

中图法分类号: TP391 文献标志码: A 文章编号: 1006-8961(2011)05-0892-06

论文索引信息: 韦虎, 张丽艳, 刘胜兰, 李博. 基于支撑域的网格简化算法 [J]. 中国图象图形学报, 2011, 16(5): 892-897

基于支撑域的网格简化算法

韦虎, 张丽艳, 刘胜兰, 李博

(南京航空航天大学机电学院, 南京 210016)

摘要: 提出一种基于 2 次误差测度(QEM)的网格简化改进算法。算法首先对折叠边所产生的新顶点定义其在初始网格上的简化支撑域, 从而建立新顶点与初始网格之间的联系; 然后计算新顶点到支撑域的 2 次距离误差作为该顶点的全局简化误差, 并将原始 QEM 中的误差作为局部简化误差; 最后将两个误差之和作为新的折叠代价目标函数以实现原有 QEM 算法的改进。多个模型的简化实验表明, 改进算法能较好地保留初始网格的细节特征, 并且较为明显地降低简化误差。

关键词: 网格简化; 支撑域; 全局误差; 2 次误差测度; 边折叠

Mesh simplification based on support region

Wei Hu, Zhang Liyan, Liu Shenglan, Li Bo

(College of Mechanical and Electronic Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016 China)

Abstract: An improvement to quadric-error-metrics-based mesh simplification algorithm is presented in this paper. Firstly, the support region on the original mesh is defined for the new vertex generated in an edge collapse operation, and the connection between the new vertex and the original mesh is established. Secondly, the quadric distance error from the new vertex to its support region is calculated as the global simplification error, and the original quadric error is regarded as the local simplification error; Finally the sum of the two errors is defined as the new cost function of the edge collapse operation. Experimental results demonstrate that the improved algorithm can preserve the details better than the original one and reduce the simplification error obviously.

Keywords: mesh simplification; support region; global error; quadric error metrics; edge collapse

0 引言

随着科学技术的发展, 在计算机图形学、虚拟现实、物体 3 维外形测量等领域中, 所构造和使用的模型越来越复杂, 数据量也越来越庞大, 这些复杂的模型对计算机的存储、处理、绘制和传输都提出了很高的要求。然而在很多情况下, 高分辨率的模型并不总是必要的, 为了降低模型数据量, 提高应用效率, 网格简化成为一种简便有效的手段。目前网格简化

的研究已有很多^[1-2], 其中边折叠算法是按照一定的误差测度下不断将引起误差最小的边折叠为一个新顶点, 并将与该边 2 个顶点相连接的顶点连接到新顶点。这种方法可以生成连续的细节层次, 支持模型的渐进传输与绘制, 适用于任意 2 维流形的三角网格模型, 因此成为一种重要的三角网格简化算法。

边折叠算法关键要解决两个问题, 即如何确定边的折叠顺序和折叠边上新顶点的位置。Hoppe 等人^[3]采用能量优化的方法, 通过建立和求解复杂的

收稿日期: 2009-12-15; 修回日期: 2010-03-14

基金项目: 国家自然科学基金项目(50875130, 50875126); 江苏省自然科学基金项目(BK2009382)。

第一作者简介: 韦虎(1974—), 男。南京航空航天大学机电学院博士研究生, 主要研究领域为非接触 3 维几何测量, 逆向工程。E-mail: weihu2005@163.com。

全局能量优化方程来实现边折叠,取得很好的简化效果,然而该算法优化求解过程为非线性的,因此计算量大,运行速度慢。为降低时间复杂度, Garland 等人^[4]提出了一种基于 2 次误差测度(QEM)的简化算法,该算法以新顶点到被折叠边的两个顶点相关联平面的距离平方和作为误差测度,计算简单、运行速度快,并且得到了与 Hoppe 能量优化方法质量很接近的简化模型,因此 QEM 算法是目前普遍认同的一种简化方法。由于 QEM 算法的误差测度中只考虑了距离的度量,简化后的网格分布较为均匀,因此大规模简化后并不能很好地保持模型表面的细节特征。此后相当多的研究以减小简化误差和保持特征为目的,围绕 QEM 算法展开^[5-11],如李现民等人^[8]在 QEM 算法中,引入蝶形子分模式来计算新顶点的位置,试图进一步缩小简化网格与初始模型的误差,然而该算法未能将实验结果与 QEM 方法进行比较。刘晓利等人^[9]引入了尖锐特征度概念,在 QEM 算法的误差测度中加入惩罚项,较好地取得了保留网格细节特征的效果。但是该算法的简化误差较 QEM 有所增大,并且惩罚系数需要预先根据经验设定。

由于细节特征往往处于曲率变化较大的区域,显然对这些区域过度简化会导致简化误差的增加,反之,若在简化过程中保持了较小的简化误差,则简化模型必将会保留较多的细节特征。由于 QEM 算法在简化过程中仅考虑了连续简化网格之间的误差最小化,并没有考虑简化网格与初始网格之间的误差,因而不可避免会产生累积的简化误差,因此本文在 QEM 算法基础进行改进,通过定义网格简化支撑域的概念,跟踪折叠边上新顶点与初始网格之间的关系,并将支撑域的误差矩阵嵌入 QEM 的误差测度中,使得简化误差得到明显改善,同时结合面积加权方法,更好地保持了网格模型上的细节特征。

1 QEM 网格简化算法

QEM 算法^[4]属于边折叠机制,如图 1 所示,对边 e (图 1 左图中的粗线)进行折叠后, e 的两个端点 v_i, v_j 简化为一个新顶点 \bar{v} 。QEM 算法首先为初始网格的每个顶点 $v_i = [x_i, y_i, z_i, 1]^T$ 分配一个对称的误差矩阵 Q_i ,每条边的误差矩阵 $Q = \{q_{ij}\}$ 由 2 个顶点的误差矩阵之和得到:

$$Q = Q_i + Q_j \quad (1)$$

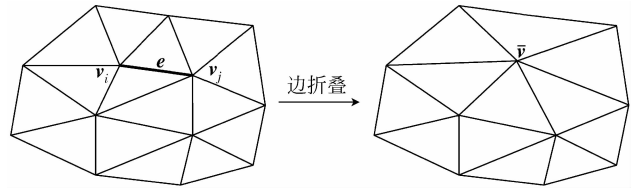


图 1 边折叠示意图

Fig. 1 Edge collapse

假设边 e 折叠后产生的新顶点为 \bar{v} ,则该折叠操作产生的简化误差(折叠代价)可定义为

$$\Delta(\bar{v}) = \bar{v}^T Q \bar{v} \quad (2)$$

新顶点的最佳位置可由极小化式(2)确定,即令

$$\frac{\partial \Delta(\bar{v})}{\partial \bar{x}} = \frac{\partial \Delta(\bar{v})}{\partial \bar{y}} = \frac{\partial \Delta(\bar{v})}{\partial \bar{z}} = 0, \text{得方程组为}$$

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \bar{v} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

解得

$$\bar{v} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

在 QEM 算法中,式(2)中的未知系数由顶点到该顶点邻接三角形所在平面的距离平方和确定,如图 1 所示,假定与顶点 v_i 或 v_j 的任一邻接三角形所在的平面方程为 $ax + by + cz + d = 0$,其中 $a^2 + b^2 + c^2 = 1$,记该平面参数为向量 $p = [a, b, c, d]^T$,则点 \bar{v} 到该三角片所在平面的距离平方为 $(p^T \bar{v})^2$,因此误差函数可写为

$$\Delta(\bar{v}) = \sum (p^T \bar{v})^2 = \sum \bar{v}^T p p^T \bar{v} = \bar{v}^T (\sum p p^T) \bar{v}$$

将上式与式(2)相比较可得 $Q = \sum p p^T$,其中

$$p p^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

QEM 算法可以概括如下:

- 1) 对每个网格顶点 v_i 计算对应的误差矩阵 Q_i ;
- 2) 选择所有有效折叠边;
- 3) 为每条有效边 $e(v_1, v_2)$ 计算简化后的新顶点 \bar{v} ,并计算该边的折叠代价 $\bar{v}^T (Q_1 + Q_2) \bar{v}$;

- 4) 对所有折叠代价进行排序;
- 5) 对最小折叠代价的边进行简化,按步骤 3) 方法更新所有受影响的边的折叠代价,再重新进行排序;
- 6) 重复执行 5),直至满足给定的终止条件。

2 改进的网格简化算法

QEM 方法中对有效边的折叠,是基于连续两次简化网格之间的误差优化,即保证在简化后的网格与当前网格之间的二次距离误差测度最小,因此从很大程度上可以保持网格的形状,使得简化后的网格分布比较均匀。但是由于没有考虑简化后的网格与初始网格之间的距离误差,因此无法避免产生简化误差的累积以及网格细节特征的丢失,本文在 QEM 算法的基础上,增加全局误差的考虑,使得模型的简化误差得到较明显的减少,同时结合面积加权方法,在一定程度上保持了初始网格上的细节特征。

2.1 简化支撑域

为在每次网格简化过程中,比较简化后网格与初始网格之间的关系,本文提出网格简化支撑域的概念

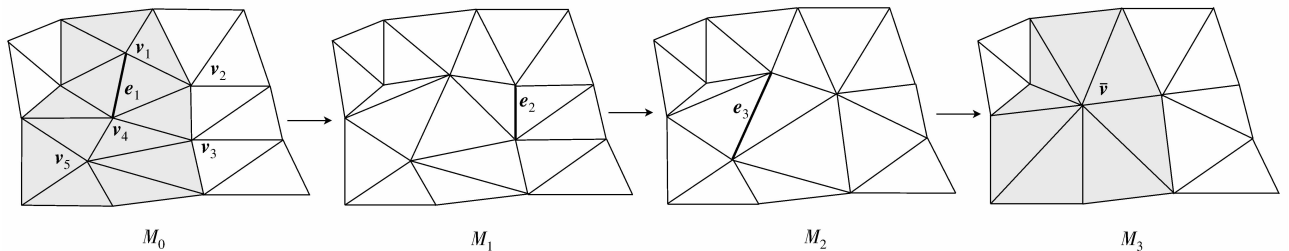


图 2 网格简化支撑域

Fig. 2 Support region for mesh simplification

支撑域的概念可以形象地理解为 $SF(\bar{v})$ 中内部点(即 $SV(\bar{v})$)“收缩”为一新顶点 \bar{v} ,而 $SF(\bar{v})$ 的形状和位置与 $NF(\bar{v})$ (图 2 网格 M_3 中灰色区域)非常接近,在一定程度上 $SF(\bar{v})$ 对 \bar{v} 的位置起着决定性作用。因此在网格的简化过程中,有必要考虑新顶点的支撑域对简化误差和细节特征保持的影响。

2.2 算法改进

由于 QEM 算法在保持相邻两层简化网格的局部形状上有着较大优势,因此本文在计算折叠代价时保留相邻两层简化网格之间的距离误差,定义该误差为局部误差。定义新顶点到其支撑域之间的距离误差为全局误差,仍然以新顶点到支撑域内三角

概念来建立简化后新顶点与初始网格的联系。

记初始网格为 M_0 , M_k 表示第 k 次边折叠后的渐进简化网格。记网格 M_k 中最小折叠代价边为 e , v_1^e 和 v_2^e 分别为 e 的两个端点, \bar{v} 为折叠边 e 之后所产生的新顶点,称 e 为 \bar{v} 的简化支撑边(简称支撑边)。判断 v_i^e (i 为 1 或 2) 是否为初始网格上的点,若不是,则查找 v_i^e 的支撑边,再判断其支撑边的端点是否为初始网格上的点,重复这个过程,直至所有支撑边的端点都为初始网格顶点。记这些网格顶点集为集合 $SV(\bar{v})$,并记 $NF(\bar{v})$ 表示与 \bar{v} 点相连的三角片集合,则定义新顶点 \bar{v} 的简化支撑域(简称支撑域) $SF(\bar{v})$ 为所有与 $SV(\bar{v})$ 中点相连的三角片集合,即

$$SF(\bar{v}) = \bigcup_{v_i \in SV(\bar{v})} NF(v_i)$$

如图 2 所示,图中 M_0 为初始网格, M_i ($i = 1, 2, 3$) 分别为对网格 M_{i-1} 中相应边 e_i (粗黑线所示) 折叠后的简化网格,对于 M_3 中新顶点 \bar{v} ,其支撑边为 M_2 中的边 e_3 ,通过查找 e_3 的两个端点直至原始网格 M_0 ,得到 $SV(\bar{v}) = \{v_1, v_4, v_5\}$,则 \bar{v} 点的支撑域为 $SF(\bar{v})$ 如图 2 网格 M_0 中灰色区域所示。

片所在平面的二次距离和来计算全局误差。最后以局部误差与全局误差的和作为最终的折叠代价目标函数。

考虑到模型上某个区域经过多次简化后,新顶点的支撑域包含的三角片个数较多,针对不同大小的三角片对特征的重要性不同^[10, 12],在计算全局误差时,对支撑域中每个三角片采用面积加权,以获得更好的特征保持。改进算法可以描述如下:

设边 e 简化后的新顶点为 \bar{v} ,对 $SF(\bar{v})$ 中每个三角片 $F_i \in SF(\bar{v})$ 计算该三角片的面积 w_{F_i} 和其在平面 p_{F_i} ,则支撑域的误差矩阵可以表示为

$$\tilde{Q} = \sum_{F_i \in SF(\bar{v})} p_{F_i} p_{F_i}^T \cdot w_{F_i} / W$$

式中, $W = \sum_{F_i \in SF(\bar{v})} w_{F_i}$ 。用 $\Delta_g(\bar{v})$ 表示全局误差, $\Delta_l(\bar{v})$ 表示式(2)的局部误差, $\Delta_{all}(\bar{v})$ 表示局部误差和全局误差的和, 有

$$\Delta_g(\bar{v}) = \bar{v}^T \tilde{Q} \bar{v} \quad (3)$$

将式(2)(3)相加, 得到新的折叠代价目标函数

$$\Delta_{all}(\bar{v}) = \Delta_g(\bar{v}) + \Delta_l(\bar{v}) = \bar{v}^T (\mathbf{Q}_i + \mathbf{Q}_j + \tilde{Q}) \bar{v} \quad (4)$$

则改进后的简化算法中, 新顶点的最佳位置可由极小化式(4)确定。由于 \tilde{Q} 也是对称矩阵, 因此本文算法极小化求解过程与 QEM 算法相同。

由于 $\Delta_g(\bar{v})$ 表示 \bar{v} 到原始网格的误差, 而 $\Delta_l(\bar{v})$ 为 \bar{v} 到渐进简化过程中相邻网格的误差, 因此一般来说, 当网络的某个区域进行了若干步的简化后, 式(4)中 $\Delta_g(\bar{v})$ 要大于 $\Delta_l(\bar{v})$ 。特别地, 在曲率变化较大区域, $\Delta_g(\bar{v})$ 要远大于 $\Delta_l(\bar{v})$, 从而导致在极小化式(4)过程中 $\Delta_g(\bar{v})$ 占据主导作用。同时, 在相同简化次数条件下, 曲率变化较大区域的全局误差要比平坦区域大得多, 从而利用极小化式(4)的方法可以有效提高曲率变化较大区域的折叠代价, 在相同简化率下保持更多的细节特征, 同时也降低了简化误差。

2.3 边界约束

很多网格模型都具有边界, 模型简化时应保持它们的边界形状。若某条边仅有一个顶点为边界点, 则称该边为准边界边, 如图 3 中边 e_4 。若两个顶点均为边界顶点, 则称该边为边界边, 如图 3 中边 e_1, e_2 和 e_3 。对于这两种边界边的简化, 不仅要考虑简化误差, 还要考虑边界形状保持。保持边界形状的关键是使得简化后的新顶点不会使得边界变形或者变形很小, 因此在第 1 节 3) 中, 以边界的形状保持为主要因素来计算折叠后的新顶点。

计算准边界边的简化误差时, 直接取该边上的边界点作为简化新顶点。如图 3 为例, 计算边界边 e_2 的简化误差时, 首先计算 e_2 和与其相连的两条边界边 e_1 和 e_3 之间的夹角, 分别记为 $\angle(e_1, e_2)$ 和 $\angle(e_2, e_3)$, 给定边界形状阈值 ε , 当 $\cos \angle(e_1, e_2) < \varepsilon$ 时, 则认为 e_1 与 e_2 之间的顶点 A 为可简化点, 否则为不可简化点, 用同样的方法可以判断 e_2 与 e_3 之间的顶点 B 是否为可简化点。当 A 和 B 点均不可简化时, 规定边 e_2 为无效边, 即不参与边折叠运算; 若 A 和 B 两顶点均为可简化点, 则取边 e_2 中点为简化后的新顶点; 若仅有 $A(B)$ 点为可简化点, 则

直接取 $B(A)$ 点作为简化后的新顶点。

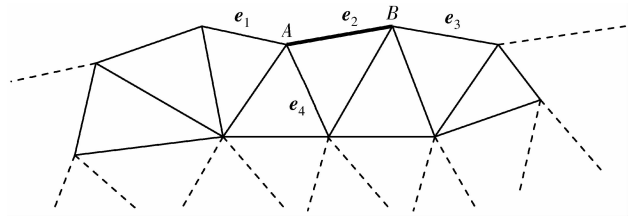


图 3 网格边界简化

Fig. 3 Mesh boundary simplification

3 实验结果

为了验证算法的效果, 用 Visual C++ 实现了算法, 并将本文算法与 QEM 算法进行了比较, 实验中 QEM 算法是利用软件 Qslim v2.0^[13] 实现的。对几种网格模型的简化结果如图 4 所示, 图中百分比为简化后的三角片数量与初始网格的比值。

从图 4 中可以看到, 应用 QEM 算法得到的简化模型网格分布十分均匀, 这样往往使得较为平坦的区域占用过密的网格, 在同样的简化比例下, 就会使得在网格曲率变化较大区域发生较多的简化, 从而会丢失一些网格细节特征。本文算法能够按模型表面的曲率变化合理分布网格, 使得平坦区域网格稀疏, 而曲率变化大的区域网格稠密, 在一定程度上能够保留模型的细节特征。如图 4(c) Bunny 模型的耳和底部(见图 5), 图(f) Armadillo 模型的 4 肢, 图(i) Cow 模型的尾巴、腿和眼, 图(l) Foot 模型的脚趾、后跟等, 这些区域与 QEM 算法简化模型相比, 网格分布更为合理, 细节特征较为明显。图 5 是 Bunny 模型简化的耳和底部细节特征比较结果。

利用 Metro^[14] 工具对简化后的网格进行误差估计, 令模型 M_1 到模型 M_2 的距离为 $E(M_1, M_2) = \max_{p \in M_1} d(p, M_2)$, 平均距离为 $E_m(M_1, M_2) = \frac{1}{|M_1|} \int_{M_1} d(p, M_2) dp$, 其中 $d(p, M) = \min_{p' \in M} d(p, p')$ 为顶点 p 到网格 M 的距离, $d(p, p')$ 为空间两点 p 和 p' 的欧氏距离, $R(M_1, M_2)$ 表示全部 $d(p, M_2)$ ($p \in M_1$) 的均方差。由于上述几种统计值的计算对模型 M_1, M_2 并不对称, 为便于误差比较, 分别统计模型 M_1 和 M_2 的最大距离(即 Hausdorff 距离): $M_{\max} = \max \{ E(M_1, M_2), E(M_2, M_1) \}$, 最大平均距离为 $M_{\text{mean}} = \max \{ E_m(M_1, M_2), E_m(M_2, M_1) \}$, 以及最大

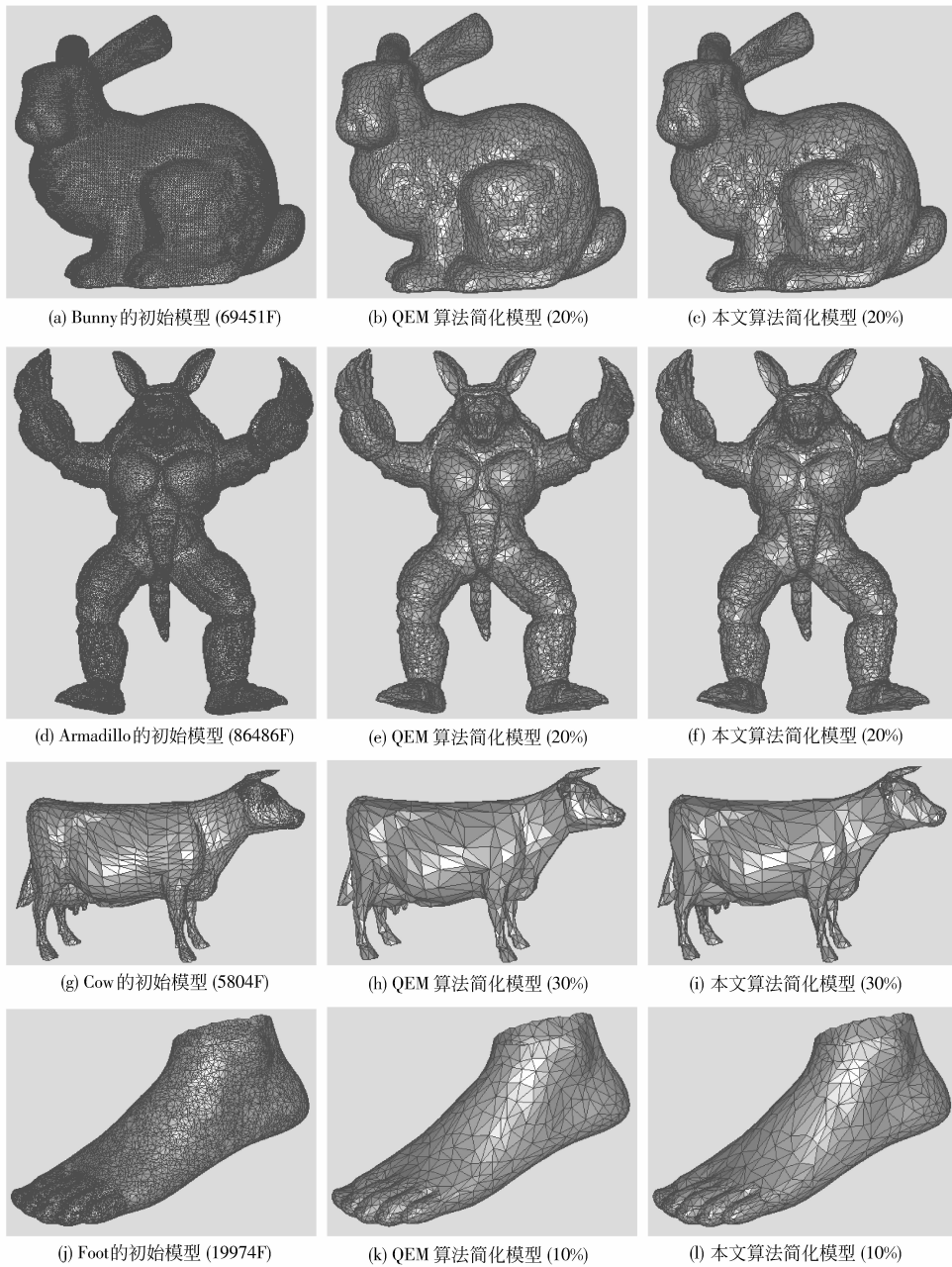


图 4 网格简化模型比较

Fig. 4 Comparison of simplified models

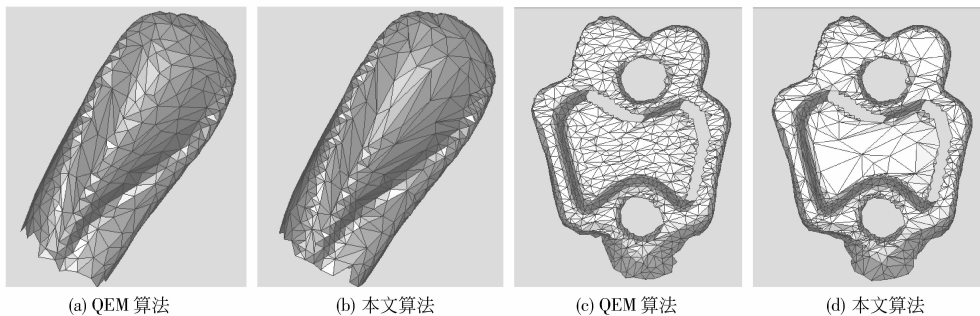


图 5 Bunny 模型简化网格细节特征比较

Fig. 5 Detail feature comparison of simplified Bunny model

标准差 $M_{rms} = \max \{R(M_1, M_2), R(M_2, M_1)\}$ 。表 1 给出了本文算法与 QEM 算法的上述几种简化误差比较。从表 1 中可以看出,本文算法的简化误差,包括最大误差、平均误差以及误差均方差都要明显优于原始的 QEM 算法。

在时间复杂度方面,本文算法较 QEM 算法在更新受影响边的简化代价时,增加了对支撑域误差矩阵的计算,因而算法处理时间较多。另外,为了避免重复性计算,还需要增加一些额外内存来存储初始网格上每个三角片的面积和平面参数。

表 1 图 4 中几种模型的简化误差比较

Tab.1 Error comparison of simplified models in Fig. 4

模型	QEM 算法			本文算法		
	M_{max}	M_{mean}	M_{rms}	M_{max}	M_{mean}	M_{rms}
Bunny(20%)	6.14×10^{-3}	5.61×10^{-4}	7.33×10^{-4}	5.37×10^{-3}	5.52×10^{-4}	7.11×10^{-4}
Armadillo(20%)	3.53×10^{-1}	3.96×10^{-2}	5.05×10^{-2}	2.89×10^{-1}	3.68×10^{-2}	4.71×10^{-2}
Cow(30%)	9.87×10^{-3}	7.02×10^{-4}	9.12×10^{-4}	4.57×10^{-3}	6.30×10^{-4}	8.41×10^{-4}
Foot(10%)	6.57×10^{-1}	7.46×10^{-2}	9.81×10^{-2}	6.49×10^{-1}	6.51×10^{-2}	8.30×10^{-2}

4 结 论

在提出网格支撑域概念的基础上,通过在误差测度中增加支撑域的误差矩阵,对原有 QEM 算法进行了改进。实验结果表明,改进后的简化算法使得简化误差得到较明显的改善,同时初始网格的细节特征得到较多的保留。

由于支撑域建立了新顶点与初始网格之间的联系,能否利用支撑域减小模型外观属性的特征误差也是值得探讨的。在网格简化速度方面,DeCoro 等人^[15]利用 GPU 对点聚类简化算法实现了实时性运算,如何引入 GPU 对本文算法进行加速将是进一步研究的问题。

参考文献 (References)

- [1] He Huiguang, Tian Jie, Zhang Xiaopeng, et al. A survey on mesh simplification[J]. Journal of Software, 2002, 13(12): 2215-2224. [何晖光,田捷,张晓鹏,等. 网格模型化简综述[J]. 软件学报, 2002, 13(12): 2215-2224.]
- [2] Cignoni P, Montani C, Scopigno R. A comparison of mesh simplification algorithms[J]. Computers & Graphics, 1998, 22(1): 37-54.
- [3] Hoppe H, DeRose T, Duchamp T, et al. Mesh optimization [C]//Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques. New York, USA: ACM, 1993: 19-26.
- [4] Garland M, Heckbert P S. Surface simplification using quadric error metrics [C]//Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. New York, USA: ACM, 1997: 209-216.
- [5] Wu Y, He Y, Cai H. Qem-based mesh simplification with global geometry features preserved [C]//Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia. New York, USA: ACM, 2004: 50-57.
- [6] Kim S, Kim C, Levin D. Surface simplification using a discrete curvature norm[J]. Computers & Graphics, 2002, 26(5): 657-663.
- [7] Lee C H, Varshney A, Jacobs D W. Mesh saliency [C]//Proceedings of SIGGRAPH. New York, USA: ACM, 2005: 659-666.
- [8] Li Xianmin, Li Guiqing, Zhang Xiaoling, et al. Edge collapse simplification based on subdivision [J]. Journal of Computer Aided Design and Computer Graphics, 2002, 14(1): 8-13. [李现民,李桂清,张小玲,等. 基于子分规则的边折叠简化方法[J]. 计算机辅助设计与图形学学报, 2002, 14(1): 8-13.]
- [9] Liu Xiaoli, Liu Zeyi, Gao Pengdong, et al. Edge collapse simplification based on sharp degree [J]. Journal of Software, 2005, 16(5): 669-675. [刘晓利,刘则毅,高鹏东,等. 基于尖特征度的边折叠简化算法[J]. 软件学报, 2005, 16(5): 669-675.]
- [10] Hussain M, Okada Y, Nijima K. Efficient and feature-preserving triangular mesh decimation [J]. Journal of WSCG, 2004, 12(1): 167-174.
- [11] Tang Hui, Luo Limin, Zhou Zhengdong. A mesh simplification method based on curvature of curves [J]. Journal of Image and Graphics, 2008, 13(11): 2224-2230. [唐慧,罗立民,周正东. 基于曲线曲率的网格简化方法[J]. 中国图象图形学报, 2008, 13(11): 2224-2230.]
- [12] Lu Wei, Zeng Dinghao, Pan Jingui. Mesh simplification for 3D models with feature-preserving [J]. Journal of Software, 2009, 20(3): 713-723. [卢威,曾定浩,潘金贵. 支持外观属性保持的三维网格模型简化[J]. 软件学报, 2009, 20(3): 713-723.]
- [13] Garland M. Qslim v2.0 [CP/OL]. (2009-01-02) [2009-10-01]. <http://mgarland.org/software/qslim20.html>.
- [14] Cignoni P, Rocchini C, Scopigno R. Metro: measuring error on simplified surfaces [J]. Computer Graphics Forum, 1998, 17(2): 167-174.
- [15] DeCoro C, Tatarchuk N. Real-time mesh simplification using the gpu [C]//Proceedings of Symposium on Interactive 3D Graphics (I3D). New York, USA: ACM, 2007: 161-166.