

基于法向的网格简化

蒋遂平 周明天 戴颖

(电子科技大学计算机科学与工程学院 成都 610054)

摘 要 网格简化是科学计算可视化和虚拟现实领域中的一个重要研究方面. 它在不严重损失物体的视觉特征的前提下, 用较少数目的多边形来表示物体, 减少了模型的复杂性, 提高了绘制的速度. 文中提出了一个适合于表面曲率变化不太大的网格简化算法, 它根据顶点的法向, 采用边收缩机制来收缩处于共面区域内的边和合并共线的特征边. 简化中既保留模型的特征点, 还能将顶点和面的法向变化控制在用户规定的阈值下, 以保留物体的视觉特征.

关键词 网格简化, 表面简化, 细节层次, 虚拟现实.

分类号 TP391

MESH SIMPLIFICATION BASED ON NORMALS

JIANG Sui-Ping ZHOU Ming-Tian DAI Ying

(College of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054)

Abstract Mesh simplification is a very important aspect in the virtual reality and scientific visualization. It uses fewer polygons to represent the model of object without significant loss in the visual characteristics of the object, thus reducing greatly the complexity of the model and achieving higher rendering speed. This paper presents a mesh simplification algorithm suitable for surfaces with lower curvature variation, which using edge collapse to delete edges in coplanar region and merge collinear edges according to the normals of the end vertices of edges. This paper also presents an error metrics based on normal and uses it to guide the simplification.

Keywords Mesh simplification, surface simplification, level of detail, virtual reality.

1 引 言

在计算机动画、科学计算可视化和虚拟现实等交互式计算机图形学的应用中, 物体的几何模型常采用多边形网格表示, 以便图形硬件能直接处理. 随着物体模型复杂性的增加, 表示物体之所需多边形数目远超过图形硬件以交互帧更新率进行绘制的能力. 一个自然的解决方法是在不严重损失物体视觉特征的前提下, 对物体的网格模型进行简化, 用较少数目的多边形来表示物体. 物体模型的不同简化版本构成物体的细节层次(Levels of detail)或多分辨

率表示.

人们已提出了多种网格简化算法, 这些简化算法按采用的简化机制大致可分为4类:

(1) 抽样(Sampling). 这类算法抽取模型上的特征点, 试图创建一个与抽样点紧密匹配的简化网格, 如文献[1, 2].

(2) 自适应细分(Adaptive subdivision). 这类算法首先构造一个简单的初始模型, 然后进行细分, 直到与原始模型的误差低于指定阈值, 如文献[3].

(3) 删除(Decimation). 这类算法重复地删除网格中的顶点或多边形, 并对形成的洞三角化, 如文献[4—10].

本文1998-11-17收到, 修改文1999-04-14收到. 蒋遂平, 男, 1966年生, 博士研究生, 主要研究方向为虚拟现实、计算机图形学、计算机网络和数据库. 周明天, 男, 1939年生, 教授, 博士生导师, 主要研究方向为计算机网络、分布对象技术、分布并行处理和虚拟现实. 戴颖, 女, 1974年生, 硕士, 主要研究方向为计算机图形学.

(4) 顶点合并(Vertex merging),如文献[11—19],这类算法的基本思想是,投影到图像空间中足够小的区域内的一组顶点可以用一个代表性顶点来取代,形成多到一的映射.合并形成的新顶点又可参与下一轮合并,因此,顶点合并容易形成层次结构,用在依赖于视点(View-dependent)的网格简化中.由于顶点合并的简单性和健壮性,它已成为多边形网格简化的主要机制.

顶点合并有多种方法,如果每次合并一个三角形的3个顶点,就称为三角形收缩或删除,如文献[11,12];如果每次只合并两个共享同一条边的顶点,就叫边收缩(Edge collapse).边收缩方法又可分为:

(1) 一般的边收缩.这类方法将网格中一条边的两个端点收缩到边的某一点上.一个典型方法是Hoppe提出的网格优化(Mesh optimization)^[13],它通过将一个能量函数极小化来计算收缩点的最佳位置.另一个典型方法是由Ronfard等人提出的^[14],该算法采用顶点到相关平面的距离来度量边收缩误差,收缩到误差最小的点.Algorri等人则采用收缩共面区域内的边和共线特征边的方法^[15],将简化限制在局部区域内.国内学者也在这方面进行了卓有成效的研究^[16,17].

(2) 推广的边收缩(Generalized edge collapse).这类方法将网格中距离最近的两个顶点收缩到一个点上,而不管这两个顶点是否有边直接相连;它可能改变物体的拓扑,使物体中原来不直接相连的部分连接在一起,消除了物体上的孔、洞等,该算法也不对原始网格的拓扑作出任何假设,因此健壮性很好.Garland等人采用了顶点到相关平面的二次型距离和作为收缩的误差度量^[18],收缩到误差最小的点上;而Schmalstieg等人的工作^[19]则是简单地收缩到一个代表性顶点上.

简化网格时,要用误差度量来控制简化过程,通常采用的是简化后网格到原始网格的距离^[4—6,18]和保持物体体积^[20]等基于几何的误差度量.我们认为,多边形简化的目的主要是加速绘制,最重要的误差度量应该是基于视觉的而不是基于几何的.由于大多数光照模型需要物体表面的法向,因此,物体的顶点和面的法向变化比物体顶点和面的位置变化对视觉的影响更大.

基于依赖于视点的网格简化框架,我们提出并实现了一个适合于表面曲率变化不太大的网格简化算法,它根据网格中顶点的法向,收缩网格中两个顶点的法向几乎平行的边,以识别和简化网格中的平面区域,识别和合并部分特征边.保留特征顶点,以

保持物体的视觉特征.另外,该算法采用顶点和面的法向变化情况来控制网格简化的程度,使顶点和面的法向变化不超过规定的阈值.

2 简化算法描述

我们的网格简化算法基于边收缩简化机制,其主要步骤为:

(1) 计算各顶点的法向.

(2) 将网格中的边按长度从短到长排序,短边位于队列头部.

(3) 从队列头取出两个顶点的法向夹角小于等于用户规定阈值的边进行收缩.

(4) 重新计算受到边收缩影响的边长度和顶点的法向,调整队列.

(5) 重复步骤(3)、(4),直到没有满足可收缩条件的边为止,完成一次常规简化.

(6) 对于网格中比较短的边,增大阈值,重复步骤(3)—(5),以消除网格模型中的高频成分,完成一次简化循环.

(7) 如果需要,将网格中顶点和面的法向误差清零,重复步骤(1)—(6),进行多次简化循环.

在简化过程中,每收缩一条邻接两个三角形的内部边,网格中的顶点就减少1个,边减少3条,三角形减少2个;每收缩一条邻接一个三角形的边界边,网格中的顶点就减少1个,边减少2条,三角形减少1个.整个算法的运行时间主要用于排序和收缩一条边后调整邻接关系,两者的时间复杂度分别为 $O(\log e)$ 和 $O(e)$,其中, e 是原始网格中边的数目.

2.1 收缩边的选择

网格简化一般需要保留模型中使物体容易辨识的形状特征,物体的主要形状特征包括:

(1) 平面区域.平面区域的多边形应合并成较大的多边形.

(2) 陡边(Sharp edge).陡边是相邻的两个多边形的二面角小于一定阈值的边,共线的陡边可以合并.

(3) 凸点(Protruded vertex).这是曲率较大的顶点,可以将附近的其它顶点合并到这一点上.

在文献[5,6]中,利用顶点法向与相邻三角形的法向夹角来识别平面区域,在文献[8,9]中,通过寻找模型中由法向几乎平行的面组成的平面区域,将平面区域内的多边形合并成较大的多边形后再三角化,从而保留模型中的平面特征.与上述方法不同,我们通过收缩由两个法向几乎平行的顶点组成的边,在网格简化的同时保留模型的平面特征.顶点的法向采用其所在三角形平面的加权平均法向,以排除网格中的噪声干扰.边收缩后,重新计算新三角形

的法向和新顶点的法向.

设顶点 V 所在的三角形的面积和单位法向分别为 A_1, A_2, \dots, A_m 和 N_1, N_2, \dots, N_m , 则顶点 V 的加权平均法向为

$$N = n / |n|,$$

$$\text{其中, } n = \sum_{i=1}^m (N_i \cdot A_i) / \sum A_i.$$

对某一条边 (V_1, V_2) , 如果顶点 V_1 和 V_2 的法向分别为 N_1 和 N_2 , 当 N_1 和 N_2 的夹角 θ 小于等于某个阈值 ϵ (即 $\theta \leq \epsilon$) 时, 就可以收缩边 (V_1, V_2) .

$$\text{因为 } N_1 \cdot N_2 = |N_1| \cdot |N_2| \cdot \cos \theta,$$

$$\text{则 } \cos \theta = (N_1 \cdot N_2) / (|N_1| \cdot |N_2|).$$

如果 N_1 和 N_2 是规格化的, 则有

$$\cos \theta = N_1 \cdot N_2.$$

为计算方便, 在 ϵ 很小时, 可以采用下式:

$$\cos \theta = N_1 \cdot N_2 > 1 - \epsilon.$$

2.2 控制顶点的法向变化

边收缩时, 物体表面的法向发生了变化, 由于法向常用于光照的计算, 因此可能影响用户的视觉. 为计算顶点法向的改变, 我们用被收缩边原来两个顶点的法向中与新顶点法向的夹角最大的那个法向来表示误差法向. 在简化开始时, 顶点误差法向就是该

顶点法向本身. 在一条原始边 (V_1, V_2) 收缩到新顶点 V 后, V, V_1 和 V_2 的规格化法向分别为 N, N_1 和 N_2 . V 的误差法向 N' 应满足

$$N' \cdot N = \max \{ \arccos(N \cdot N_i) \}, \quad i=1, 2,$$

或

$$N' \cdot N = \min \{ N \cdot N_i \}, \quad i=1, 2.$$

如果 V'_1 和 V'_2 是已收缩后的顶点, 我们将它们的误差法向迭加在 V 的误差法向上. 设边 (V'_1, V'_2) 的顶点 V'_1 和 V'_2 的法向分别为 N_1 和 N_2 , 误差法向分别为 N'_1 和 N'_2 , 收缩到新顶点 V , 其法向为 N , 误差法向为 N' , 则 N' 应满足

$$N' \cdot N = \max \{ \arccos(N \cdot N_i) \}, \quad i=1, 2, 1', 2',$$

或

$$N' \cdot N = \min \{ N \cdot N_i \}, \quad i=1, 2, 1', 2'.$$

采用高斯映射 (Gauss map), 将单位法向映射到单位球表面的对应点上, 顶点的法向和它的误差法向就形成了一个法向锥 (Cone of normals), 其中顶点的法向是锥的轴, 误差法向是锥的母线, 这两个法向的夹角为半锥顶角. 这样, 一个顶点的法向锥包含了收缩到这个顶点的所有其它顶点的法向. 边收缩过程中形成的法向锥如图 1 和图 2 所示.

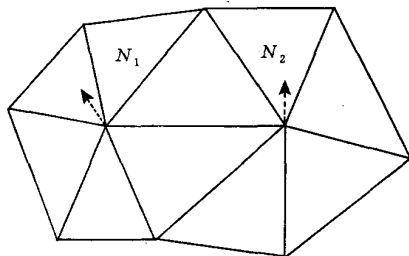


图 1 原始边收缩时形成的法向锥

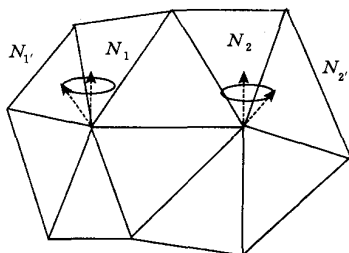
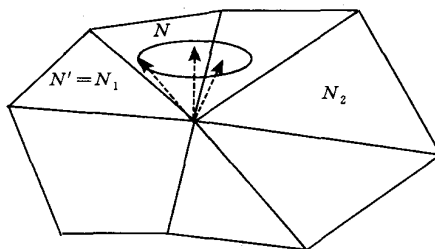


图 2 边收缩时法向锥的合并

边收缩时, 一个重要问题是要确定收缩到边的哪一点上. 对于其中一个端点是边界顶点的边, 我们的算法是让边收缩到边界顶点上; 对于其它边, 则让

边收缩到一点, 使得新顶点的误差法向与新顶点法向的夹角最小, 并用迭代法计算新点的位置. 对于边 (V_1, V_2) , 首先计算分别收缩到中点 $V = (V_1 + V_2) / 2$,

V_1 和 V_2 上的法向锥,即计算 N 和 N' ,然后选择 V_1, V 和 V_2 中法向锥半锥顶角最小的两点进行迭代,直到达到指定的迭代次数,或中点的法向锥半锥顶角不再减小为止,最后选择法向锥半锥顶角最小的点作为收缩的顶点.

在引入误差法向后,边 (V_1, V_2) 收缩到顶点 V 的条件变为

$$\max\{\arccos(N \cdot N_i)\} \leq \epsilon, \quad i=1, 2, 1', 2',$$

在 ϵ 很小时,可以采用

$$\min\{N \cdot N_i\} > 1 - \epsilon, \quad i=1, 2, 1', 2',$$

即将边收缩到新顶点的最大半锥顶角 $\leq \epsilon$, 边 (V_1, V_2) 的两个顶点的法向满足 $\cos\theta = N_1 \cdot N_2$, 即满足 $\theta \leq 2\epsilon$ 后,再进行上述测试.

如果需要保留网格模型中的视觉特征点,即落在模型轮廓上的概率比较大的点,则可以选择边 (V_1, V_2) 的两个顶点之一,而不选用该边上使法向锥顶角最小的点. 顶点 V 落在模型轮廓上的概率 V_p 的计算方法如下^[3]:

$$V_p = L \cdot \cos(\theta/2),$$

其中,

$$L = \max\{|VV_i|, (V, V_i) \text{ 是顶点 } V \text{ 所邻接的边};$$

$$\theta = \max\{\theta_{ij}\}, \theta_{ij} \text{ 是顶点 } V \text{ 邻接边 } (V, V_i) \text{ 和 } (V, V_j) \text{ 所成的角}, i \neq j.$$

V_p 值越大,顶点的视觉特征就越大. 因此对于边 (V_1, V_2) , 可以选择概率 V_p 比较大的顶点,但仍需要其法向锥顶角满足指定的条件,否则,一些顶点的概率值将越来越大,导致网格向少数顶点收缩.

2.3 控制面的法向变化

并非所有满足条件的边都需要收缩,如果包围要收缩边的空间多边形是凹多边形时,收缩这条边可能引起面的法向翻转,造成表面折皱(Fold).

为避免出现折皱,我们在收缩边时进行折皱检测. 其检测方法是判断顶点位置被改变的三角形的法向方向是否逆转,如果逆转,则拒绝收缩这条边.

为控制三角形在简化过程中的法向变化过大,引起物体视觉特征的急剧改变,我们将三角形的面的法向变化也控制在规定的阈值内,使三角形的新法向 N_{new} 与原始网格中的法向 N_{ini} 的夹角也小于阈值 ϵ :

$$\arccos(N_{\text{new}} \cdot N_{\text{ini}}) \leq \epsilon,$$

为计算方便,在 ϵ 很小时,可采用

$$N_{\text{new}} \cdot N_{\text{ini}} > 1 - \epsilon.$$

这样,在收缩边时,只要新顶点的邻接三角形中,有一个三角形的法向变化程度超过规定阈值,就拒绝收缩这条边,避免了折皱的产生.

2.4 进一步简化

上述简化过程对网格模型中的平面区域和特征边进行了简化,消除了模型中的低频成分,但模型中一些表现物体细节的高频成分,如微小的凸块等并未消除. 这些特征在物体远离观察者时,是不明显的,应该消除. 我们的算法对这些特征进行了部分简化,即对模型中长度小于一定阈值的边,将其边收缩到的新顶点的法向半锥顶角放宽. 这样,模型的简化程度得到了进一步提高. 目前,我们对网格中长度小于平均长度的 $1/n$ 的边,允许其收缩后的顶点的法向半锥顶角 $\leq (n+1)\epsilon \leq 1.0$ 弧度. 实验结果表明,采用该方法,可以将顶点的数目再减少 30%—50%.

3 实验结果

我们在微机 (Pentium100MHz, RAM64MB) 上,实现了以上算法,对人头塑像和兔子塑像的网格模型进行了简化. 图 3 是对人头塑像的简化结果,图 3(b)选取边上顶点法向锥顶角最小的点,图 3(c)选取边上视觉特征最大的顶点,两个方法的简化深度相差不大,但前者产生的网格更均匀. 图 4 是对兔子塑像的简化结果,图 4(c)的顶点数目为原顶点数的 2%,结果还可接受,如果进一步简化,将产生较大的误差.



图 3 人头塑像简化结果



图 4 对兔子塑像的简化结果

与我们的算法相似的是 Algorri 等人在文献[15]中提出的算法. 在文献[15]中, 如果一条边所在的两个面的法向夹角大于指定的平面性阈值(Planarity-threshold) θ_p , 这条边就被标记为特征边(Feature edge). 根据每个顶点所邻接的特征边数目进行标记, 需要简化的边就事先静态确定了. 该算法采用了 3 个简化步骤: 首先将连接两个标记同时为“0”顶点的边简单地收缩到中点, 两个顶点标记为“0”的边即平面区域的边; 然后将两条连接同一个标记为“2”的顶点的特征边合并, 要求这两条边的夹角小于共线性阈值(Collinearity-threshold) θ_c (通常 $\theta_p = \theta_c$), 这相当于收缩到某个端点的边收缩; 最后删除被其它顶点包围的“0”顶点, 将形成的洞三角化.

我们的算法通过顶点的法向可一次性地判别出文献[15]中算法(以下简称算法[15])的平面性和共线性条件, 用一个参数取代了算法[15]的两个参数; 而且, 在一次简化循环中, 我们的算法可以加大短边的简化阈值; 此外, 我们的算法对要收缩的边是动态确定的, 并采取了优化措施来确定收缩的位置. 因而可以取得比算法[15]更好的简化效果.

采用文献[21, 22]提出的方法, 将我们的算法和算法[15]对于兔子模型的简化结果进行了比较, 结果见表 1. 其中, 误差定义为原始网格和简化网格的双边 Hausdorff 距离与原始网格的包围盒对角线长度的百分比. 简化程度定义为简化网格顶点数目与原始网格顶点数目的百分比. 简化时间的单位为 s.

表 1 简化结果对比

简化程度	最大误差		平均误差		简化时间	
	本算法	算法[15]	本算法	算法[15]	本算法	算法[15]
50%	0.5222	0.4836	0.0139	0.0186	31	17
25%	0.6452	0.8017	0.0403	0.0567	43	21
10%	1.2851	1.3329	0.1017	0.1509	52	24
5%	2.2656	2.1763	0.2052	0.2919	52	26
2%	4.0855	7.2519	0.3701	0.8428	56	25

从以上测试数据可以看出, 我们的算法在控制简化的几何误差方面, 比算法[15]有了较大的提高, 特别是在简化程度较大时, 效果更明显. 由于算法[15]在对边的顶点进行分类后, 其后的简化过程几乎没有浮点运算, 因此简化速度快; 我们的算法由于在简化过程中需要调整法向, 需要进行一些浮点运算, 简化速度比算法[15]慢, 得到的好处是提高了简化质量.

采用法向来控制网格简化的主要算法有文献[5, 6, 8, 9] (简称算法[5], [6], [8], [9]). 算法[5]和[6]利用顶点的法向与周围三角形法向的最大夹角作为选择删除顶点的准则. 在算法[8]中, 先将法向几乎平行的邻接多边形合并为一个大的多边形, 此

多边形的法向与这些多边形的平均法向的夹角不超过指定阈值 ϵ , 然后再将大多边形三角化. 此算法适合于表面法向变化较小的区域, 不适合于表面曲率变化较大的区域. 在算法[9]中, 法向几乎平行的邻接多边形也被合并为一个大的、称为超面(Super-face)的多边形, 超面中任何两个多边形法向的夹角不超过指定阈值 θ , 调整邻接超面的边界后再将超面三角化, 同时将新顶点到原始网格的距离限制在指定的阈值 ϵ 内. 算法[9]是这类算法中最复杂的一种, 也是最精确的一种. 以上算法采用删除顶点或三角形的简化机制, 很难集成进依赖于视点的简化框架中. 此外, 由于这些算法都需要对删除顶点或三角形后形成的洞三角化, 其过程较复杂, 简化速率较

慢. 我们的算法的简化速率超过了它们. 几种算法的简化速率(千三角形/秒)如表 2 所示.

表 2 几种算法的三角形简化速率(千三角形/秒)

	算法[5]	算法[6]	算法[8]	算法[9]	本算法
硬件	SGI Indigo	SGI Indy			
	R4400/150MHz	R4600/100MHz	不详	RS/6000(550)	Pentium 100MHz
最大速率		0.8	0.90	1.07	1.20
最小速率	0.2		0.70	0.9	1.10

4 结 论

我们提出了一个基于边收缩的网格简化算法, 它采用顶点法向作为边收缩的依据和简化时的误差度量, 法向误差比距离误差或体积误差更直观. 本算法能够识别和简化模型中的平面区域, 合并共线的特征边, 保留网格模型中的特征点, 因此能有效地实现表面曲率变化不大的网格模型的简化, 并保留模型的形状特征. 实验数据表明, 我们的算法在简单性、简化速率和简化质量方面取得了较好的效果. 目前, 我们正就如何将颜色和纹理坐标等物体表面属性集成到简化过程中, 如何更有效地消除网格模型中的高频分量, 以及如何在简化过程中利用模型的拓扑信息等问题, 进行更深入的研究.

参 考 文 献

1 Rossignac J *et al.* Multi-resolution 3D approximations for rendering complex scenes. IBM Research Division, T J Watson Research Center, IBM Technical Report: RC17697, 1993

2 Low K L *et al.* Model simplification using vertex-clustering. In: ACM SIGGRAPH Special Issues on 1997 Symposium on Interactive 3D Graphics, Providence, RI, USA: ACM Press, 1997. 75-81

3 DeHaemer M *et al.* Simplification of objects rendered by polygonal approximations. *Computer & Graphics*, 1991, 15(2):175-184

4 Schroeder W *et al.* Decimation of triangle meshes. *Computer Graphics*, 1992, 26(2):65-70

5 Schroder F *et al.* Managing the complexity of digital terrain models. *Computer & Graphics*, 1994, 18(6):775-783

6 Nakamae E *et al.* Simplification of polygonal surface with attributes. In: Proceedings of ICSC'95, Hong Kong, 1995. 132

-139

7 潘志庚等. 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

8 Hinker P *et al.* Geometric optimization. In: IEEE Proceedings of Visualization, Los Alamitos, 1993. 189-195

9 Kalvin A *et al.* Surfaces: Polygonal mesh simplification with bounded error. *IEEE Computer Graphics & Applications*, 1996, (3):64-77

10 马小虎等. 基于三角形移去准则的多面体模型简化方法. 计算机学报, 1998, 21(6):492-498

11 Hamann B. A data reduction scheme for triangulated surfaces. *Computer Aided Geometric Design*, 1994, 11(3):197-214

12 周 昆等. 基于三角形折叠的网格简化算法. 计算机学报, 1998, 21(6):506-513

13 Hoppe H *et al.* Mesh optimization. In: ACM SIGGRAPH. Anaheim, California, 1993. 19-26

14 Ronfard R *et al.* Full-range approximation of triangulated polyhedra. *Computer Forum*, 1996, 15(3):67-76

15 Algorri M E *et al.* Mesh simplification. *Computer Graphics Forum*, 1996, 15(3):78-86

16 陶志良等. 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

17 李 捷等. 三维复杂模型的实时连续多分辨率绘制. 计算机学报, 1998, 21(6):481-491

18 Garland M *et al.* Surface simplification using quadric error metrics. In: ACM SIGGRAPH, Los Angeles, 1997. 209-216

19 Schmalstieg D *et al.* Smooth of levels of detail. In: IEEE Proceedings of Virtual Reality Annual International Symposium, Los Alamitos, 1997. 12-19

20 Gueziec A. Surface simplification inside a tolerance volume. IBM Research Division, T J Watson Research Center, Technical Report: RC 20440, 1996

21 Cignoni P *et al.* A comparison of mesh simplification algorithms. *Computer & Graphics*, 1998, 22(1):37-54

22 Cignoni P *et al.* Metro: Measuring error on simplified surfaces. In: Eurographics'98, 1998