最小误差的效果, 因为从顶点数目固定的角度很难找到衡量原始网格与简化网格在形体上差别大小的定量标准, 更不要说找到那个具有最小“误差”的简化网格. 最小顶点数目算法则不一样, 它是先给出一个衡量误差的标准, 在这个标准下尽量删除顶点或多边形, 得到一个在这个标准范围内的具有最少顶点数目的简化网格. 这样, 简化网格逼近原始网格的程度就可通过这个标准来判定. 基于包络网格的多边形网格简化算法^[5]就是这一类算法中较自然、较有效的一种. 该算法的误差衡量标准是简化网格上的所有点离原始网格的距离. 算法的基本思想就是, 用两个包络网格将原始网格面包围起来, 然后在这两个包络网格之间进行网格简化. 对于用户给出的距离误差 ϵ , 构造一个偏离原始网格 $\epsilon/2$ 的外包络网格和一个偏离原始网格 $-\epsilon/2$ 的内包络网格, 对夹在这两个包络网格之间的原始网格进行简化 (删除点或多边形), 生成一个既不与两个包络网格相交

* 本文研究得到国家自然科学基金和浙江大学曹光彪高科技基金资助. 作者张明敏, 女, 1968年生, 讲师, 主要研究领域为多媒体计算, 虚拟现实. 周昆, 1975年生, 硕士生, 主要研究领域为计算机图形. 潘志庚, 1965年生, 博士, 研究员, 博士生导师, 主要研究领域为虚拟现实, 多媒体计算技术.

本文通讯联系人: 潘志庚, 杭州 310027, 浙江大学 CAD&CG 国家重点实验室

本文 1998-01-08 收到原稿, 1998-07-14 收到修改稿

也不自交的简化网格。

Cohen^[5]提出的算法能在保证简化网格与原始网格之间的误差不超过一定范围的前提下,大大删除原始网格上的顶点。另外,它可以对网格进行自适应简化,有效地保持了特征边。然而,该算法的缺点也是很明显的:(1)算法不适用于具有自交现象的网格。算法在检测到原始网格存在自交时就不再处理下去了,而在实际应用中,自交的网格模型是普遍存在的;(2)包络网格构造复杂,计算量大。正是由于算法要通过判断包络网格是否自交来调整每个顶点的偏移量,因此算法的计算量比较大。

基于上述分析,本文提出了一种基于超包络的三角形网格简化算法。这里,超包络的概念是包络网格的拓展,它可以不构成网格,而仅由一些包围在原始网格周围(外部或内部)的三角面片组成。算法具有 Cohen 算法的所有优点,并且可以处理任意拓扑结构的网格模型,允许网格中的三角形自交。包络的构造也非常简单,计算量小。另外,我们还提出了一种基于顶点删除的累进网格的构造方法,从而可以支持多个 LOD(level of detail)模型之间的连续过渡。本文第 1 节描述算法。第 2 节介绍基于顶点删除的累进网格的构造。第 3 节先给出算法实现和实验结果,然后把我们的算法与现有的具有全局误差控制的简化算法进行比较。最后是讨论。

1 算法描述

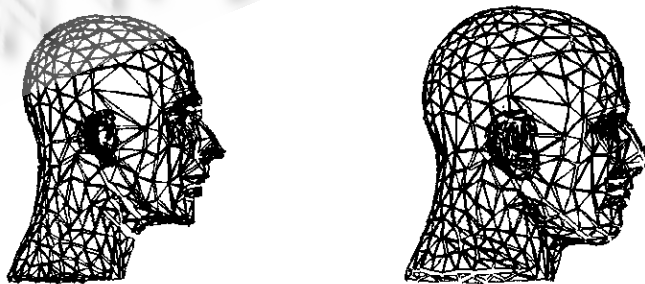
1.1 包络网格概念的推广及超包络的构造

通过仔细分析,我们不难发现,简化算法并不需要利用包络网格的网格性,它仅仅是利用了包络网格的空间位置以使原始网格不致被过分地简化。以顶点删除算法为例:算法先试着删除原始网格上的一个顶点,对删除该顶点后形成的多边形区域局部三角化;然后,对每个由局部三角化得到的新三角形,判断其是否与包络网格相交,如果没有相交,则说明该顶点可以被删除,否则不能被删除。因此,我们对包络网格的概念进行推广,提出了超包络的概念。超包络不一定是网格,它可以是仅仅用来包围原始网格的一些独立的三角形集合。

基于这个分析,我们可以有两种方法来构造超包络。一种方法是,沿法向量方向偏移原始网格的顶点。我们并不要求偏移后得到的超包络是网格,它可以是自交的。另一种方法是,让原始网格的每个三角形沿其法向量方向偏移,这样得到的超包络其实是空间中一些离散三角形的集合。

为了简化超包络的构造,我们让原始网格的每个顶点或三角形偏移相同的距离,即只利用超包络来控制简化的全局误差;在对原始网格进行简化时,我们再考虑具体的几何元素(顶点或三角形)的局部特征。以顶点删除算法为例,我们可以计算每个顶点的曲率,优先删除曲率小的顶点,而曲率则代表了原始网格的局部特征。这样,本文提出的算法既考虑了原始网格的全局特征,又考虑了原始网格的局部特征。另外,由于我们对超包络几乎没有任何限制,此算法可适用于存在自交的网格模型。

图 1 是我们在推广了包络网格的概念后,对人头网格数据做的一个示例。图 1(a)和(b)分别是对应于网格的内超包络和外超包络。该超包络是通过让原始网格上的所有三角形沿其法向量偏移绝对距离 10 后得到的。从图中可以看出,内外超包络由空间一组三角形组成,而并未连成网格。



(a) 内超包络(偏移量-10)

(b) 外超包络(偏移量+10)

图 1 三角形偏移得到的超包络示例

当然,在包络网格的概念推广成超包络后,生成超包络的算法就可以不用检测超包络是否自交,而直接让原

始网格的每个顶点或三角形偏移用户指定的偏移量.但在我们的算法实现中仍包含了检测超包络是否自交这一步,原因是用户指定的偏移量具有很大的盲目性,检测超包络是否自交能提供一些有价值的信息给用户.算法根据用户的需要来决定是否对原始网格模型进行简化.

1.2 简化网格的生成

为了保证得到的简化网格与原始网格的距离误差不超过用户指定的大小,就必须使简化网格夹在内、外超包络之间.在这个前提下可以选择多种简化算法,可以是删点^[1]、删三角形或者是边折叠^[6].以删除顶点的算法为例,既可以使用局部误差又可以使用全局误差.下面给出一个基于曲率的局部删除顶点简化算法.

算法. 基于曲率的局部删除顶点简化算法.

步骤 1: 对原始网格中的每一顶点求出其曲率.

步骤 2: 从没被试探过的所有顶点中选一个曲率最小的顶点,删除该顶点.

步骤 3: 对上一步删除顶点形成的顶点环构成的空间多边形进行三角化.

步骤 4: 对上一步三角化得到的每个新三角形,判断其是否与超包络相交.如果相交,则该顶点不能被删除,恢复该顶点;否则,该顶点被删除.

步骤 5: 若所有顶点都被试探过了,则转步骤 6; 否则,转步骤 2.

步骤 6: 结束.

步骤 1 曲率的计算可以参考文献[12]的方法.如图 2 所示,设简单顶点 Q , \vec{n} 为 Q 点的平均平面的法向量, P_1, P_2, \dots, P_k 为 Q 点周围的顶点环中的所有顶点, \vec{X}_i 为从 Q 点到 P_i 点的向量($i=1, 2, \dots, k$), θ_i 为 \vec{n} 与 \vec{X}_i 的夹角,则 Q 点的曲率可以如下计算.

$$curvature = \frac{1}{\left| \operatorname{tg} \frac{\theta}{2} \right|} = \frac{1}{\left| \operatorname{tg} \frac{\sum_{i=1}^k \theta_i}{k} \right|} \quad (1)$$

这样,我们就得到了在指定误差范围内具有相对较少顶点数目的简化网格.当然,也可以进行一些优化工作(例如,避免尖长三角形的出现),前提是在超包络之间进行.

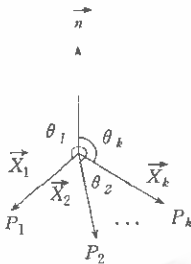


图2 简单顶点曲率的计算

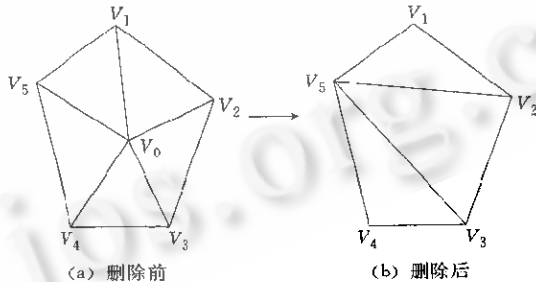


图3 顶点删除示意图

1.3 连续 LOD 模型的生成

显然,上述算法可以用于离散 LOD 模型的生成.为了支持 LOD 模型间的平滑过渡,有必要生成连续 LOD 模型.而累进网格就是一种构造连续 LOD 模型的有效方法.这种方法不仅能大大节省存储量,而且由算法对原始模型作一次处理就可生成.另外,随着网络的飞速发展,累进网格作为对三维图形数据的一种有效的增量传送方法,也具有广阔的应用前景.

累进网格的一种构造方法是基于边折叠^[7].对一个原始网格模型,每执行一次边折叠操作就得到一个简化模型,把这个边折叠操作记录下来.当算法得到原始网格模型的最简表示时,也得到了一个边折叠操作序列,以这个序列的逆序对最简网格模型执行边折叠操作的逆操作(顶点分裂),就可以得到原始网格模型和最简网格模型之间的任意层次的网格模型.

我们提出的方法是以顶点删除为基础的.如图 3 所示, V_0 是要被删除的顶点, V_1, V_2, V_3, V_4 和 V_5 是 V_0 周围的顶点环.删除 V_0 后,对形成的多边形区域三角化就得到了图 3(b)所示的 3 个三角形.我们把 V_0 的坐标和它在

顶点表中的序号以及它周围的顶点环(或半环)记录下来,就不难从图 3(b)得到图 3(a). 对于一个包含 n 个顶点的初始网格模型 M_n , 每删除一个顶点,我们就把与该顶点相关的信息 d_i 记录下来,这样,当得到具有 m 个顶点的简化模型 M_m 时,也就得到了一个信息序列 $\{d_n, d_{n-1}, \dots, d_{m+1}\}$, 由该信息序列 $\{d_n, d_{n-1}, \dots, d_{m+1}\}$, 我们就不难从 M_m 得到模型序列 $\{M_n, M_{n-1}, \dots, M_{m+1}\}$.

2 算法实现、实验结果和算法比较

2.1 算法实现

本文提出的算法已在 SUN SPARC 工作站上用 C 语言实现. 算法实现中最重要的数据结构是八叉树. 由于简化算法每删除一个顶点就要判断新生成的三角形会不会和超包络相交,所以对超包络建立八叉树的数据结构就不用扫描整个超包络,从而大大加快了算法的速度. 本文算法所用到的数据结构有:(a) 顶点表;(b) 三角形表;(c) 与每个顶点相关的三角形组成的表;(d) 超包络的八叉树.

2.2 实验结果

实践表明,该算法对较光滑和不光滑的模型数据都是比较有效的. 如图 4 所示,兔子的模型在简化 95% 之后与原模型还很接近,当简化率为 99% 时,所得到的简化模型还是可以接受的. 对图 5 中假山的网格模型而言,该算法可使大部分山峰和山谷得到保留,因而基本上保持了原网格的特征. 另外,该算法也特别适用于三角化的机械零件模型,如图 6 所示. 图 6(a) 给出的是一个三角化的零件的原始网格模型,图 6(b), (c), (d) 分别是其简化 39%、72% 和 81% 以后的简化模型. 可以看到,零件的直棱边特征得到了很好的保持.

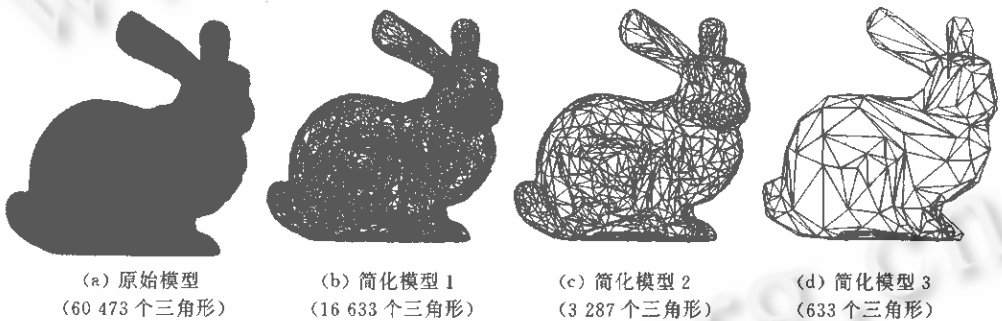


图 4 算法对兔子模型的简化

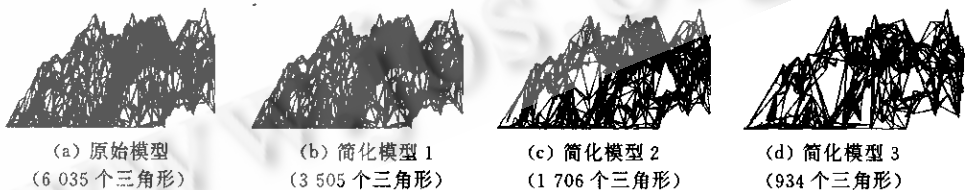


图 5 算法对假山模型的简化

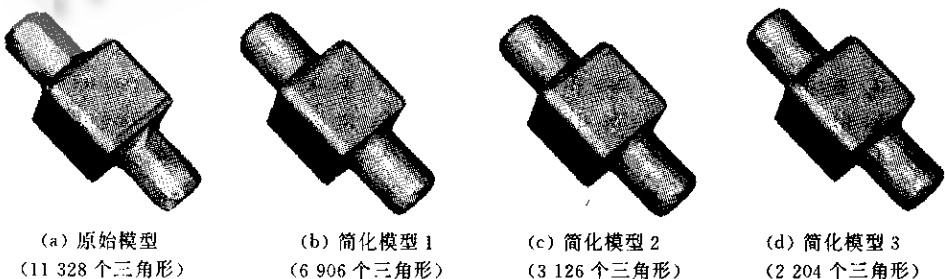


图 6 算法对机械零件模型的简化

3 讨论

本文提出的基于超包络的网格简化算法速度快且效果好,其另一个重要特性是可以与视点结合起来,应用在三维交互式图形系统中.我们可以对模型的不同部分采取不同的偏移量.例如,离视点较远的部分需要刻画得详细一些,离视点较远的部分则可以刻画得粗糙一些;在具体实现中,可考虑对离视点较远的部分采取较小的偏移量,离视点较远的部分采取较大的偏移量.这样,采取较小偏移量的部分就简化得少一些,而采取较大偏移量的部分就简化得多一些.进一步的研究工作是,把该算法与视点和绘制算法结合起来,将该算法应用到具体的三维交互式图形系统(如交互式可视化和虚拟现实)中去.

参考文献

- Schroeder W J, Zarge J A *et al.* Decimation of triangle meshes. *Computer Graphics*, 1992,26(2):65~70
- Turk G. Re tiling polygonal surface. *Computer Graphics*, 1992,26(2):55~64
- Hoppe H, DeRose T *et al.* Mesh optimization. *Computer Graphics*, 1993,27(SIGGRAPH'93):19~26
- Rossignac J, Borrel P. Multi-resolution 3D approximation for rendering complex scenes. In: Falcidieno B, Kunii T eds. *Geometric Modeling in Computer Graphics*. Springer Verlag, 1993. 455~465
- Cohen J, Varshney A, Manocha D *et al.* Simplification envelopes. *Computer Graphics*, 1996,30(SIGGRAPH'96):119~128
- Garland M, Heckbert P S. Surface simplification using quadric error metrics. *Computer Graphics*, 1997,31(SIGGRAPH'97):209~216
- Hoppe H. View-dependent progressive meshes. *Computer Graphics*, 1997,31(SIGGRAPH'97):189~198
- Technical Report, RC20440. IBM Research Division, T. J Watson Research Center, 1995
- Bajaj C L, Schikore D R. Error-bounded resolution of triangle mesh with multivariate data. In: Insethuraman P ed. *Proceedings of SPIE on Electronic Imaging*, SPIE, 1996,2656:34~45
- Cohen J, Olano M, Manocha D *et al.* Appearance-preserving simplification. *Computer Graphics*, 1998,32(SIGGRAPH'98):115~122
- 陶志良,潘志庚,石教英.基于能量评估的网格简化算法及其应用. *软件学报*,1997,8(12):881~888
(Tao Zhi-liang, Pan Zhi-geng, Shi Jiao-ying. Mesh simplification algorithm based on energy evaluation and its application. *Journal of Software*, 1997,8(12):881~888)
- 潘志庚,马小虎,石教英.虚拟环境中多细节层次模型自动生成算法. *软件学报*,1996,7(9):526~531
(Pan Zhi-geng, Ma Xiao-hu, Shi Jiao-ying. The generation generation algorithm for multiple levels of detail in virtual environment. *Journal of Software*, 1996,7(9):526~531)

Triangle Mesh Simplification Algorithm Based on Super-envelopes

ZHANG Ming-min¹ ZHOU Kun¹ *PAN Zhi-geng²

¹(State Key Laboratory of CAD & CG Zhejiang University Hangzhou 310027)

²(Department of Computing Hongkong Polytechnic University Hongkong)

Abstract An algorithm for triangle mesh simplification based on super-envelopes is presented in this paper. This algorithm can not only deal with arbitrary meshes, but also control the global error bound of the simplification. It runs fast and the result is perfect. The authors also give a method for constructing continuous level of detail models. Examples illustrate efficiency of the algorithm. Finally, the authors compare the algorithm with those algorithms with global error control.

Key words Triangle mesh, envelope, mesh simplification, progressive meshes.