

面向移动端的渐进网格简化算法

褚苏荣^{1,2}, 牛之贤^{1*}, 宋春花¹, 牛保宁¹

(1. 太原理工大学 信息与计算机学院, 太原 030600; 2. 山西大学商务学院 实验实训教学中心, 太原 030031)

(* 通信作者电子邮箱 niuniurose63@163.com)

摘要:针对现有渐进网格(PM)简化算法在网格高度简化时无法保持模型关键特征、简化速度慢、无法适应多种模型等问题,提出一种以可变参数结合二次误差和类曲率特征度的边折叠算法(QFVP),用于构建面向移动端的渐进网格。首先,该算法通过设置可变参数 w ,调整二次误差和类曲率特征度在边折叠误差中的相对大小,提升了算法的简化质量,扩大了算法的适用范围;其次,训练了一个误差反向传播(BP)神经网络,用于确定模型 w 值;再次,提出了边折叠过程中法向量线性估算法,提高算法简化速度,与Gouraud估算法相比,平均缩短网格简化时间23.7%。对比实验显示,QFVP简化生成渐进网格的基网格整体误差小于二次误差度量(QEM)算法和Melax算法;简化时间比QEM算法平均延长7.3%,比Melax算法平均缩短54.7%。

关键词:渐进网格;网格简化;二次误差;边折叠;Hausdorff距离;误差反向传播神经网络

中图分类号:TP391.41 **文献标志码:**A

Progressive mesh simplification algorithm for mobile devices

CHU Surong^{1,2}, NIU Zhixian^{1*}, SONG Chunhua¹, NIU Baoning¹

(1. College of Information and Computer, Taiyuan University of Technology, Taiyuan Shanxi 030600, China;

2. Teaching Center for Experiment and Practical Training, Business College of Shanxi University, Taiyuan Shanxi 030031, China)

Abstract: To solve the problems that existing Progressive Mesh (PM) simplification algorithms are facing, such as, losing key features when meshes are highly simplified, low simplification speed and limited applicability for various models, an edge-collapsing mesh simplification algorithm combining Quadric Error Metric (QEM) and curvature-like Feature value with Variable Parameter (QFVP) was proposed to build progressive meshes for mobile devices. Firstly, the variable parameter w was set to control the relative magnitude of quadratic error and curvature-like value in edge-collapsing error, improving the simplification quality of the algorithm and making the algorithm more applicable. Secondly, an error Back Propagation (BP) neural network was trained to determine the w value of the model. Thirdly, the normal vector linear estimation method in the edge-collapse process was proposed, which shortens the mesh simplification time by 23.7% on average compared to Gouraud estimation method. In the comparison experiments, the PM's basic meshes generated by QFVP have smaller global error (measured by Hausdorff distance) than those generated by QEM algorithm or Melax algorithm. And QFVP has simplification time about 7.3% longer than QEM algorithm and 54.7% shorter than Melax algorithm.

Key words: Progressive Mesh (PM); mesh simplification; quadric error; edge-collapsing; Hausdorff distance; error Back Propagation (BP) neural network

0 引言

渐进网格(Progressive Mesh, PM)^[1]把高分辨率三角网格模型表示成一个分辨率极低的基网格和一组连续的渐进记录,是一种改进的三角网格表示形式,适用于模型的多分辨率显示、多层次细节(Levels of Detail, LoD)显示^[2]和渐进传输,广泛用于3D游戏、实景地图等应用。随着移动互联网的普及,渐进网格的应用延伸到移动端。移动端资源高度受限,迫切需要保持模型特征的低分辨率三角网格模型。基网格是渐进网格生成多分辨率网格的基础,其质量直接影响渐进网格生成低分辨率网格模型的质量,因而,保持模型关键特征的基

网格是当前研究的关键。

渐进网格的构造过程是网格简化的过程,现有网格简化算法根据折叠误差度量指标的数量,分为单一指标算法和多指标算法。单一指标算法可以找到折叠误差对应的几何量,易从数学角度证明折叠误差定义的合理性。单一指标算法包括:Garland等^[3]提出的二次误差度量(Quadric Error Metrics, QEM)算法、周元峰等^[4]提出的体积平方和度量算法等。此类算法计算速度快,但网格高度简化时损失模型关键特征。

多指标算法无法找到折叠误差对应的几何量,但在保持模型特征、提升简化网格质量方面有重大实践意义,是近年来

收稿日期:2019-07-03;修回日期:2019-09-10;录用日期:2019-09-10。 基金项目:国家自然科学基金资助项目(61572345)。

作者简介:褚苏荣(1990—),女,山西太原人,硕士研究生,主要研究方向:计算机图形建模、数据挖掘;牛之贤(1963—),女,山西长治人,副教授,硕士,主要研究方向:信息检索、数据挖掘、计算理论与算法;宋春花(1966—),女,山西大同人,副教授,博士,CCF会员,主要研究方向:计算机图形建模、可视化仿真、数据库建模;牛保宁(1964—),男,山西晋中人,教授,博士,CCF会员,主要研究方向:数据库系统性能管理、自主计算、云计算。

研究热点。Melax以类曲率特征度^[5-6]乘以折叠边长作为折叠误差,保持模型尖锐特征,简化速度快^[7]。Hoppe等^[8]定义了能量函数作为折叠误差,网格简化质量高,但简化速度慢。近年来,大量结合二次误差和类曲率特征度的算法涌现出来,李红波等^[9]、刘峻等^[10]、黄佳等^[11]、焦越等^[12]都做了保持特征的改进。但是,现有多指标算法以固定方式结合多种指标,结合指标多时、网格简化速度慢;结合指标少时,不能普遍适用于多种模型。

针对现有算法的不足,本文提出了一种以可变参数结合二次误差和类曲率特征度的边折叠^[1]简化算法(edge-collapsing mesh simplification algorithm combining Quadric error metrics and curvature-like Feature value with Variable Parameter, QFVP),用于简化生成渐进网格。该算法设置可变参数 w ,该参数可以改变二次误差和类曲率特征度的结合方式,提高简化质量,扩大算法适用范围;训练了一个误差反向传播(error Back Propagation, BP)神经网络,用于根据模型数据确定 w 值;提出了边折叠过程中法向量线性估算法,缩短了简化时间。实验结果表明,本文算法计算速度快,普遍适用于多种类型三角网格模型,简化生成的基网格保持模型关键特征,是生成高质量、低分辨率渐进网格的有效算法。

1 相关工作

1.1 渐进网格原理

渐进网格常用元素折叠类网格简化方法,包括边折叠法^[1]和三角形折叠法^[13]。本文算法基于边折叠,图1画出了一次边折叠(ecol)的过程。一次边折叠使网格中减少了一个顶点和两个三角形;多次执行边折叠操作,就可以实现网格简化。未经简化的原始网格记作 M^n ,经过 n 次边折叠得到了基网格 M^0 。

$$M^n \xrightarrow{ecol_1} M^{n-1} \xrightarrow{ecol_2} \dots \xrightarrow{ecol_{n-1}} M^1 \xrightarrow{ecol_n} M^0$$

边折叠的逆过程称为点分裂(vsplit)。每一次点分裂对应一条渐进记录,基网格 M^0 和渐进记录的集合 $\{vsplit_1, vsplit_2, \dots, vsplit_n\}$ 共同构成了渐进网格。

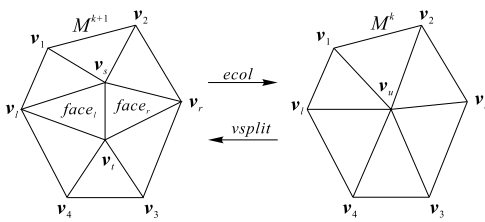


图1 边折叠(点分裂)过程示意图

Fig. 1 Schematic diagram of edge collapsing (vertex splitting) process

渐进网格的构造和存储在服务器端进行。移动端根据应用需求,向服务器端请求获取模型的基网格和一定数量的渐进记录,快速重建出所需分辨率的模型,实现模型的快速多分辨率重建和模型的LoD显示。

渐进网格中常用概念的符号说明如下。1)顶点:网格中顶点坐标用齐次坐标表示,记作 $v_i = [x_i, y_i, z_i, 1]^T$;2)平面:网格中每个三角形都对应一个平面,由平面方程 $ax + by + cz + d = 0$ (其中 $a^2 + b^2 + c^2 = 1$)确定,记作 $p = [a, b, c, d]^T$;3)边:顶点 v_s 和 v_t 构成的边记作 $\{v_s, v_t\}$;4)边折叠:边 $\{v_s, v_t\}$ 折叠到顶点 v_u 的过程记作 $\{v_s, v_t\} \rightarrow v_u$ 。5)邻接三角形:以顶点 v 为顶点的三角形称为顶点 v 的邻接三角形;6)邻接平面:顶点 v

邻接三角形对应的平面称为邻接平面,顶点 v 邻接平面的集合记作 $plane(v)$ 。

1.2 二次误差的定义

QEM算法定义边折叠 $\{v_s, v_t\} \rightarrow v_u$ 的折叠误差为顶点 v_u 到平面 $p(p \in plane(v_s) \cup plane(v_t))$ 距离的平方和,如式(1):

$$\Delta v_u = v_u^T Q_u v_u \quad (1)$$

其中: v_u 是折叠后新顶点,其坐标为 $[x_u, y_u, z_u, 1]^T$ 。QEM算法为网格中任意顶点 v 定义了一个 4×4 的对称矩阵 Q , $Q = \sum_{p \in plane(v)} pp^T$ 。 Q_u 的定义如式(2):

$$Q_u = \sum_{p \in plane(v_s) \cup plane(v_t)} pp^T \quad (2)$$

QEM算法还定义了 Q 在边折叠 $\{v_s, v_t\} \rightarrow v_u$ 过程中的线性关系,即:

$$Q_u = Q_s + Q_t = \sum_{p \in plane(v_s)} pp^T + \sum_{p \in plane(v_t)} pp^T \quad (3)$$

式(3)巧妙地用简单的矩阵加法代替了复杂的点到平面距离的计算,大大提高了简化速度^[3]。

1.3 类曲率特征度的定义

传统曲率定义在光滑的曲线或曲面上。为了反映三角网格表面特征的强弱,许多研究在离散的三角网格模型表面定义了类似曲率的概念^[5-6],称为类曲率特征度。网格表面平坦区域、曲率小;尖锐区域、曲率大。要保持模型关键特征,应大幅简化曲率小的区域、适当保留曲率大的区域。类曲率特征度的定义方式多种多样,定义越复杂,计算耗时越长。为了保障网格简化的效率,对于边折叠 $\{v_s, v_t\} \rightarrow v_u$,定义类曲率特征度^[9]为:

$$f = 1 - \cos \theta \quad (4)$$

其中 θ 代表折叠边端点 v_s, v_t 法向量的夹角。折叠边附近区域越平坦,式(4)中顶点 v_s, v_t 法向量的夹角越小, f 的值越接近0;反之,顶点法向量夹角越大, f 值越接近2。

2 QFVP

本章重点说明QFVP折叠误差定义、参数 w 确定、法向量估算和算法步骤。本章涉及的实验中,简化网格质量采用客观指标——简化误差(Hausdorff距离^[14]),和主观指标——基网格视觉效果来评价。由于本文模型的坐标都标准化到 $[-1, 1]$ 区间内,所以Hausdorff距离没有单位。

2.1 折叠误差的定义

QFVP的折叠误差在二次误差的基础上结合了类曲率特征度。对于边折叠 $\{v_s, v_t\} \rightarrow v_u$,本文定义边折叠误差为:

$$\Delta v_u = v_u^T Q_u v_u f^w; w \geq 0 \quad (5)$$

式(5)定义的折叠误差中 $v_u^T Q_u v_u$ 是1.2节定义的二次误差; f 是1.3节定义类曲率特征度, w 是 f 的指数,反映 f 对折叠误差影响力大小。式(4)代入式(5),可得式(6):

$$\Delta v_u = v_u^T Q_u v_u (1 - \cos \theta)^w; w \geq 0 \quad (6)$$

对式(6)求 Δv_u 关于 θ 的偏导数可得式(7):

$$\frac{\partial \Delta v_u}{\partial \theta} = v_u^T Q_u v_u w (1 - \cos \theta)^{w-1} \sin \theta \quad (7)$$

显然,当 $w > 0$ 时, $\frac{\partial \Delta v_u}{\partial \theta}$ 恒大于0,所以式(7)中的 Δv_u 是关于 θ 的增函数,即该折叠区域越平坦(θ 越小),边折叠误差 Δv_u 越小,该边优先折叠。

现有的多指标算法以固定的方式融合多个指标,使得融

合指标较少时,对模型的普适性较差;融合指标较多时,简化速度缓慢。为了用较少的指标适应多种模型,本文结合二次误差和类曲率特征度时,引入了可变参数 w 。

为了证明参数 w 对网格简化结果的影响,进行了以下实验。实验用到了牛、直升机、兔、手骨4种模型,模型原始网格如图2。

表1列举了 w 取不同值时QFVP简化网格的整体误差。由表1可以看出:

- 1) 参数 w 的取值显著影响网格简化质量,针对模型设置参数 w 值,能提升模型简化质量。
- 2) 使模型简化误差最小的 w 取值与原始网格数量、模型简化量、模型简化比例相关。
- 3) 使网格误差最小的 w 值一般在0~2,因此 w 取值范围为

$w \in [0, 2]$ 。

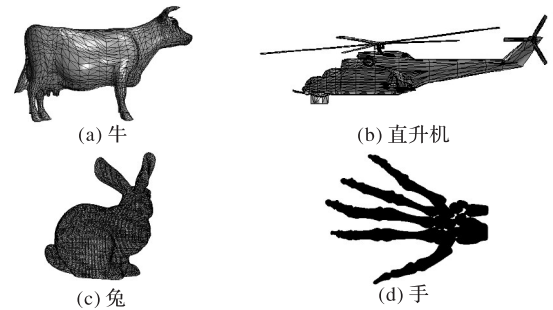


图2 牛、直升机、兔、手模型原始网格

Fig. 2 Original meshes of cow, hind, bunny, hand

表1 不同 w 取值下QFVP生成基网格简化误差

Tab. 1 Simplification errors of QFVP basic meshes with different values of w

模型	原始网格 三角形数	简化网格 三角形数	模型简化 比例/%	简化误差				
				$w=0.0$	$w=0.5$	$w=1.0$	$w=1.5$	$w=2.0$
牛	5804	1161	80	0.024733	0.028809	0.020492	0.036881	0.051415
		581	90	0.048358	0.032823	0.055442	0.112377	0.064680
直升机	6448	1290	80	0.023351	0.023421	0.044818	0.020067	0.054557
		645	90	0.034309	0.025606	0.078654	0.070586	0.068641
兔	69451	13891	80	0.004618	0.005632	0.008788	0.009180	0.009821
		6946	90	0.006265	0.008224	0.009154	0.010427	0.009315
手	654666	130934	80	0.000965	0.000616	0.000873	0.001019	0.001083
		65467	90	0.001087	0.001127	0.001126	0.001041	0.001388

2.2 参数 w 取值

BP神经网络可以实现从输入到输出的非线性映射,主要用于模式识别、函数逼近和分类等问题^[15]。由2.1节实验知, w 取值与模型简化数据之间不存在简单的线性关系。根据模型简化情况,在0.0、0.5、1.0、1.5、2.0中选择恰当值作为参数 w 的值,可以归纳为一个分类问题,采用BP神经网络解决。

网络共分3层。输入层有3个神经元,分别对应输入向量的3个分量。输入向量 $X=(x_1, x_2, x_3)$ 表示模型简化数据,其中 x_1 表示原始网格三角形个数, x_2 表示简化网格三角形个数, x_3 表示简化比例。输出层5个神经元,分别对应输出向量的5个分量。网络输出向量 $Y=(y_1, y_2, y_3, y_4, y_5)$, y_1 到 y_5 依次对应 w 的可能取值0.0、0.5、1.0、1.5、2.0。根据文献[15]确定隐含层神经元个数为4,转移函数为Sigmoid函数 $f(x)=1/(1+e^{-x})$,学习速率为0.1,误差函数为期望输出与实际输出的平均误差。

牛、直升机、兔、手4种模型的原始网格数据量覆盖千、万、十万多个数量级;网格类型涵盖均匀网格与非均匀网格;模型形状特征多样,具有代表性。因此,基于这4种模型来构造学习样本。由于渐进网格基网格简化比例大于50%,因此设定样本的简化比例 $x_3 \in (0.5, 1.0)$ 。首先,每种模型随机生成40个简化比例,计算出对应的输入向量;然后,通过实验,确定每个输入向量对应的期望输出向量。构造完成160条样本后,随机选取80%作为训练样本,20%作为测试样本。

通过训练,该网络对32条测试样本,预测准确27条,预测准确率约为84.4%。

2.3 顶点法向量估算方法

式(5)中类曲率特征度的计算需要用到顶点法向量。光滑曲面上任意顶点的法向量根据定义可以精确求出。然而三角网格不是光滑曲面,顶点法向量需要估算。本文在网格初始化阶段,采用Gouraud提出的估算法^[16]来计算原始网格中顶点的法向量;在网格简化过程中,提出线性估算方法来计算边折叠产生新顶点的法向量。

2.3.1 原始网格顶点法向量Gouraud估算法

某些三角网格模型文件包含了原始网格的顶点法向量。对于没有给出顶点法向量的三角网格模型,需要在网格初始化阶段对原始网格顶点法向量进行估算。彭玉辉等^[16]对几种三角网格顶点法向量估算方法的精度进行了对比分析,其中,Gouraud估算法的平均误差与最优方法的差距在5个百分点内,但是避免了邻接三角形面积、顶角等复杂计算,显著减少了计算量。因此,本文选Gouraud估算法来计算原始网格顶点法向量,即:

$$n_v = \left(\sum_{p \in \text{plane}(v)} n_p \right) / c \quad (8)$$

其中: n_v 表示原始网格中顶点 v 的法向量, n_p 表示顶点 v 邻接平面 p 的法向量, c 表示顶点 v 邻接平面的个数。式(8)的含义是,原始网格中任意顶点的法向量等于其邻接平面法向量的平均值。

2.3.2 边折叠过程中顶点法向量线性估算法

如图1所示,边折叠终点 v_u 的邻接三角形是新产生的,按照Gouraud法估算顶点 v_u 的法向量,则需重新计算这些三角形法向量,计算量大、速度慢,因此,本文提出边折叠过程中的法向量线性估算法。

如图1所示,折叠终点 v_u 在网格 M^k 和 M^{k+1} 上都存在,所以法向量 n_{v_u} 既可以用网格 M^k 来估计,也可以用网格 M^{k+1} 来估计;由于点 v_u 是边 $\{v_s, v_t\}$ 上的点,所以法向量 n_{v_u} 受顶点 v_s 、 v_t 邻接三角形法向量的影响。因此,在网格 M^{k+1} 上用顶点 v_s 、 v_t 邻接三角形法向量的平均值来估算法向量 n_{v_u} ,即:

$$n_{v_u} = \frac{\sum_{p \in \text{plane}(v_s) \cup \text{plane}(v_t)} n_p}{|\text{plane}(v_s) \cup \text{plane}(v_t)|} \quad (9)$$

符号“ $|\cdot|$ ”表示集合中元素的个数,然而,式(9)求并集计

算较复杂,因此对式(9)进行近似计算,即:

$$\mathbf{n}_{v_u} = \frac{\sum_{p \in \text{plane}(v_s)} \mathbf{n}_p + \sum_{p \in \text{plane}(v_t)} \mathbf{n}_p}{|\text{plane}(v_s)| + |\text{plane}(v_t)|} = \frac{c_s \mathbf{n}_{v_s} + c_t \mathbf{n}_{v_t}}{c_s + c_t} \quad (10)$$

其中: c_s 、 c_t 分别表示顶点 v_s 、 v_t 邻接三角形个数, $c_s = |\text{plane}(v_s)|$ 、 $c_t = |\text{plane}(v_t)|$ 。显然,式(10)重复计算了图1中三角形 $face_i$ 和 $face_j$ 对应平面的法向量,但是建立了顶点 v_u 、 v_s 、 v_t 法向量的线性关系,能够显著简化计算。故,本文对于边折叠 $\{v_s, v_t\} \rightarrow v_u$, 定义折叠终点 v_u 法向量 $\mathbf{n}_{v_u}$ 线性估算式如下:

$$\mathbf{n}_{v_u} = \frac{c_s \mathbf{n}_{v_s} + c_t \mathbf{n}_{v_t}}{c_s + c_t} \quad (11)$$

三角网格简化过程中,网格表面不断变化,不能采用文献[16]的静态评估法求估算精度。为了检验法向量线性估算法的有效性,进行了以下实验。实验用到了猫、车、苹果、龙4种模型,模型原始网格如图3,原始网格数量及简化情况见表2。模型根据2.2节神经网络获得 w 值,分别为2.0、0.5、1.5、0.5,经验证, w 值预测准确。

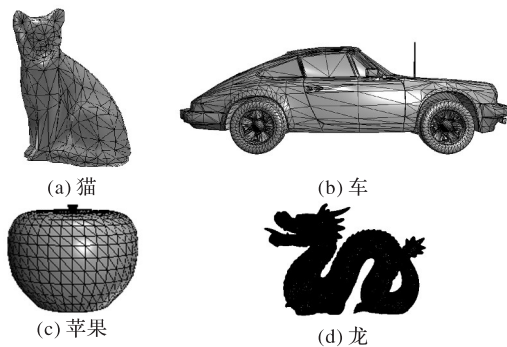


图3 猫、车、苹果、龙模型原始网格

Fig. 3 Original meshes of cat, car, apple, dragon

表2 猫、车、苹果、龙模型简化说明

Tab. 2 Simplification description of cat, car, apple, dragon

模型名称	原始网格 三角形数	基网格 三角形数	简化 比例/%	QFVP w 值
猫	671	87	83.0	2.0
车	10 474	716	93.0	0.5
苹果	1 704	144	90.0	1.5
龙	871 414	2 252	99.3	0.5

实验对比了边折叠阶段分别采用法向量 Gouraud 估算法和线性估算法,对 QFVP 简化时间和基网格简化误差的影响。实验数据见表3,简化生成的基网格见图4。

表3 法向量估算法对 QFVP 简化时间和误差的影响

Tab. 3 Effect on QFVP's simplification time and error caused by different normal vector estimation methods

模型名称	简化时间/s		时间 缩短 比例/ %	基网格简化误差 (Hausdorff 距离)		简化误差 增减(+/-) 比例/%
	Gouraud 估算	线性 估算		Gouraud 估算	线性 估算	
猫	0.277	0.210	24.2	0.031 452	0.024 457	-22.2
车	5.229	4.020	23.1	0.040 833	0.030 573	-25.1
苹果	0.809	0.630	22.1	0.006 108	0.004 262	-30.2
龙	525.658	392.816	25.3	0.002 579	0.002 699	+4.7

从中可以看出,边折叠阶段采用法向量线性估算法平均

缩短23.7%的简化时间;对于简化量较小的模型显著减少了简化误差,对于简化量较大的模型增加了简化误差,误差增幅不超过5%,增幅可控。因此,法向量线性估算法是一种有效的法向量估算法。

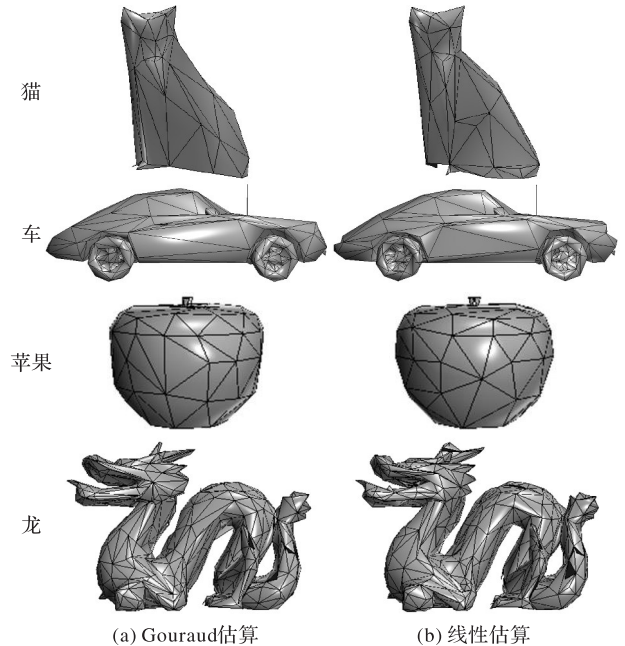


图4 QFVP 采用不同法向量估算法生成的基网格

Fig. 4 Basic meshes simplified by QFVP taking different normal vector estimation methods

2.4 算法步骤

2.4.1 算法步骤

1) 设置网格简化数据。设置基网格简化比例;根据2.2节神经网络,预测 w 值。

2) 初始化原始网格。读入原始网格;根据式(2)和式(8)分别初始化各顶点的 Q 值和法向量。

3) 初始化待折叠边列表。根据式(6)计算原始网格中每条边折叠终点的位置和折叠误差,将每条边插入待折叠边列表(该表是一个小根堆)。

4) 网格简化。①取待折叠边列表中折叠误差最小的边,判断该边是否可折叠,如果可折叠执行4)②,不可折叠执行4)③。②进行边折叠:生成折叠对应的渐进记录;计算折叠区域内新边折叠终点的位置和折叠误差,将新边插入待折叠边列表;更新当前网格折叠区域的拓扑结构。③从待折叠边列表中删除当前折叠边。④判断简化终止条件是否达到,如果达到,则停止简化,执行步骤5);如果没有达到,则继续简化,执行步骤4)①。

5) 输出渐进网格。

2.4.2 算法细节说明

步骤3)、4)②中折叠终点的位置在折叠边的两个端点和中点之间选择折叠误差较小的点,这种做法比求最优位置^[4]简单、比半边折叠^[10]精确。由于在待折叠边列表中查找包含某一个顶点的边非常耗时,所以对待折叠边列表只进行新边插入、不进行旧边删除,步骤4)①中不可折叠边指网格中实际已不存在的旧边。步骤4)④中简化终止条件有2个:达到

指定的简化量或待折叠边列表为空。

3 实验验证

实验将 QFVP 与经典的 QEM 算法和 Melax 算法进行对比。对比内容包括各算法生成渐进网格的简化时间和基网格质量,其中,基网格质量采用客观指标——简化误差(Hausdorff 距离)和主观指标——网格视觉效果来评价。

实验的环境为:配置 Intel Celeron 1007U 1.5 GHz 的 CPU、4 GB 内存和 Windows 7 操作系统的 PC。开发环境为 Visual Studio 2017;编程语言是 C++ 11 和 OpenGL 3.3。

实验用到猫、车、苹果、龙模型,模型原始网格及渐进网格简化情况见图 3、表 2。QFVP 在网格初始化阶段和边折叠阶段,分别采用 Gouraud 估算法和线性估算法,估算顶点法向量。猫、车、苹果、龙神经网络预测得到的 w 值分别为 2、0.5、1.5、0.5,经检验,预测正确。

3.1 简化时间对比分析

表 4 列出了 QEM 算法、Melax 算法和本文的 QFVP 简化生成渐进网格的时间。由表 4 可见,本文算法简化生成渐进网格的时间,比 Melax 算法平均缩短 54.7%,比 QEM 算法平均延长 7.3%。

表 4 各算法生成渐进网格的简化时间对比

Tab. 4 Simplification time comparison of each algorithm for generating PM

模型名称	简化生成渐进网格时间/s			QFVP 比 QEM 时间延长比例/%	QFVP 比 Melax 时间缩短比例/%
	QEM 算法	Melax 算法	QFVP 算法		
猫	0.195	0.431	0.210	7.7	51.3
车	3.782	9.155	4.020	6.3	56.1
苹果	0.588	1.603	0.630	7.1	60.7
龙	362.872	797.356	392.816	8.2	50.7

QEM 算法和 Melax 算法是同类型算法中简化速度快的算法。这表明,本文算法简化速度快,具备处理大规模三角网格模型的潜力。

3.2 简化质量对比分析

表 5 列出了各算法简化生成渐进网格基网格的简化误差。由表 5 可见,本文 QFVP 生成渐进网格基网格的简化误差小于 QEM 算法和 Melax 算法。

表 5 各算法生成渐进网格的基网格简化误差对比

Tab. 5 Basic meshes' simplification error comparison of each algorithm for generating PM

模型名称	基网格简化误差(Hausdorff 距离)		
	QEM 算法	Melax 算法	QFVP
猫	0.028 994	0.039 878	0.024 457
车	0.039 553	0.211 860	0.030 573
苹果	0.005 037	0.005 143	0.004 262
龙	0.003 579	0.011 230	0.002 699

图 5 画出了各算法简化生成的基网格。从主观视觉效果上看:对于猫模型,本文算法对应的基网格保留了猫耳朵,而 QEM 算法损失了这一特征,Melax 算法使猫脚部明显变形。对于车模型,本文算法清晰保留了天线、后视镜、门把手、车头包含保险杠在内的全部细节,车轮形状保持好;而另外两种算法在这些关键特征上有不同程度的损失和变形。对于苹果模型,本文算法对苹果梗的形态保持好,模型整体较圆润;Melax 算法和 QEM 算法苹果梗变形严重。对于龙模型,网格表面存在空洞,Melax 算法对空洞扩大严重,模型形变明显;本文算法和 QEM 算法的视觉效果接近,相比而言,本文算法对于龙前、后腿与身体连接处的形态保持较好。

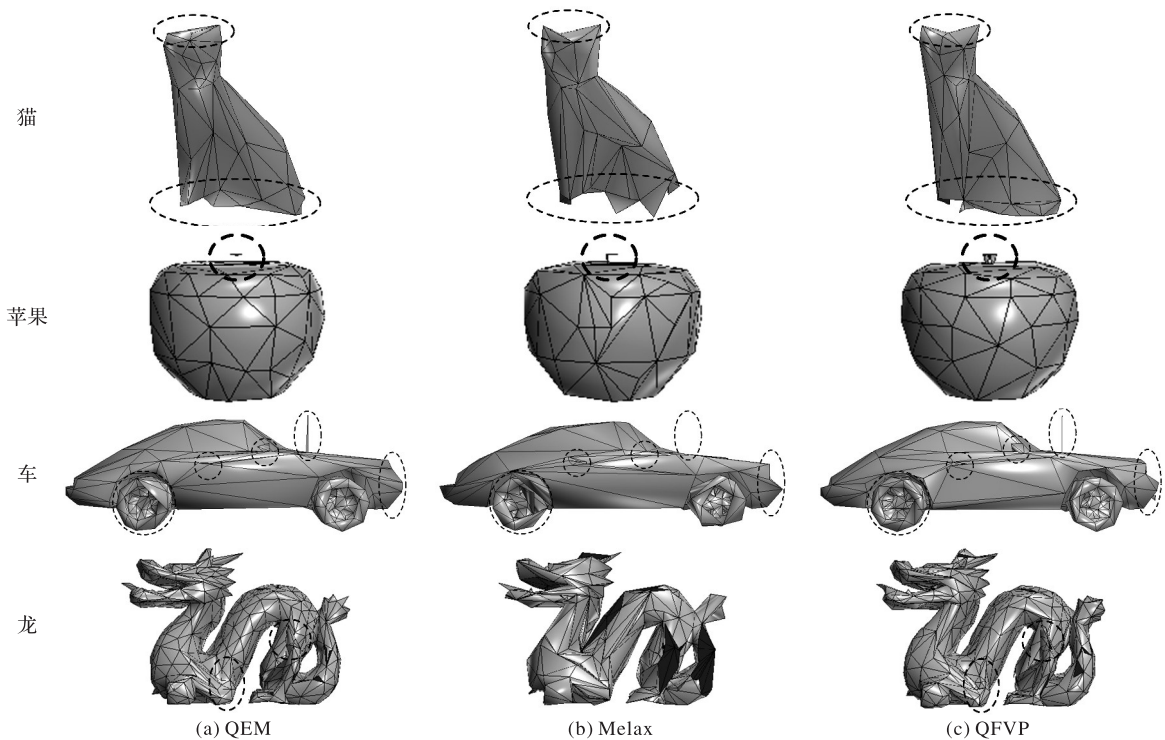


图 5 各算法简化生成渐进网格的基网格

Fig. 5 Basic meshes generated by different simplification algorithm

从客观简化误差和主观视觉效果看,本文算法简化生成基网格质量都优于其他两种算法。

综合实验结果,可以得出如下结论:1)本文算法简化速度快、简化质量高,在网格高度简化后保持模型关键特征,是生成面向移动端渐进网格的有效算法。2)本文算法的优越性体现在:生成的渐进网格在构造低分辨率模型时,能更好地保持模型特征。3)无论三角网格模型规模大小、是否均匀,本文算法普遍适用。4)本文算法的局限性在于,生成的渐进网格适合于构造低分辨率模型,构造出的高分辨率模型质量非最优。

4 结语

移动端由于资源受限,需要保持模型关键特征的低分辨率网格。面向移动端的渐进网格研究的重点是保持模型关键特征的基网格。本文提出了一种以可变参数结合二次误差和类曲率特征度的算法(QFVP);为了针对模型设置可变参数 w ,训练了BP神经网络;为了进一步提高该算法的速度,提出了边折叠过程中的法向量线性估算法。通过实验,证明了相应改进的有效性。通过和QEM算法和Melax算法对比,证明了本文算法简化速度快,高度简化后保持特征,生成的低密度网格质量高,是生成面向移动端渐进网格的有效算法。

本文研究在参数 w 的确定上,仍有待改进。对于输入单元缺乏模型形状特征、 w 取离散值、样本量小、BP神经网络固有缺陷等问题,进行改进,都可能提高 w 值的预测准确性,进一步提升网格质量。

参考文献 (References)

- [1] HOPPE H. Progressive meshes [C]// Proceedings of the 23rd Annual Conference on Graphics and Interactive Techniques. New York: ACM, 1996: 99-108.
- [2] BILJECKI F, LEDOUX H, STOTER J. An improved LOD specification for 3D building models[J]. Computers, Environment and Urban Systems, 2016, 59: 25-37.
- [3] GARLAND M. Quadric-based polygonal surface simplification[D]. Pittsburgh, PA: Carnegie Mellon University, 1999: 35-82.
- [4] 周元峰,张彩明,贺平. 体积平方度量下的特征保持网格简化方法[J]. 计算机学报, 2009, 32(2): 203-212. (ZHOU Y F, ZHANG C M, HE P. Feature preserving mesh simplification algorithm based on square volume measure [J]. Chinese Journal of Computers, 2009, 32(2): 203-212.)
- [5] 方惠兰,王国瑾. 三角网格曲面上离散曲率估算方法的比较与分析[J]. 计算机辅助设计与图形学学报, 2005, 17(11): 2500-2507. (FANG H L, WANG G J. Comparison and analysis of discrete curvatures estimation methods for triangular meshes [J]. Journal of Computer-Aided Design and Computer Graphics, 2005, 17(11): 2500-2507.)
- [6] 偶春生,张佑生. 基于特征保持的曲面网格简化[J]. 计算机应用研究, 2013, 30(10): 3162-3164, 3168. (OU C S, ZHANG Y S. Mesh simplification method based on preserving shape features [J]. Application Research of Computers, 2013, 30(10): 3162-3164, 3168.)
- [7] MELAX S. A simple, fast and effective polygon reduction algorithm [J]. Game Developer, 1998, 11: 44-49.
- [8] HOPPE H, DEROSE T, DUCHAMPY T, et al. Mesh optimization [C]// Proceedings of the 20th Annual Conference on Graphics and Interactive Techniques. New York: ACM, 1993: 19-26.
- [9] 李红波,刘昱晟,吴渝,等. 基于二次误差度量的大型网格模型简化算法[J]. 计算机工程与设计, 2013, 34(9): 3158-3162. (LI H B, LIU Y S, WU Y, et al. Simplification algorithm for large mesh models based on quadric error metrics [J]. Computer Engineering and Design, 2013, 34(9): 3158-3162.)
- [10] 刘峻,范豪,孙宇,等. 结合边折叠和局部优化的网格简化算法[J]. 计算机应用, 2016, 36(2): 535-540. (LIU J, FAN H, SUN Y, et al. Mesh simplification algorithm combined with edge collapse and local optimization [J]. Journal of Computer Applications, 2016, 36(2): 535-540.)
- [11] 黄佳,温佩芝,李丽芳,等. 保持细节特性的局部误差渐进网格简化算法[J]. 计算机应用, 2016, 36(6): 1704-1708. (HUANG J, WEN P Z, LI L F, et al. Local error progressive mesh simplification algorithm for keeping detail features [J]. Journal of Computer Applications, 2016, 36(6): 1704-1708.)
- [12] 焦越,王慧青,吴煜豪,等. 结合面积度量和误差校正的网格简化算法[J]. 计算机工程与应用, 2019, 55(17): 221-226. (JIAO Y, WANG H Q, WU Y H, et al. Mesh simplification algorithm combined with area measurement and error correction [J]. Computer Engineering and Applications, 2019, 55(17): 221-226.)
- [13] 段黎明,杨尚朋,张霞,等. 基于遗传算法的三角网格折叠简化[J]. 光学精密工程, 2018, 26(6): 1489-1496. (DUAN L M, YANG S P, ZHANG X, et al. Collapsing simplification of triangular mesh based on genetic algorithm [J]. Optics and Precision Engineering, 2018, 26(6): 1489-1496.)
- [14] 张德军,何发智,田龙,等. 基于Z线和八叉树的高效Hausdorff距离计算方法[J]. 计算机辅助设计与图形学学报, 2018, 30(10): 1794-1800. (ZHANG D J, HE F Z, TIAN L, et al. Efficient Hausdorff distance calculation algorithm based on Z-order and octree [J]. Journal of Computer-Aided Design and Computer Graphics, 2018, 30(10): 1794-1800.)
- [15] 朱苗苗,潘伟杰,刘翔,等. 基于BP神经网络代理模型的交互式遗传算法[J/OL]. 计算机工程与应用: 1-8. [2019-02-27] <http://kns.cnki.net/kcms/detail/11.2127.tp.20190225.1624.006.html>. (ZHU M M, (PAN W J, LIU X, et al. The interactive genetic algorithm based on BP neural network and user cognitive surrogate model [J/OL]. Computer Engineering and Applications: 1-8. [2019-02-27]. <http://kns.cnki.net/kcms/detail/11.2127.tp.20190225.1624.006.html>.)
- [16] 彭育辉,高诚辉. 基于形状修正的三角网格模型顶点法矢估算方法[J]. 中国图象图形学报, 2010, 15(1): 142-148. (PENG Y H, GAO C H. An improved algorithm for vertex normal computation of triangular meshes based on shape correction [J]. Journal of Image and Graphics, 2010, 15(1): 142-148.)

This work is partially supported by the National Natural Science Foundation of China (61572345).

CHU Surong, born in 1990, M. S. candidate. Her research interests include computer graphics modeling, data mining.

NIU Zhixian, born in 1963, M. S., associate professor. Her research interests include information retrieval, data mining, computation theory and algorithms.

SONG Chunhua, born in 1966, Ph. D., associate professor. Her research interests include computer graphics modeling, visual simulation, database modeling.

NIU Baoning, born in 1964, Ph. D., professor. His research interests include performance management of database system, autonomic computing, cloud computing.