

结合边分割的改进二次误差测度算法^①



张悠然

(长春大学 网络安全学院, 长春 130022)

通信作者: 张悠然, E-mail: 190701110@mails.ccu.edu.cn

摘要: 为了有效解决二次误差测度算法 (quadric error metrics, QEM) 容易产生异常三角面、失去局部特征、几何结构异常等问题, 提出一种结合边分割的改进二次误差测度算法 (quadric error metrics with edge splitting, ESQEM). 该算法添加顶点高斯曲率作为边折叠代价之一, 通过参数调节模型特征保留情况; 添加边长查询机制, 对细长三角面进行边分割操作. ESQEM 算法能有效维护网格模型高曲率区域特征、保持网格几何结构、消除狭长三角面, 简化后的模型有更好的视觉效果, 高简化率下的简化精度更高.

关键词: 二次误差测度算法; 网格简化; 边分割; 高斯曲率; 边折叠

引用格式: 张悠然. 结合边分割的改进二次误差测度算法. 计算机系统应用, 2022, 31(6): 271-278. <http://www.c-s-a.org.cn/1003-3254/8543.html>

Improved Quadric Error Metrics Algorithm Combining Edge Splitting

ZHANG You-Ran

(School of Cybersecurity, Changchun University, Changchun 130022, China)

Abstract: For the abnormal triangular surfaces, loss of local features, and abnormal geometric structures occurring in quadric error metrics (QEM), an improved quadric error metrics with edge splitting (ESQEM) algorithm is proposed. The ESQEM algorithm adds the Gaussian curvature of vertex as one of the edge collapse costs and adjusts the model feature retention by parameters. The edge length query mechanism is added to perform the edge splitting on the slender triangular surface. The ESQEM algorithm maintains the high-curvature region characteristics of mesh model and keeps the geometric structure, eliminating the slender triangular surface. The simplified model has better visual effect and higher accuracy than QEM algorithm under high simplification rate.

Key words: quadric error metrics (QEM); mesh simplification; edge splitting; Gaussian curvature; edge collapse

三维网格模型在计算机图形学、计算机动画、虚拟现实等领域有着广泛应用, 然而复杂场景的三维模型极大地消耗了带宽资源、增加了处理时间, 不利于模型的存储、计算、渲染等. 因此, 通过使用相对简单的模型代替原始的高细节模型, 即对网格模型进行简化成为当前的主流方案^[1]. 网格简化指通过删除或修改对视觉影响不大的点、边、面来减少网格面片数量^[2], 对几何模型在各领域的应用有着重要意义.

自 Clark^[3] 于 1976 年提出网格简化思想以来, 国

内外学者对简化算法进行了广泛而深入的研究. 其中, 1997 年由 Garland 和 Heckbert 提出的二次误差测度 (QEM) 算法是目前使用最广泛的网格简化算法之一. QEM 算法运算速度快, 简化后的模型质量较高, 是一种兼顾速度、质量和通用性的较理想的算法^[4]. 然而, 该算法仅以点到相关平面距离的二次方作为误差测度, 容易使简化后的模型失去局部特征和几何结构, 并且易产生狭长三角面, 导致渲染效果差. 针对上述问题, 很多学者提出改进方案. Kim 等人^[5] 将离散曲率范数

① 收稿时间: 2021-09-17; 修改时间: 2021-10-14; 采用时间: 2021-10-29; csa 在线出版时间: 2022-05-26

与距离度量结合共同作为边折叠代价,保留了原始网格的高曲率区域. Tang 等人^[6]选择引入反向成本概念增加高、低曲率差异,并选择中点作为新顶点使算法更简单、直观. 王继东等人^[7]把三角形法向量约束因子嵌入误差矩阵中,让体积变化平方作为误差度量,并自动分配网格疏密,保留模型局部特征. Li 等人^[8]提出考虑纹理的边缘折叠改进方法,并提前计算模型边界以保留模型基本拓扑结构. Mao 等人^[9]利用平均二次误差代替累积误差作为折叠代价. Tang 等人^[10]引入高斯曲率作为简化因子,以此保留网格细节特征.

综上所述, QEM 算法的改进方向多是通过优化误差度量来保持三维网格模型特征,没有重点解决简化后模型的三角面异常问题. 基于此,本文提出一种结合边分割的网格简化算法. 实验表明,该算法简化后的模型视觉效果更好,高简化率下的模型精度更高.

1 QEM 网格简化算法

QEM 算法实际上是一种基于边折叠的网格简化算法. 算法通过顶点对的迭代收缩来简化三维模型,即选择合适的边进行折叠,而边折叠顺序依赖于误差.

1.1 边折叠操作

边折叠是一种基于网格边代价的迭代收缩算法. 该算法具有可逆性且折叠代价的计算和更新仅依靠局部信息,使得算法效率较高.

边折叠操作前提是计算三角网格边的折叠代价,每次操作取出队列最顶端的边(即折叠代价最小的边),进行边折叠操作^[11]. 如图 1 所示,边 v_1v_2 为选中的折叠代价最小的边.

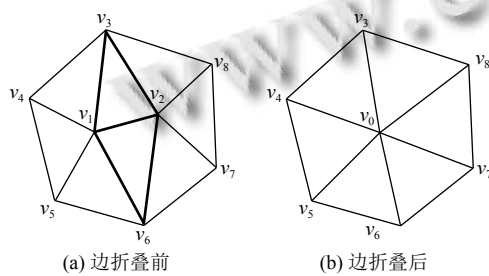


图 1 边折叠示意图

根据边 v_1v_2 的两个端点 v_1 、 v_2 和局部信息计算出误差最小的新顶点 v_0 的位置. 删除顶点 v_1 、 v_2 及两点所在三角网格 $v_1v_2v_3$ 和 $v_1v_2v_6$, 删除相应的边 v_1v_3 、 v_2v_3 、 v_1v_6 、 v_2v_6 . 添加新顶点 v_0 和新生成的边 v_0v_3 、

v_0v_6 , 更新顶点 v_0 的局部顶点信息、边信息的拓扑结构. 重复上述步骤,直至三角网格数量达到要求.

1.2 QEM 算法

QEM 算法中的顶点对迭代收缩是边折叠的扩展,顶点对的折叠代价即为边折叠代价. Garland 等人^[4]将顶点误差定义为点到点所在平面集距离的二次方和,并用二次矩阵保留当前模型的每个顶点近似误差.

故预先对网格中的每一个顶点定义一个 4×4 的误差矩阵 Q , 则顶点 $v=[v_x \ v_y \ v_z \ 1]^T$ 的误差形式确定为其二次项 $(v)=v^T Q v$. 将空间中的每个顶点与其所在平面相关联,定义误差为点与平面集距离的二次方和,如式 (1) 所示:

$$\Delta(v) = \Delta\left([v_x \ v_y \ v_z \ 1]^T\right) = \sum_{p \in \text{planes}(v)} (p^T v)^2 \quad (1)$$

其中, $\text{planes}(v)$ 是包含顶点 v 的所有三角平面集; $p=[a \ b \ c \ d]^T$ 表示由方程 $ax+by+cz+d=0$ 所定义的平面,且 $a^2+b^2+c^2=1$. 由式 (1) 中的误差度量可重写为二次形式:

$$\begin{aligned} \Delta(v) &= \sum_{p \in \text{planes}(v)} (v^T p)(p^T v) \\ &= \sum_{p \in \text{planes}(v)} v^T (pp^T) v \\ &= v^T \left(\sum_{p \in \text{planes}(v)} K_p \right) v \end{aligned} \quad (2)$$

其中, K_p 表示平面 p 的基本二次误差矩阵,如式 (3) 所示:

$$K_p = pp^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (3)$$

通过 K_p 计算空间中任意顶点到该三角面的距离平方,则顶点的二次误差矩阵 Q 可表示为:

$$Q_{(v)} = \sum_{p \in \text{planes}(v)} K_p \quad (4)$$

对于给定的顶点收缩对 $(v_1 \ v_2) \rightarrow \bar{v}$, 则 \bar{v} 的二次误差矩阵和误差公式分别如式 (5) 和式 (6) 所示:

$$Q_{\bar{v}} = Q_{v_1} + Q_{v_2} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix} \quad (5)$$

$$\Delta(\bar{v}) = \bar{v}^T (Q_{v_1} + Q_{v_2}) \bar{v} \quad (6)$$

通过计算 \bar{v} 误差的局部最小值得到新顶点的最佳位置,当 $Q_{\bar{v}}$ 可逆时,计算新顶点 \bar{v} 最佳位置的公式如下所示:

$$\bar{v} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

当 $Q_{\bar{v}}$ 不可逆时,选择 v_1 、 v_2 或 $(v_1+v_2)/2$ 作为新顶点 \bar{v} .新顶点的误差作为对应边的折叠代价,算法依次选取最小代价的边进行折叠,直至无可折叠的边.

综上,QEM算法边折叠代价实则为顶点对 (v_1, v_2) 收缩后产生的新顶点 \bar{v} 的近似误差, \bar{v} 近似误差的取决于 v_1 、 v_2 的近似误差,即点到与其相关的三角面距离的平方和.这种仅依靠距离度量作为近似误差的简化算法会使简化后的模型过于均匀,难以保留局部特征,不适合高精度、高细节模型的简化需求.此外,顶点对的迭代收缩在消除一些三角面的同时会使新顶点所在三角形的边长拉长,容易产生细长三角面,导致简化效果差.因此,根据上述分析将对QEM算法进行改进,改进后的ESQEM算法能较好解决QEM算法所存在的问题.

2 改进 QEM 网格简化算法

针对原始QEM网格简化算法所存在的问题引入高斯曲率算子和边缘分割操作,提出结合边分割的二次误差测度(ESQEM)算法.

首先,只考虑距离度量的QEM算法会使简化后的网格和顶点均匀分布,不利于边缘特征保留.因此将能反应顶点尖锐度的顶点高斯曲率加入二次误差度量中,并提取和保留网格模型边界点,以此维护高曲率区域特征和几何结构.其次,在简化过程中添加边长查询机制,根据网格模型平均边长设置目标长度、边中点作为分割点,对异常三角形进行边分割操作,以此消除不规则的边.

2.1 添加高斯曲率算子

高斯曲率反应了曲面局部弯曲程度,三角网格上某点的高斯曲率反应该点的尖锐程度,因而不均匀区域和特征边缘的高斯曲率较大.本文采用Kim等人^[12]提出的方法来计算网格中内顶点和边界顶点的高斯曲率,顶点高斯曲率与其相连的角度和面积有关,如图2.

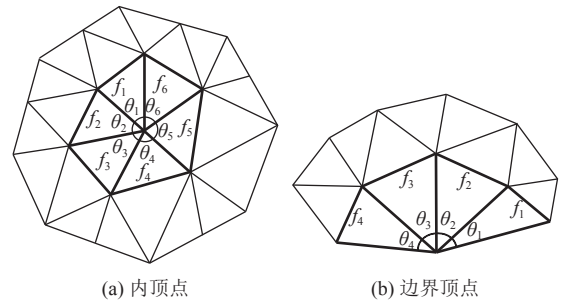


图2 顶点高斯曲率

图2中 θ_i 是三角形 f_i 的内角,三角形面积用 S_i 表示, n 表示与顶点 v 相连的三角形个数.则三角网格中内顶点和边界顶点的高斯曲率计算公式如式(8)、式(9)所示:

$$K(v) = \frac{2\pi - \sum_{i=1}^n \theta_i}{\frac{1}{3} \sum_{i=1}^n S_i} \quad (8)$$

$$K(v) = \frac{\pi - \sum_{i=1}^n \theta_i}{\frac{1}{3} \sum_{i=1}^n S_i} \quad (9)$$

若边 e 由顶点 v_1 和顶点 v_2 组成,而边的高斯曲率与边上的顶点相关^[13],则 e 的高斯曲率可通过式(10)计算:

$$K(e) = |K(v_1)| \times \omega_1 + |K(v_2)| \times \omega_2 \quad (10)$$

其中, $K(v_1)$ 和 $K(v_2)$ 分别代表边 e 两个端点 v_1 和 v_2 的高斯曲率,且权重 ω_i 可通过式(11)计算:

$$\omega_i = \frac{\sum_{i=1}^n s_i(v_i)}{\sum_{i=1}^n s_i(v_1) + \sum_{i=1}^n s_i(v_2)} \quad (11)$$

在QEM算法中,选择最小的边代价进行边缘折叠操作,以获得最佳近似结果.当在边折叠代价中加入高斯曲率算子后,便能有效规避具有显著特征的边被折叠.将二次误差和高斯曲率结合,获得新的边塌陷成本,如式(12)所示:

$$Cost(e) = K(e) \times \omega + Q(e) \quad (12)$$

其中,权重 ω 用来平衡高斯曲率 $K(e)$ 和二次误差 $Q(e)$,且 ω 可通过式(13)计算:

$$w = \frac{a \times Q_{avg}}{K_{avg}} \quad (13)$$

其中, Q_{avg} 和 K_{avg} 分别是所有边的平均二次误差和平均高斯曲率, 而参数 a 用于调整 $K(e)$ 和 $Q(e)$ 的数量级关系, 通常将高斯曲率设为比二次误差小两个数量级。

2.2 边分割策略

边分割是三角剖分的一种经典操作. 实质是将离散点集连接成一定大小的三角形, 而 Delaunay 三角剖分的应用最为广泛. Dyer 等人^[14] 定义非 Delaunay 边为两个对角之和大于 π 的边, 并通过递归分割非 Delaunay 边将任意一个流形三角网格转化为 Delaunay 网格; Liu 等人^[15] 考虑到非 Delaunay 边的蝴蝶形区域提出了一种计算最优分割点的方法. 张顺岚等人^[16] 利用 Delaunay 三角形剖分技术实现了战场仿真模型的局部三角化. 若对已知三维网格中进行边分割操作需要确定要分割的边及分割点的位置。

因此, 为了消除由 QEM 算法产生的狭长三角面, 提出一种基于网格边长的边分割方法. 该方法对简化后网格的边进行遍历以获取边长信息, 根据网格平均边长求出目标长度, 并将目标边的中点作为分割点进行边缘分割操作, 边分割流程如图 3 所示。

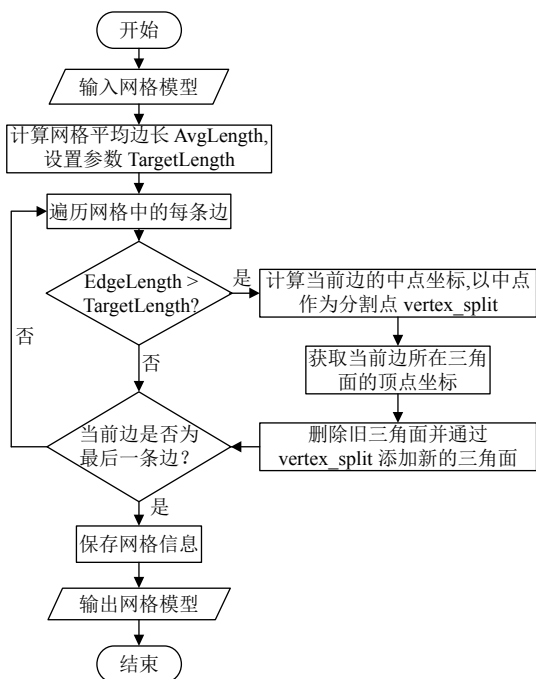


图 3 基于网格边长的边分割流程图

- Step 1. 读取三维网格模型;
- Step 2. 计算网格平均边长 AvgLength, 令目标长

度 TargetLength 为 b 倍 AvgLength. 其中, 倍数 b 可根据实际模型做适当调整;

Step 3. 遍历网格中的每条边, 获取每条边的长度 EdgeLength;

Step 4. 判断边长是否大于目标长度, 若 EdgeLength 大于 TargetLength, 进入 Step 5, 否则跳至 Step 8;

Step 5. 计算当前边的中点位置, 设中点为分割点 vertex_split;

Step 6. 获取当前边所在三角面的顶点坐标 $v[0]$ 、 $v[1]$ 、 $v[2]$ 、 $v[3]$;

Step 7. 删除以点 $v[0]$ 、 $v[1]$ 、 $v[2]$ 和 $v[0]$ 、 $v[2]$ 、 $v[3]$ 组成的 2 个旧三角面, 添加由点 $v[0]$ 、 $v[1]$ 、vertex_split, 点 $v[1]$ 、 $v[2]$ 、vertex_split, 点 $v[2]$ 、 $v[3]$ 、vertex_split 和点 $v[3]$ 、 $v[0]$ 、vertex_split 组成的 4 个新三角面;

Step 8. 判断当前边是否为最后一条边, 如果是则进入 Step 9, 否则跳至 Step 3;

Step 9. 保存边分割后的网格信息;

Step 10. 输出模型。

2.3 算法描述

综上所述, 改进的二次误差测度算法描述如算法 1。

算法 1. ESQEM 算法

- 1) 读取网格文件, 获取原始三维网格信息.
- 2) 遍历网格中的每个顶点, 计算每个顶点的二次误差矩阵 Q .
- 3) 区分网格内顶点和边界顶点, 根据式 (8) 和式 (9) 计算内定点和边界顶点的高斯曲率, 并根据式 (10) 计算每条边的高斯曲率.
- 4) 计算 $Q_{\bar{v}}$ 并判断是否可逆. 如果 $Q_{\bar{v}}$ 可逆, 求新顶点 \bar{v} 的最佳位置, 设置参数 ω , 根据式 (12) 计算当前边的最小代价并插入最小堆中; 如果 $Q_{\bar{v}}$ 不可逆, 分别把 v_1 、 v_2 和 $(v_1+v_2)/2$ 作为新顶点 \bar{v} 计算边折叠代价, 找出 3 种情况的最小边代价, 插入最小堆中.
- 5) 边折叠操作. 从堆顶删除代价最小的边, 添加新顶点 \bar{v} 合并最小代价边的两个端点, 更新 \bar{v} 及近邻边的误差矩阵和高斯曲率, 并返回步骤 4).
- 6) 边分割操作. 遍历网格的每条边, 当 EdgeLength 大于 TargetLength 时, 删除两个旧的三角面, 同时根据顶点和中点位置添加 4 个新的三角面; 当 EdgeLength 小于 TargetLength 时, 继续判断下一条边, 直至循环结束.
- 7) 保存简化后的网格模型.

算法 1 不但可以通过参数 ω 调整网格特征保留情况, 还能通过边长判断异常三角面. TargetLength 通常为 AvgLength 的 4/3 倍, 也可以根据原始网格模型调整最佳 TargetLength 值, 使简化后的三角面更均匀。

3 实验分析

实验分别采用 QEM 算法和 ESQEM 算法对 Cow

模型和 Dinosaur 模型进行简化, 通过对比两种算法的简化效果及简化前后的 Hausdorff 距离来验证 ESQEM 算法在维持模型边缘、避免异常三角面和保留局部特征方面比原始算法更有效。

3.1 简化效果对比

ESQEM 算法是基于 VS 2019 开发平台在 QEM 算法源码基础上进行改进的。为增加可信度, 实验所测

试机相同 (Intel(R) Core(TM) i5-8265U CPU 1.60 GHz, RAM 8 GB), 输入数据相同, 控制参数相同, 且未对改进算法做任何加速处理。为保证在相同简化率的情况下进行比较, 参数 ω 允许的最大误差不超过 0.01 个精度, 参数 b 为固定值。简化后模型以白模和网格两种形式进行整体和局部对比展示, 随着简化率增高, 网格模型信息不断变动, 表 1 为实验过程中的网格数据。

表 1 实验模型数据基本信息

数据名称	简化前点面数(个)	简化方法	简化率(%)	简化后点面数(个)	实验结果图
Cow	$V:2904 F:5804$	ESQEM	58.03	$V:1220 F:2436$	图5 (a)
		QEM	58.03	$V:1220 F:2436$	图6 (a)
		ESQEM	73.23	$V:779 F:1554$	图5 (b)
		QEM	73.23	$V:779 F:1554$	图6 (b)
		ESQEM	82.77	$V:502 F:1000$	图5 (c)
		QEM	82.77	$V:502 F:1000$	图6 (c)
		ESQEM	92.69	$V:214 F:424$	图5 (d)
		QEM	92.69	$V:214 F:424$	图6 (d)
Dinosaur	$V:7026 F:14048$	ESQEM	58.09	$V:2946 F:5888$	图10 (a)
		QEM	58.09	$V:2946 F:5888$	图11 (a)
		ESQEM	83.02	$V:1195 F:2386$	图10 (b)
		QEM	83.02	$V:1195 F:2386$	图11 (b)
		ESQEM	89.28	$V:755 F:1506$	图10 (c)
		QEM	89.28	$V:755 F:1506$	图11 (c)
		ESQEM	93.10	$V:487 F:970$	图10 (d)
		QEM	93.10	$V:487 F:970$	图11 (d)

Cow 的原始模型如图 4 所示。采用两种算法对 Cow 模型进行简化, 结果见图 5 和图 6。由图 4 可知, 简化后的模型边缘始终保持完整, 从图 5(a) 到图 5(c) 简化率由 58.03% 升到 82.77%, 此过程中模型逐渐变得棱角分明, 却并无异常面和异常拓扑产生。当简化率超过 90% 时, 图 5(d) 中牛脚部和面部特征出现模糊, 但曲率更高的牛角处仍然清晰可见, 且未产生细长三角面。对比图 6(d), 当使用 QEM 算法简化率超过 90% 时, 牛脚部及面部特征退化, 而牛角的上半部分直接消失, 此外还产生了多个细长三角面。而图 6(c) 中, 虽然边缘特征仍在, 但出现孔洞及异常三角面。

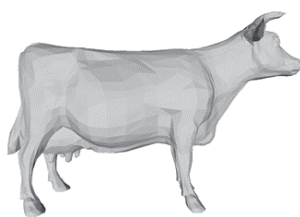


图 4 Cow 原始模型

为了更清晰的观察两种算法的简化效果, 将 Cow 模型的局部位置进行放大比较。如图 7、图 8 所示, 图 7(a) 和图 7(b) 为 Cow 模型在简化率为 58.03% 时的背部对比图, 图 8(a) 和图 8(b) 为 Cow 模型在简化率为 92.69% 时的头部对比图。可以看出, ESQEM 算法能消除异常三角面和保留高曲率特征。

Dinosaur 的原始模型如图 9 所示。图 10 为 ESQEM 算法对 Dinosaur 模型的简化效果图, 图 10(a) 到图 10(c) 的简化率从 58.09% 到 89.28%, 在此过程中模型外表逐渐锋利, 但几何结构与局部特征依旧清晰可辨。而当简化率超过 90% 时, 从图 10(d) 可以看出模型面部、手部等轮廓依旧清晰, 且三角面和拓扑也并无异常。图 11 为 QEM 算法对 Dinosaur 模型的简化效果图, 从图 11(a) 到图 11(d) 的模型轮廓无显著变化, 但在简化过程中产生了细长三角面。并且当简化率超过 90% 时, 由图 11(d) 明显看到, 模型部分手部特征消失, 恐龙的爪子由 3 个退化 2 个。

为了更清晰的对比两种算法的不同效果, 将

Dinosaur 模型的局部位置进行放大比较. 图 12 和图 13 是简化率为 93.10% 时, Dinosaur 模型的背部和手部对比图, 从红圈标注的位置可以看出, ESQEM 算法不仅能更多地保留模型局部特征, 还能将较大的三角面进行分割.

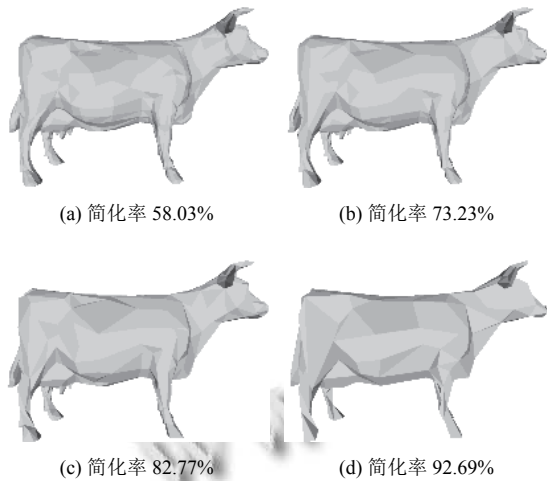


图 5 Cow 模型 ESQEM 算法简化结果

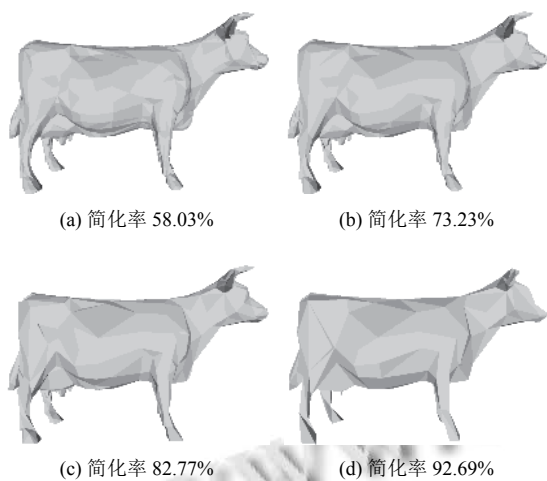


图 6 Cow 模型 QEM 算法简化结果

3.2 Hausdorff 距离对比

Hausdorff 距离是用来测量欧氏空间中两个点集之间距离的一种量度. 在三维网格模型简化过程中, 通常使用 Hausdorff 距离来比较简化前和简化后模型的相似程度, Hausdorff 距离越小, 则简化前后的模型就越相似, 精度就越高, 反之, 简化模型的精度就越低. 因此, 在使用两种不同算法的情况下, 计算简化前后模型的 Hausdorff 距离, 能有效对比 ESQEM 算法和 QEM 算法简化后的模型精度. 图 14 和图 15 为两种算法简化

Cow 模型和 Dinosaur 模型的 Hausdorff 距离值. 其中, 横坐标代表模型简化率, 纵坐标代表 Hausdorff 距离.

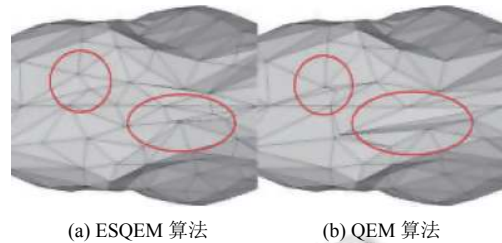


图 7 Cow 模型背部对比图

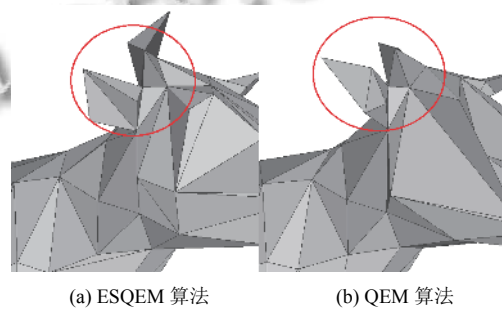


图 8 Cow 模型头部对比图



图 9 Dinosaur 原始模型

由图 14 可知, 当简化率低于 75% 时, 使用 ESQEM 算法得到的模型精度整体略低于 QEM 算法. 尤其当简化率在 35%–75% 时, Hausdorff 距离相差较大. 然而, 当到简化率超过 75% 时, ESQEM 算法的精度开始高于 QEM 算法, 且简化率越大, 精度越高.

图 15 反映了相似的情况, 前期简化率较低时, 两种算法的 Hausdorff 距离值相差不大. 简化率在 50% 左右时, 使用 ESQEM 算法所得的 Hausdorff 距离值出现波动, 模型精度不够稳定. 当简化率超过 70% 时, 使用 ESQEM 算法所得模型精度开始高于 QEM 算法. 综上所述, ESQEM 算法更适合高简化率的简化需求, 模

型所需面数越少, ESQEM 算法较 QEM算法的优势就越大.

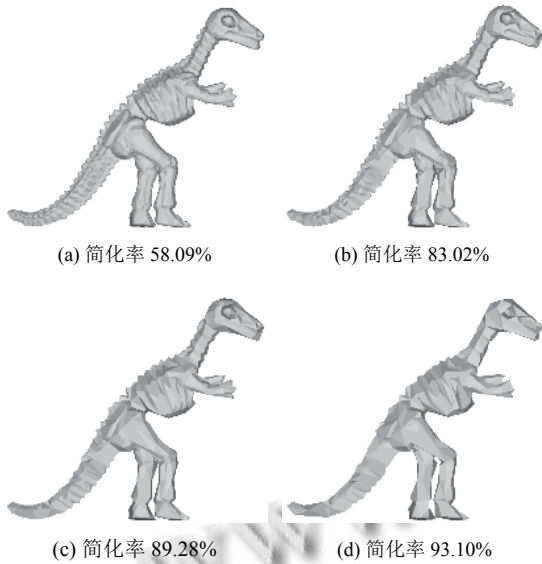


图 10 Dinosaur 模型 ESQEM 算法简化结果

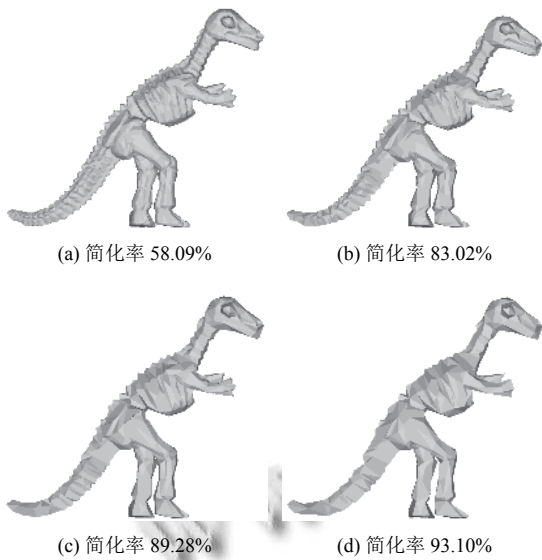


图 11 Dinosaur 模型 QEM 算法简化结果

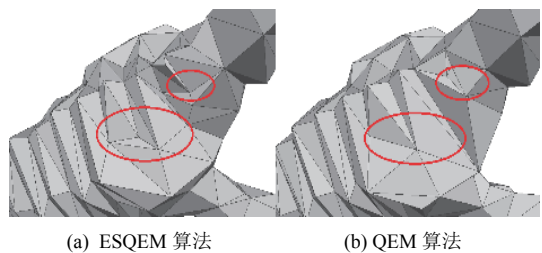


图 12 Dinosaur 模型背部对比图

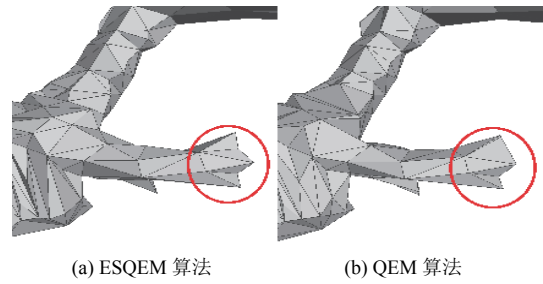


图 13 Dinosaur 模型手部对比图

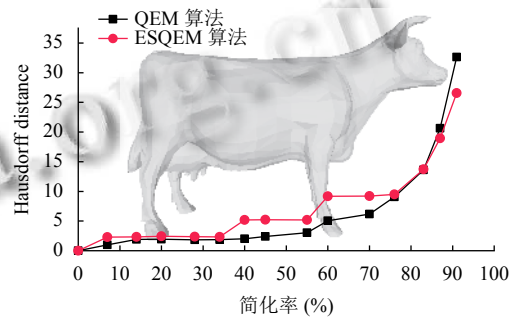


图 14 Cow 的 Hausdorff 距离

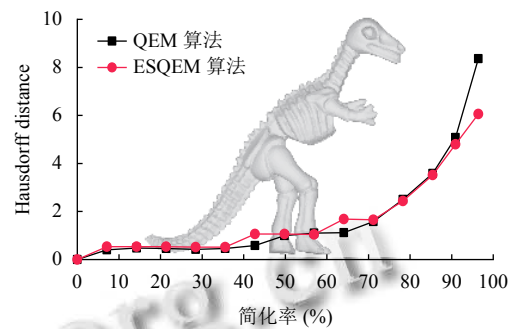


图 15 Dinosaur 的 Hausdorff 距离

4 结论与展望

针对 QEM 算法难以保留简化后模型的细节特征、几何结构和容易产生异常三角面等缺点, 提出一种结合边分割操作的改进算法 (ESQEM). 该算法加入了高斯曲率算子和长边分割操作, 在维持几何结构的基础上, 通过调整参数保留模型局部特征、优化异常三角面. 实验证明 ESQEM 算法具有更好的简化效果, 能更多的保留模型细节、均匀三角面, 并在高简化率的条件下, 简化后的模型精度比 QEM 算法更高. 在接下来的研究中, 将会考虑重新计算边的分割点. 如第 3.2 小节所示, 仅以边中点作为分割点无法在低简化率条件下保证模型精度. 为了在简化过程中始终能得到较高精度的模型, 后续会考虑引入边分割代价, 以保证

新的分割点与原始模型相同位置的点最接近。

参考文献

- 1 黄飞鸿. 基于法向量和体积的半边折叠简化算法. 计算机应用研究, 2020, 37(S1): 334–336.
- 2 董方敏, 张蕊, 刘勇, 等. 网格模型简化算法综述. 中国学术期刊文摘, 2007, 13(14): 8.
- 3 Clark J. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, 1976, 19(10): 547–554. [doi: [10.1145/360349.360354](https://doi.org/10.1145/360349.360354)]
- 4 Garland M, Heckbert P S. Surface simplification using quadric error metrics. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. New York: ACM, 1997. 209–216. [doi: [10.1145/258734.258849](https://doi.org/10.1145/258734.258849)]
- 5 Kim SJ, Kim CH, Levin D. Surface simplification using a discrete curvature norm. *Computers & Graphics*, 2002, 26(5): 657–663. [doi: [10.1016/S0097-8493\(02\)00121-8](https://doi.org/10.1016/S0097-8493(02)00121-8)]
- 6 Tang Z, Yan S, Lan C. A new method of mesh simplification algorithm based on QEM. *Information Technology Journal*, 2010, 9(2): 391–394. [doi: [10.3923/itj.2010.391.394](https://doi.org/10.3923/itj.2010.391.394)]
- 7 王继东, 张芸, 杨斌. 一种新的边折叠网格模型简化算法. 计算机工程与应用, 2013, 49(1): 195–198. [doi: [10.3778/j.issn.1002-8331.1105-0364](https://doi.org/10.3778/j.issn.1002-8331.1105-0364)]
- 8 Li XX, Wan WG, Wang L. Using canny algorithm in QEM simplification for textured 3D models. *IET International Communication Conference on Wireless Mobile and Computing (CCWMC 2011)*. Shanghai: IET, 2011. 315–319. [doi: [10.1049/cp.2011.0900](https://doi.org/10.1049/cp.2011.0900)]
- 9 Mao YL, Yang J, Zhu BP, *et al.* A new mesh simplification algorithm based on quadric error metric. 2015 IEEE 5th International Conference on Consumer Electronics-Berlin (ICCE-Berlin). Berlin: IEEE, 2015. 463–466. [doi: [10.1109/ICCE-Berlin.2015.7391311](https://doi.org/10.1109/ICCE-Berlin.2015.7391311)]
- 10 Tang Y, Zhang QC. Edge-collapse mesh simplification method based on Gauss curvature. 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing. Dalian: IEEE, 2011. 660–662. [doi: [10.1109/iThings/CPSCCom.2011.93](https://doi.org/10.1109/iThings/CPSCCom.2011.93)]
- 11 李世俊, 姜晓彤, 唐慧. 保持细节特征的带纹理模型的高质量简化算法. 计算机应用研究, 2020, 37(1): 300–303, 312. [doi: [10.19734/j.issn.1001-3695.2018.04.0481](https://doi.org/10.19734/j.issn.1001-3695.2018.04.0481)]
- 12 Kim SJ, Jeong WK, Kim CH. LOD generation with discrete curvature error metric. *Proceedings of 2nd Korea Israel Bi-National Conference on Geometrical Modeling and Computer Graphics in the WWW Era*. 1999. 97–104.
- 13 Liang YQ, He FZ, Zeng XT. 3D mesh simplification with feature preservation based on whale optimization algorithm and differential evolution. *Integrated Computer-Aided Engineering*, 2020, 27(4): 417–435. [doi: [10.3233/ICA-200641](https://doi.org/10.3233/ICA-200641)]
- 14 Dyer R, Zhang H, Möller T. Delaunay mesh construction. *Proceedings of the 5th Eurographics Symposium on Geometry Processing*. Goslar: Eurographics Association, 2007. 273–282. [doi: [10.2312/SGP/SGP07/273-282](https://doi.org/10.2312/SGP/SGP07/273-282)]
- 15 Liu YJ, Xu CX, Fan D, *et al.* Efficient construction and simplification of Delaunay meshes. *ACM Transactions on Graphics (TOG)*, 2015, 34(6): 1–13. [doi: [10.1145/2816795.2818076](https://doi.org/10.1145/2816795.2818076)]
- 16 张顺岚, 刘钊, 韩传久. 基于点删除网格简化算法在战场仿真中的应用. 计算机系统应用, 2005, 14(4): 28–31. [doi: [10.3969/j.issn.1003-3254.2005.04.008](https://doi.org/10.3969/j.issn.1003-3254.2005.04.008)]