

渐进网格及其在移动计算中的应用

马建平 罗笑南 凌若天 马识佳

(中山大学计算机应用研究所, 广州 510275)

摘要 在移动计算中,3维图形通常是由几何造型的网格来表示。为了解决移动图形的存储、传输和显示问题,提出了一种基于逆细分的构建渐进网格的算法,给出了渐进网格通过网格传输和在移动终端上渲染3维图形的方法。细密的网格通过逐层地、分批地删除其冗余信息,最后生成由基网格和一系列误差值组成的渐进网格。在算法实施时,将Loop逼近型细分模式作为插值型细分模式进行操作。该算法共分3个关键步骤:网格分裂、奇点预测、网格更新。简化后的渐进网格可以无损还原。实验结果表明,该算法效率高,比以往的方法速度快。

关键词 渐进网格 渐进传输 逆细分 图形渲染

中图法分类号: TP391.41 **文献标识码:** A **文章编号:** 1006-8961(2007)02-0250-06

Progressive Mesh Generation and its Application in Mobile Computing

MA Jian-ping, LUO Xiao-nan, LING Ruo-tian, MA Shi-jia

(Institute of Computer Application, Sun Yat-sen University, Guangzhou 510275)

Abstract In Mobile Graphics Computing, 3D graphics is generally illustrated by meshes of 3D geometric model. In order to overcome the difficulty of storage, transmission and rendering for Mobile Graphics, a novel algorithm to construct progressive meshes based on reverse subdivision is proposed. A method for transmission progressive meshes over wireless network and its rendering on mobile devices is presented. The dense mesh is simplified into coarse base mesh and a series of errors by decimating the redundant information level by level. Loop subdivision scheme, which is an approximating scheme, is adopted as interpolatory one in the simplification process. The implementation of our algorithm consists of three key steps: splitting, predicting and updating. The mesh can be reconstructed lossless. The experimental results show that the algorithm is highly efficient and faster than previous related techniques.

Keywords progressive mesh, progressive transmission, reverse subdivision, graphics rendering

1 引言

随着无线网络和多媒体技术的飞速发展,人们对移动设备上多媒体服务和3维图形的应用需求越来越紧迫。而移动设备只有有限的运算、存储、显示能力,能否利用互联网服务器强大的运算能力和丰富的内容资源,通过无线网络对服务器上的内容进行快速访问以满足用户对信息内容的需求,这是多媒体技术和移动计算所面临的新挑战。目前尚需解

决两方面的问题:一是无线网络传输带宽,二是功能强大的图形处理移动终端^[1]。

3维图形的表示是通过复杂的、高密度的三角网格进行渲染而得到的。选择适当的层次细节模型(LOD),可以控制移动终端的平均负载和平衡网络带宽。Hoppe首先提出了渐进网格^[2]的概念,将三角网格通过边折叠和点分裂表示成多分辨率框架,此后,研究人员在渐进网格的生成和网格简化方面做了大量的研究工作^[3-6],其中文献[3]将任意网格通过二次误差测量(QEM)^[7]进行压缩后,生成

基金项目: 国家杰出青年科学基金项目(60525213);国家自然科学基金重点项目(60533030)

收稿日期: 2006-10-13; **改回日期:** 2006-11-02

第一作者简介: 马建平(1963~),女,高级工程师。中山大学计算机应用研究所计算机软件与理论专业博士研究生。主要研究方向为移动图形计算、3维CAD。长期从事计算机应用系统、CAD系统、信息与通信等方面的产品研发和研究工作。完成科研与工程项目20余项。
E-mail: rpimajp@126.com

控制网格,再将控制网格通过曲面拟合^[8]生成了具有细分连通性的网格;文献[5]提出基于表面曲率变化的简化方法,将法向量相近的两顶点利用边折叠方法删除;文献[6]提出了利用体积优化来确定删除目标点的方法,以控制网格简化的最大误差。

本文提出了一种新的网格简化方法。它是基于逆细分的思想,将网格中的冗余信息分层次地删除;最后形成由一个基网格和一系列误差值组成的渐进网格。该算法适用于具有细分连通性的三角形网络,对于任意网格可通过曲面拟合等方面进行预处理^[8]。算法中将逼近型的 Loop 细分模式^[9]当作插值型对其进行操作,避免了原网格偶点坐标的重新调整。另外,由于 Loop 细分模式的顶点仿射组合^[10]相关联的顶点数目少,使得网格简化和网格还原或重建过程的运算速度相对较快,在实际应用中更为有效。

2 渐进网格编码

2.1 Loop 细分与逆细分

曲面细分是一种有效的曲面几何造型生成模式,通过一组细分规则将粗糙的任意的网格逐渐地加密,而得到光滑平顺的曲面。常用的基于三角网格的细分模式有:Loop 细分^[9]、Butterfly 细分^[11]等。Loop 细分是 1~4 三角形分裂的逼近型模式,它的规则是:在每条边加入一个新顶点生成边点(edge point);原网格顶点坐标根据其相邻顶点的所占的权值做相应的调整生成新的顶点(vertex point)。重复上述操作,在极限情况下,可生成具有 C^2 连续的细分曲面。为了便于说明,定义 edge point 为奇点,vertex point 为偶点。Loop 细分的模板如图 1 所示。

图 1 中奇点 v_e 、偶点 v_v 的坐标计算公式如下:

$$v_e = \frac{3}{8}(v_1 + v_2) + \frac{1}{8}(v_3 + v_4) \quad (1)$$

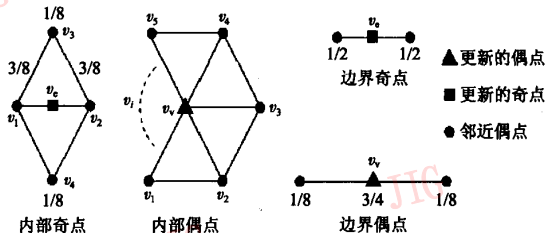


图 1 Loop 细分模板

Fig. 1 Loop subdivision mask

$$v_v = (1 - n\beta_n)v + \beta_n \sum_{i=0}^{n-1} v_i \quad (n \geq 3) \quad (2)$$

其中, v_1, v_2 是内部边的两个顶点, v_3, v_4 是共享此边的两个三角形的另外两个顶点, v 是内部顶点, $v_i (i=0, 1, \dots, n)$ 是与顶点 v 相邻的所有顶点, $\beta_n = \frac{1}{n} \left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4} \cos\left(\frac{2\pi}{n}\right) \right)^2 \right)$ 。对于第 j 层网格,

Loop 细分可以用细分矩阵表示:

$$\begin{bmatrix} V^{j+1} \\ E^{j+1} \end{bmatrix}_{(n+m) \times 1} = M_{(n+m) \times n} [V^j]_{n \times 1} \quad (3)$$

其中, V 和 E 分别表示偶点集和奇点集, M 为细分矩阵。可见, E^{j+1} 对 V^{j+1} 不产生影响。将细分的过程做逆操作,称为逆细分。逆细分可以将密集的三角形网格变成稀疏的网格。也就是在式(3)中,已知 V^{j+1} 和 E^{j+1} ,求 V^j 。此过程是复杂的矩阵方程求解,过于耗费运算时间,故本文采用另一种方法。仔细分析 Loop 细分规则产生了以下想法:

(1) 通过细分生成的奇点只与上一层网格的顶点有关,而这些顶点又是本层的偶点,也就是说,在逆细分中,奇点对偶点坐标位置不产生影响,可以作为冗余信息删除。

(2) 在 Loop 细分时,偶点坐标是根据上一层网格顶点所占的权值计算得到的,而调整偶点坐标的目的是生成光滑的曲面。但是,逆细分操作的前提是假定原图形曲面已经是光滑的,故利用逆细分进行网格简化时,偶点坐标无需调整,可以保持不变。也就是说,逼近型的 Loop 细分在逆细分时可当作插值型处理。

2.2 渐进网格生成原理

首先引入以下定义。在三角网格内部顶点中,与顶点相连边的数量称为该顶点的度。度数为 6 的顶点定义为正则点,度数不等于 6 的顶点定义为非正则点或奇异点;所有顶点都是正则点的网格定义为正则网格,而绝大多数顶点为正则点的三角网格为半正则网格(semi-regular)。对于奇异点过多而造成逆细分效果不理想的模型,可以通过重网格化的方法使网格符合要求。

对于正则网格,先任选一个顶点为偶点。根据上一节分析,如图 2 所示,带圈的顶点都是奇点,其他点为偶点。在逆细分过程中,所有奇点是冗余信息,可以删除,所有偶点保留到下一层。

对于半正则网格,如图 3 所示,由于网格中的奇异点通常表示曲面的尖锐特性,需要全部保留下来,

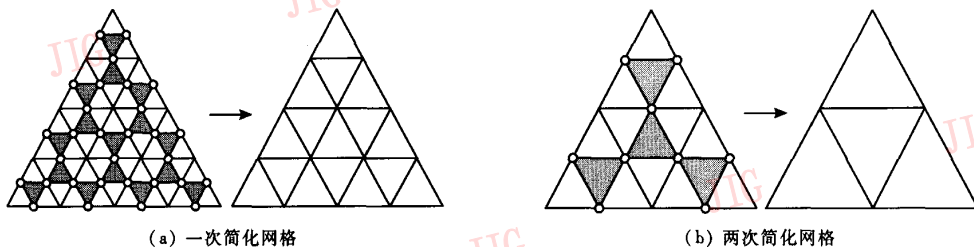


图 2 正则网格简化示意图

Fig. 2 Regular mesh simplification

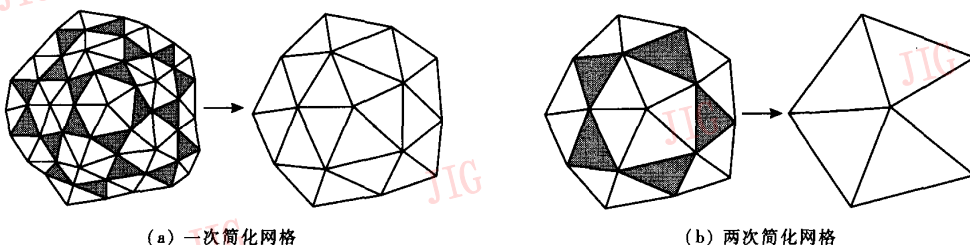


图 3 半正则网格简化示意图

Fig. 3 Irregular mesh simplification

故首先任选网格上一个奇异点为偶点,与此偶点相邻的顶点均为奇点。图中阴影三角形的顶点是网格的奇点,可以删除,其余点为偶点,保留至下一层。

观察图 2、图 3 可以发现,奇点、偶点有规律地分布在三角网格中,每个偶点都被奇点所包围。

2.3 渐进网格生成算法

引用文献[12]的网格表示方法。设三角网格 $M = (P, K)$, 其中 P 表示三角网格顶点坐标的集合, $P_i = (x_i, y_i, z_i) (1 \leq i \leq n)$, K 表示网格拓扑信息集合。本算法的目标就是将原始网格 $M^n = (P^n, K^n)$ 经过多次简化生成简化网格 $M^j = (P^j, K^j) (0 \leq j \leq n)$, 其中 $M^0 = (P^0, K^0)$ 为基网格。算法限定输入的网格是 2 维流形的多面体。对于封闭的 2 维流形曲面没有完全是正则的情况, 网格中至少存在一个奇异点。

设初始曲面网格为 M^n , 经过一次简化后得到 M^{n-1} , 第 i 次简化后的网格为 M^{n-i} 。对于第 j 层网格 M^j , 其简化过程包括以下 3 个步骤。

(1) 网格分裂 将已有的三角网格 M^j 的顶点 P^j 分裂成奇点集 ODD^j 和偶点集 $EVEN^j$ 。

由于封闭的网格中至少有一个奇异点, 故在算法实现时首先选任意奇异点 v 设为偶点集, 然后将与奇异点相邻的所有顶点 va 归为奇点集, 对这些奇点外围与本偶点对称位置的顶点 vs 设为偶点, 归为

偶点集。顶点关系如图 4 所示。

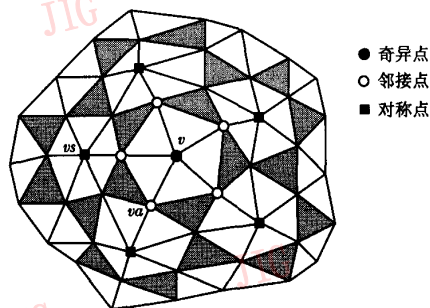


图 4 顶点关系示意图

Fig. 4 Vertices relationship

设置偶点是利用递归调用, 直到网格中所有顶点分裂完毕, 设置偶点的算法如下:

SetEvenVertex(v)

```
{
    如果  $v$  已经在偶点集中, 返回成功;
    如果  $v$  已经在奇点集中, 返回失败;
    设置  $v$  到偶点集 SetEvenVertex( $v$ );
    对于  $v$  的每一个相邻点  $va$ 
```

```
{
    如果  $va$  是奇异点或者  $va$  已经在偶点集中, 返回失败;
    设置  $va$  到奇点集 SetOddVertex( $va$ );
```

找到与 va 和 v 的对称点 vs ;

设置 vs 到偶点集 $SetEvenVertex(vs)$;

返回成功

对于半正则网格,要求三角网格具有细分连通性,也就是满足下式。

$$Distance(P_i, P_j) = 2^k, k = 1, 2, 3, \dots \quad (4)$$

其中, P_i, P_j 表示网格中任意两个奇异点, $Distance(P_i, P_j)$ 表示两奇异点间连接边的个数, $k-1$ 是可进行简化的次数。

(2) 预测过程 为了3维曲面造型的网格还原和重建,在删除奇点之前,采用式(1)预测还原时可能生成每个奇点 i 的位置 ODD_i^j ,将现有的各个奇点 ODD_i^j 与预测值对应相减得到一组误差值 $e_i^j (j=1, 2, \dots, n)$ 。

$$e_i^j = ODD_i^j - ODD_i^j \quad (5)$$

图5(a)中 O 点为奇点, E 点为偶点,图5(b)是删除 O 点前,用细分模式预测得到点 O' ,每个奇点 O 与其相应的预测点 O' 的差值即得到误差值 e^j 集。

(3) 更新网格 删除掉奇点集 ODD^j 后,剩余的偶点集 $EVEN^j$ 形成下一层的顶点 P^{j+1} ,更新这些顶点的连接信息组成新的拓扑 K^{j+1} ,形成了一个简化后的新网格 $M^{j+1} = (P^{j+1}, K^{j+1})$ 。图5(c)示意删除奇点后重建的网格。

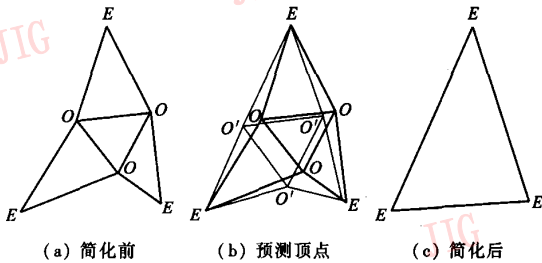


图5 预测与更新示意

Fig. 5 Predict and update process

重复上述3步,可以将复杂的初始网格数据 M^n 简化到网格 M^0 。可以看出,每一层的网格简化,偶点的坐标并不需要改变。简化后得到一个基网格和一系列误差集 e^j ,构成多个层次的数据: $M^0 \rightarrow e^0 \rightarrow e^1 \rightarrow \dots \rightarrow e^{n-2} \rightarrow e^{n-1}$,也就生成了渐进网格。渐进网格 (M^0, e^0) 可重建生成 M^1 , (M^1, e^1) 重建生成 $M^2, \dots, (M^{n-1}, e^{n-1})$ 可重建生成 M^n 。渐进网格表示如表1所示。

表1 由基网格和误差组成的渐进网格

Tab.1 Progressive Mesh with a base mesh and errors

渐进网格	M^n	M^{n-1}	...	M^1	M^0
误差值		e^{n-1}	...	e^1	e^0

3 网格渐进传输和图形渲染

尽管移动终端的性能不断的提高,但在移动环境下还是满足不了3维图形所需要的巨大存储空间的要求。采用移动计算技术,将3维图形的网格进行简化,生成渐进网格并存储在互联网服务器端,当移动用户需要时,通过无线网络,对渐进网格进行渐进传输,并在移动终端上通过图形渲染引擎对网格进行还原并渲染给用户。文献[13]提出了无线网络中渐进网格的传输方法,本文的方法与此方法相似,但在网格还原和重建采用 Loop 细分模板而不是蝶形细分模板,从而加快了图形重建的速度。

渐进网格在网络上的传输将依据网络的带宽分层次依次渐进地进行,渐进传输的格式是:

$$(M^0) \rightarrow (M^0, e^0) \rightarrow (M^0, e^0, e^1) \rightarrow \dots \rightarrow$$

$$(M^0, e^0, e^1, \dots, e^{n-2}, e^{n-1})$$

渐进网格在移动终端的重建也是渐进进行的。从网格 $M^j = (P^j, K^j)$ 重建网格 $M^{j+1} = (P^{j+1}, K^{j+1})$ ($j=0, 1, \dots, n-1$) 需要以下步骤:

(1) 将 P^j 作为 $j+1$ 层的偶点 $EVEN^{j+1}$;

(2) 用 P^j 预测 $j+1$ 层的所有奇点 ODD^j ;

(3) 利用误差 e^j 和 ODD^j ,以及式(1)得到奇点集 ODD^{j+1} 中各点的坐标;

(4) $EVEN^{j+1}$ 和 ODD^{j+1} 构成 $j+1$ 层的顶点 P^{j+1} ,重新剖分三角网格得到新的拓扑信息 K^{j+1} 。这样网格 M^{j+1} 就形成了。

可见,利用基网格 M^0 和误差值 e^0 ,可重建 M^1 。当渐进传输一部分误差信息 $e^0, e^1, \dots, e^{j-1}$ 后,可构建中间层网格 M^j ,当所有细节信息 e^{n-1} 被传输到移动终端后,可得到简化前的原始网格 M^n ,此时网格完全恢复到原始的形状,这时网格为无损还原。

在实际应用中,网格重建是根据终端设备的分辨率按需完成的。当设备分辨率低时,就没有必要完整地重建网格,只利用基网格和部分误差值进行重建生成 M^j 就能满足分辨率和视觉要求。 M^j 只是渐进网格的中间层,与原网格 M^n 相比会有一定的误差,这时网格为有损还原。由于 Loop 细分顶点仿射组合相关联顶点为4,而蝶形细分为8,因而,采用

Loop 细分的算法可加快网格重建的速度。

4 实验结果

在 VC++ 的环境下分别用 Loop 和蝶形逆细分对头像、e-Sphere、Venus 等 3 维图形网格模型进行了算法测试,两种方法生成的网格拓扑相同,所产生

的误差值不同,但个数相同,图 6、图 7 分别给出头像、e-Sphere 渐进网格效果,图 7 是 e-Sphere 网格经过渲染的效果。对使用逆 Loop 细分模式与逆蝶形细分的算法速度进行了比较 (CPU: AMD 1800Mhz, RAM: 256MB, OS: Windows2000),结果如表 2 所示。可见,利用逆 Loop 细分的网格简化与重建比利用逆蝶形细分要快得多。

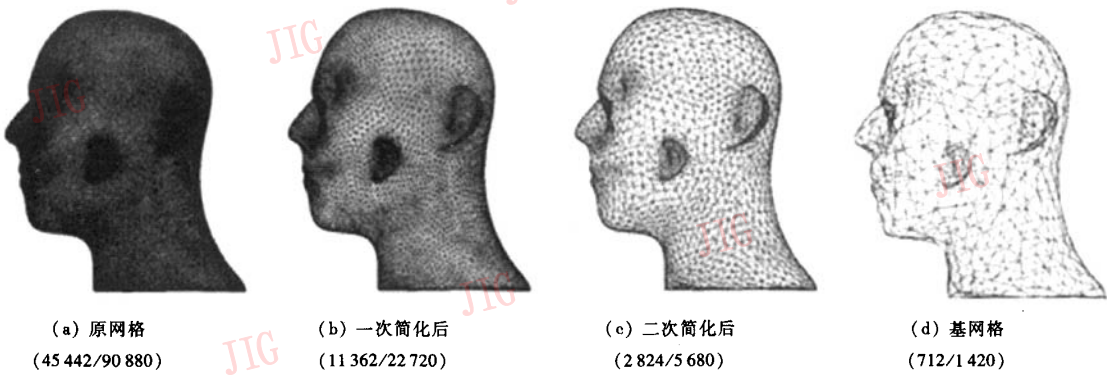


图 6 头像模型渐进网格(顶点数/三角面数)
Fig.6 Four levels of Head model(vertices/faces)

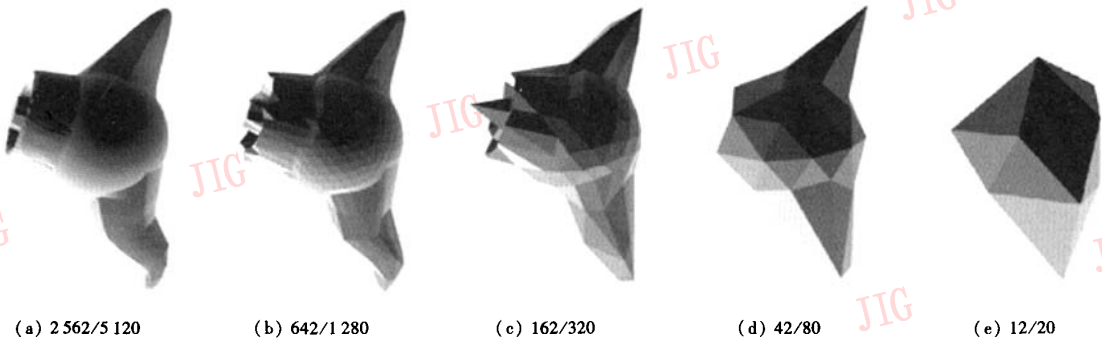


图 7 e-Sphere 模型渐进网格(顶点数/三角面数)
Fig.7 Five levels of e-Sphere model(vertices/faces)

表 2 网格简化与重建运行时间统计对比表

Tab.2 Runtimes statistics for simplification and reconstruction

网格模型	顶点数/三角形数 (输入网格)	顶点数/三角形数 (输出网格)	简化次数	运行时间(s)			
				网格简化		网格重建	
				Loop	Butterfly	Loop	Butterfly
Head	45 442/90 880	712/1 420	3	101.1	152.232	17.744	23.314
e-Sphere	2 562/5 120	12/20	4	3.164	4.005	0.512	0.646
Venus	9 603/19 200	153/300	3	14.617	21.124	2.406	3.152

在移动环境下,可以根据移动设备的分辨率和用户视觉等因素使用不同层次的渐进网格。当网格

完全重建时,网格还原是无损的;但当使用中间层次的渐进网格时,相对于原始网格将会产生误差。使

用网格误差评估软件 Metro (v4.06)^[14]对误差进行了测量,如果如表 3,其中 M^4 是原始网格, M^0 是基网格,其他是渐进网格的中间层,表中的 Max 误差、Mean 误差和 Hausdorff 距离是与原始网格 M^4 进行比较的结果。

表 3 e-Sphere 模型误差分析结果
Tab.3 Error Statistics for e-Sphere Model

渐进网格	顶点数	三角形数	MAX 误差	Mean 误差	Hausdorff 距离
M^4	2 562	5 120	0	0	0
M^3	642	1 280	0.197 355	0.020 310	0.225 822
M^2	162	320	0.949 826	0.078 561	0.949 826
M^1	42	80	0.932 492	0.152 490	1.626 944
M^0	12	20	1.506 532	0.374 888	3.836 133

5 结 论

提出了一种基于逆细分的渐进网格生成算法。它是通过对原始细密的网格进行分层次地简化,最后形成由一个基网格和一系列误差值组成的渐进网格。简述了算法的 3 个主要步骤和渐进网格在移动环境下的传输重建的方法。所生成的渐进网格可以无损地还原。文中还简述了渐进网格在网络中的传输及图形在移动设备上部分地重建的策略。本文方法的特点在于:

(1) 将逼近型的 Loop 细分模式,在网格简化过程中作为插值型细分模式处理,从而避免了调整偶点坐标值,加快了算法运行速度。

(2) 提出了由网格分裂、奇点预测、网格更新 3 个关键步骤组成的渐进网格生成算法,避免了求解细分矩阵方程(式 3)的复杂过程。

(3) 采用 Loop 逆细分,不必像文献[13]使用顶点仿射组合复杂的蝶形逆细分,大大降低算法的复杂度。加快了网格简化和网格重建的速度。

实验结果表明,采用基于逆 Loop 细分生成渐进网格的算法效率高,可使图形在终端设备上快速渲染,与以往的算法相比,网格相关联顶点少,运行速度快,更易于在实际中应用。另一方面,本文提出的算法是基于带有细分连通性的网格,存在着一定的局限性,故在图形的压缩和图形拟合方面尚需做大量的研究工作。另外,改进算法的数据结构也会大大地提高算法运行的速度。

参考文献 (References)

1 Agrawal P, Famolari D. Mobile computing in next generation wireless networks[A]. In: Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications[C], Seattle, USA, 1999:32 ~ 39.

2 Hoppe H. Progressive meshes [A]. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques [C], New Orleans, LA, USA, 1996: 99 ~ 108.

3 Mongkolnam P, Razdam A, Farin G. Loop progressive geometry compression for triangle meshes [J]. Computer Aided Geometric Design, 1999, 16(8):837 ~ 840.

4 Khodakovsky A, Schröder P, Sweldens W. Progressive geometry compression[A]. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques[C], New Orleans, Louisiana, USA, 2000: 271 ~ 278.

5 Jiang Sui-ping, Zhou Min-tian, Dai Ying. Mesh simplification based on normals [J]. Chinese Journal of Computers, 1999, 22(10): 1074 ~ 1079. [蒋遂平,周明天,戴颖. 基于法向的网格简化[J]. 计算机学报, 1999, 22(10):1074 ~ 1079].

6 Tang Jie, Zhang Fu-yan. An algorithm for generating multiresolution model of mesh based on error control [J]. Chinese Journal of Computers, 2005, 28(9):1534 ~ 1540. [唐杰,张福炎. 一种基于误差控制的网格多分辨率模型生成算法[J]. 计算机学报, 2005, 28(9): 1534 ~ 1540.]

7 Garland M, Heckbert P S. Surface simplification using quadric error metrics[A]. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97) [C], New York, USA, 1997: 209 ~ 216.

8 Suzuki H, Takeuchi S, Kanai T. Subdivision surface fitting to a range of points [A]. In: The 7th Pacific Conference on Computer Graphics and Applications[C], Seoul, Korea, 1999: 158 ~ 167.

9 Loop C. Smooth Subdivision Surface Based on Triangles[M]. Utah: University of Utah, Department of Mathematics, 1987.

10 Farin G E. Curves and Surfaces for Computer Aided Geometric Design 5th Edition[M]. New York: Academic Press, 2001.

11 Dyn N, Levin D, John A. A butterfly subdivision scheme for surface interpolatory with tension control [J]. ACM Transactions on Graphics, 1990, 9(1): 160 ~ 169.

12 Lee A, Sweldens W, Schroder P. MAPS: Multiresolution adaptive parameterization of surface [A]. In: Proceedings of ACM SIGGRAPH '98[C], Orlando, Florida, USA, 1997: 95 ~ 104.

13 Luo X N, Zheng G F. Progressive Meshes Transmission over a Wired-to-wireless Network[EB/OL]. <http://dx.doi.org/10.1007/s11276-006-7603-1>, 2006-06-11

14 Cignoni P, Rocchini C, Scopigno R. Metro: measuring error on simplified surfaces [J]. Computer Graphics Form, 1998, 17(2): 167 ~ 174.