



论文

一种生成渐进动画网格模型的改进算法

张世学^{①②*}, 赵金字^①, 吴恩华^{②,③}

① 中国科学院长春光学精密机械与物理研究所, 长春 130033

② 澳门大学电脑及资讯科技系, 澳门

③ 中国科学院软件研究所, 计算机科学国家重点实验室, 北京 100080

* 通信作者. E-mail: shixuezhang@tom.com

收稿日期: 2009-11-06; 接受日期: 2009-12-02

国家高技术研究发展计划 (批准号: 2008AA01Z301, 2008AA8080202)、国家自然科学基金 (批准号: 60833007) 和澳门大学博士研究生奖学金与研究基金资助项目

摘要 在计算机图形学中, 动画网格模型被广泛地用来表示时变数据, 而模型的渐进网格表示方法经常可以用来加快处理、传输和存储速度. 文中提出一种生成渐进动画网格模型的高效方法. 该方法使用一种改进的基于曲率的二次误差测量准则来计算边折叠代价, 可以有效地保持曲面上更多的局部特征. 同时, 定义了原始动画序列的变形程度权值, 并将其加入到累加的边折叠代价中, 这样动画模型中的动作变形特征就得到了有效的保持. 最后, 对动画序列提出了一种优化算法, 可以有效地减少动画输出的视觉跳变, 提高相邻帧的时间一致性. 实验结果证明, 本方法高效、易于实现, 并且可以在任意细节层次上生成高质量的渐进动画网格模型.

关键词

动画模型
细节层次
渐进网格
网格优化

1 简介

在计算机图形学与虚拟现实, 越来越多的动画网格模型 (也称为变形表面) 被应用到电影、游戏、仿真等领域^[1]. 为了加快视觉处理、传输、存储的速度, 模型与图像的多分辨率表示技术^[2]被广泛地应用在图像处理^[3~5]、数字几何处理等方面. 到目前为止, 已有的网格简化算法大多是针对静态单一模型的, 而对于动画变形网格的相关工作却很少.

对于动画模型的简化, 一种比较直观的方法是对变形序列中的每一帧模型分别进行简化. 这种方法虽然可以在每一帧上产生具有最小误差的近似模型, 但是由于它没有考虑到相邻帧之间的时间一致性, 这样就会在动画输出中产生很不连续的视觉跳变, 这些在模型表面上摆动和颤抖的效果是我们所不期望的. 因此, 静态网格的简化方法不能简单地直接应用在动画网格中.

我们因此提出一种生成高质量渐进动画模型的改进方法. 本方法可以在时间一致性与几何失真上取得较好的折中, 也就是说在最大化时间一致性的同时, 最小化简化模型的几何失真. 我们使用一种基于曲率的二次误差测量来计算边折叠代价, 可以有效地保持表面的局部特征. 同时在计算累加的边收缩代价时, 加入了整个动画序列的变形程度权值, 这样变形较大的区域就得到了有效地保持. 在此

基础上, 我们提出了一种对动画序列的网格优化方法, 通过移动顶点位置来调整三角形形状, 以进一步提高动画输出中相邻帧之间的时间一致性. 实验结果证明, 该方法可以在动画输出的每一帧都产生高质量的简化模型.

本文的其余部分组织如下: 第 2 节介绍变形网格简化相关领域的工作; 第 3 节详细描述本文算法的推导过程, 并且讨论其优越性; 第 4 节给出本文方法的实验结果; 第 5 节进行总结.

2 相关工作

简化算法与 LOD 目前关于网格简化的方法有很多, 大致可以分为以下 5 类: 顶点删除^[6]、顶点聚集^[7]、区域合并^[8]、细分方法和重复边收缩^[9~12]. 关于相关算法更全面的描述可以参考文献^[13, 14]. 在这些方法中, 应用最为普遍的就是基于边收缩的方法, 代表性算法包括 Hoppe^[10] 提出的渐进网格方法 (progressive mesh) 和 Garland 等^[9] 提出的二次误差测量方法 (QEM). 此类方法在模型表面上建立了一个简单的多分辨率框架, 并可以用来对网格进行自适应的细化操作. 传统的网格简化算法可以在静态网格上取得比较好的效果, 但是由于它没有考虑时间一致性, 所以并不能直接应用到动态网格中.

网格优化 有关网格优化的工作有很多, 其中大多数都是针对改善网格质量参数的局部操作. Hoppe 等^[15] 使用最小化能量方程来优化网格. 文献^[16] 介绍了一种使用非线性数字优化过程来驱动顶点移动的方法. Liu 等^[17] 使用全局 Laplace 算子与约束条件对全局网格进行优化.

动画模型的简化 Shamir 等^[18,19] 首先提出对于变形表面的简化方法. 他们设计了一种叫做基于时间的有向循环图 (time-dependant directed acyclic graph, TDAG) 的全局多分辨率结构. TDAG 是一种可以存储顶点有效时间的数据结构, 它可以记录和查询连结性的改变, 动态网格的每一帧模型都被记录在这个统一的有向图之中. 该方法的缺点是所需结构比较复杂, 并且不容易控制. Mohr 与 Gleicher^[20] 提出了一种基于变形敏感删除 (DSD) 的方法. 该方法直接将动态网格中每一帧的 QEM 误差相加, 以得到统一的边折叠代价. 该算法实际上是在整个动画的所有帧中寻找平均值, 所以只有当原始表面变形程度不大时才会产生比较理想的结果. Kircher 和 Garland^[21] 提出了一种具有动态连结性的变形表面多分辨率表示方法. 当前帧的简化模型通过在前一帧模型的一系列顶点交换操作来得到, 多次连续的边折叠操作被看作是一次重聚类过程. 由于在操作过程中连接性的转换, 此方法的效率很高. Huang 等^[22] 提出了一种基于变形误差测量与动态连结性更新的算法, 该方法可以产生比较理想的简化模型, 但是需要比较复杂的结构. 最近, Landreneau 等^[23] 提出了一种基于针对附属骨骼变形的模型简化方法, 但是这种方法也只局限于具有骨骼的模型简化上.

3 算法过程

本文的方法主要由 3 部分组成: (1) 使用一种基于曲率的二次误差测度来计算边折叠代价. (2) 定义了变形网格中的变形程度权值, 以保持变形程度较大区域的特征. (3) 使用网格优化算法来提高动画序列的时间一致性. 接下来详细介绍各个部分.

3.1 计算边收缩代价

到目前为止, 在基于边收缩的简化方法中, 二次误差测量 (QEM) 仍然被广泛认为是最为高效的

方法之一, 所以首先快速回顾一下该算法.

QEM 使用新顶点到与原始边相邻的三角平面的距离平方和作为边折叠代价. 假设顶点 v 是对边 (v_i, v_j) 进行收缩操作后的新顶点, 那么点 v 的几何误差 $\Delta(v)$ 被定义为 v 到特定三角形面片的距离平方 $(\Delta_i^2(v))$ 之和. 用 $\Delta_i^2(v)$ 来表示顶点 v 到第 i 个面片的距离平方. 面片的平面方程表示为 $a_i x + b_i y + c_i z + d_i = 0$, 法向量表示为 $n_i = [a_i \ b_i \ c_i \ d_i]^T$. 其中 $\Delta_i^2(v)$ 可以表示为

$$\Delta_i^2(v) = ([a_i \ b_i \ c_i \ d_i] [x \ y \ z \ 1]^T)^2 = (n_i^T v)^2 = (v^T n_i)(n_i^T v) = v^T (n_i n_i^T) v = v^T K_{pi} v,$$

其中 $v^T = [x \ y \ z \ 1]$, K_{pi} 为

$$K_{pi} = \begin{bmatrix} a_i a_i & a_i b_i & a_i c_i & a_i d_i \\ b_i a_i & b_i b_i & b_i c_i & b_i d_i \\ c_i a_i & c_i b_i & c_i c_i & c_i d_i \\ d_i a_i & d_i b_i & d_i c_i & d_i d_i \end{bmatrix}.$$

Garland 等称 K_{pi} 为二次误差矩阵, 这样, 顶点 v 的误差可以表示为 $\Delta(v) = \sum_{i=1}^m v^T K_{pi} v = v^T Q_v v$, 其中 $Q_v = \sum_{i=1}^m K_{pi}$, m 是邻接的三角形面片的个数. 对边 (v_i, v_j) 进行收缩操作, 假设收缩点为 v_f , 收缩代价 $\Delta(v_f)$ 可以表示为 $\Delta(v_f) = v_f^T [Q_{vi} + Q_{vj}] v_f = v_f^T Q_{ij} v_f$.

在简化过程中, 在执行边收缩操作的同时, 矩阵 Q 也在执行累加操作, 以生成新的替换点.

QEM 方法可以在快速简化的同时, 取得较好的简化结果. 但是, 由于其忽略了模型上一些重要的特征, 所以会导致一些区域的过度简化. 通过在二次误差矩阵中加入曲率与边长权值, 来改进二次误差测量的边折叠代价. 其中曲率描述了顶点的形状特征, 而边长则表示了该边的影响区域.

首先需要计算顶点的法向量, 可以近似表示为与其相邻三角形的法向均值. 假设三角形的 3 个顶点分别为 v_1, v_2 和 v_3 , 则该平面的法向量为

$$n = \begin{pmatrix} v_2 \cdot x - v_1 \cdot x \\ v_2 \cdot y - v_1 \cdot y \\ v_2 \cdot z - v_1 \cdot z \end{pmatrix} \times \begin{pmatrix} v_3 \cdot x - v_2 \cdot x \\ v_3 \cdot y - v_2 \cdot y \\ v_3 \cdot z - v_2 \cdot z \end{pmatrix}.$$

那么顶点 v 的法向量 n_v 可以由下式得到: $n_v = \sum_{p \in \text{planes}(v)} n_p$, 其中 $\text{planes}(v)$ 是与顶点 v 相邻的三角形集合, $n_p = n/|n|$ 是三角形的单位法向量. 通过顶点法向量, 我们使用下式来计算顶点 v 的曲率: $c_v = \max_{p \in \text{planes}(v)} \theta(n_v, n_p)$, 其中 $\theta(n_v, n_p)$ 表示 n_v 与 n_p 之间的夹角. c_v 称为顶点 v 的相对曲率, 表示了该顶点在模型中的几何重要性. 顶点 v 的相对曲率值越高, 表示该点的显著性特征越重要.

接下来我们可以定义边 (v_i, v_j) 的特征值 $F(i, j) = l_{ij}^2 \times [1 + \frac{1}{2}(c_i + c_j)]$, 其中 l_{ij} 表示边 (v_i, v_j) 的长度. 使用该特征值, 可以将 (1) 式修改为 $\Delta'(v_f) = v_f^T Q_{ij} v_f + \lambda \cdot F(i, j)$, 其中 λ 是一个调整特征值影响的参数. 同时定义 $\lambda * F(i, j)$ 为一个 4×4 矩阵:

$$K_f = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{\lambda \cdot F(i, j)} \end{bmatrix},$$

这样可以将 (2) 式修改为 $\Delta'(v_f) = v_f^T [Q_{ij} + K_f] v_f$.

在这种情况下, 由于在边折叠代价中考虑了曲率与边长的影响, 具有较高特征值的边将会延后进行收缩操作, 这样模型中的显著特征就得到了有效地保持.

3.2 变形程度测量

基于 DSD 的算法在处理变形网格时, 简单地将每一帧的边收缩代价相加, 定义某条边 (v_i, v_j) 的收缩代价为 $\text{DSD}_{ij} = \sum_{t=1}^f (v_f^t)^T Q_{ij}^t v_f^t$, 其中 v_f^t 表示在第 t 帧模型中最小化边 (v_i, v_j) 的折叠代价的新顶点, 并且 $Q_{ij}^t = Q_{v_i}^t + Q_{v_j}^t$.

本文在 DSD 收缩代价的基础上, 加入了附加的变形程度权值, 可以更好地保持变形程度较大的区域. 使用不同帧中的边折叠代价变化来测量变形程度信息. 在变形程度较大的区域中, 边折叠变化值一定较大; 相反, 边折叠变化小的区域中, 变形程度也一定较小. 动画序列的变形权值由下式定义: $\sum_{t=1}^f |\Delta_{ij}^t - \bar{\Delta}_{ij}|$, 其中 Δ_{ij}^t 为第 t 帧模型中边 (v_i, v_j) 的折叠代价, $\bar{\Delta}_{ij}$ 为边 (v_i, v_j) 在整个动画序列中折叠代价的均值. 将此变形权值加入到 DSD 代价函数中:

$$\text{cost}_{ij} = \text{DSD}_{ij} + k_d * \sum_{t=1}^f |\Delta_{ij}^t - \bar{\Delta}_{ij}| = \sum_{t=1}^f \Delta_{ij}^t + k_d * \sum_{t=1}^f |\Delta_{ij}^t - \bar{\Delta}_{ij}|,$$

其中 k_d 是在 0~1 之间的系数, 用来调节变形权值的影响.

3.3 网格优化

接下来我们提出了一种针对简化模型序列的网格优化方法, 本方法可以在调整三角形形状的同时, 进一步提高动画输出的时间一致性.

控制三角形形状的基本思想就是对顶点进行重复地平均加权移动操作. 对于网格中的某个顶点 v_i , 假设其每个相邻顶点 v_j 的权值为 λ_{ij} , 那么顶点 v_i 的最优位置 v'_i 为 $v'_i = \sum_{v_j \in N(v_i)} \lambda_{ij} v_j$, 其中 $N(v_i)$ 为 v_i 的相邻顶点集合. 一般需要每个 λ 都大于 0, 并且累加值为 1, $\sum_{v_j \in N(v_i)} \lambda_{ij} = 1$. 这样上式描述了凸合并的情况, 并且 v_i 的最优位置都会在其所有邻居顶点的凸外壳中.

如果将邻居顶点的每个 λ_{ij} 都取为等值, 那么最优点就会落在多边形的质心, 对所有顶点进行重复移位操作之后就会使得三角形趋近于等边三角形. 本文算法将 λ_{ij} 选取为 v_i 相对于 v_j 的平均值坐标^[24], 这样原始的三角形形状就会被很好地保持.

在变形网格序列中, 我们希望相邻帧模型中具有尽可能相似的三角形形状, 这样在动画输出中就会产生最小的视觉跳变. 受到 Summer 等人^[25] 变形传递方法的启发, 在计算某网格 M 的均值坐标权值时, 我们可以使用另一个网格 N 的均值坐标权值来代替, 这样就把 N 中的三角形形状传递到 M 中. 当然这需要两个模型具有完全相同的连接性, 而本文的简化模型序列显然是具有这种属性的.

首先计算出第 1 帧模型中每个顶点相对于其邻居顶点的均值坐标权值, 然后利用此权值, 对第 2 帧模型的顶点进行均值移位操作. 同样道理, 第 2 帧的权值应用到第 3 帧模型顶点中, 以此类推. 这样就把当前帧模型的三角形形状均匀地传递到下一帧模型中. 由于在整个变形序列中, 相邻帧模型具有相同的连接性和相似的三角形形状, 所以动画输出的视觉跳变会大大降低.

3.4 算法步骤

本文的算法步骤可以大致总结为如下:



图 1 Dancer 动画网格模型, 共 201 帧

上行为具有 7061 个顶点的原始模型, 下行为删除 90%面片的模型输出

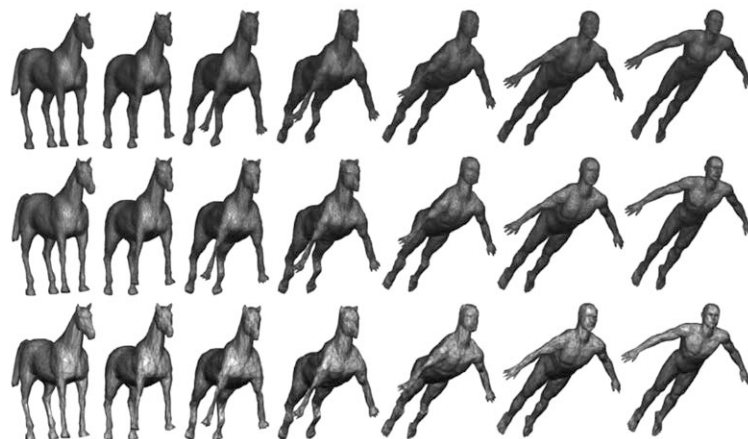


图 2 Horse-to-man 变体动画模型, 共 200 帧

第 1 行: 原始模型 (17489 个顶点), 第 2 行: 具有 3200 个顶点的简化模型 (删除 80 %), 第 3 行: 具有 800 顶点的简化模型 (删除 95%)

- 1) 首先用 3.1 小节描述的方法计算原始网格中的边折叠代价.
- 2) 如 DSD 方法一样, 求出整个动画序列中每条边的累加折叠代价.
- 3) 计算每条边的变形程度权值, 得到整个序列的统一的边收缩顺序.
- 4) 使用重复边收缩操作对每一帧模型进行化简, 直到达到所需的分辨率.
- 5) 使用 3.3 小节的网格优化算法对简化模型序列进行优化.

4 实验结果

我们在 VC2005 和 OpenGL 的编程环境下实现了本文算法, 运行环境为 Pentium4 3.2 GHz CPU 和 4 GB 内存. 图 1 为具有 201 帧的 Dancer 动画模型及其简化输出, 其中简化模型的点面数仅为原始模型的 10 %, 可以看到简化序列中 dancer 每一帧的轮廓特征都被很好的保持. 图 2 中的 horse-to-man 变体动画序列则相对复杂, 动画共有 200 帧, 每个模型具有 17489 个顶点, 其变形过程是完全非刚性的. 第 2 和第 3 行分别显示了简化了 80 %与 95 %的模型输出, 模型中的几何与动作特征都被很好地保持.

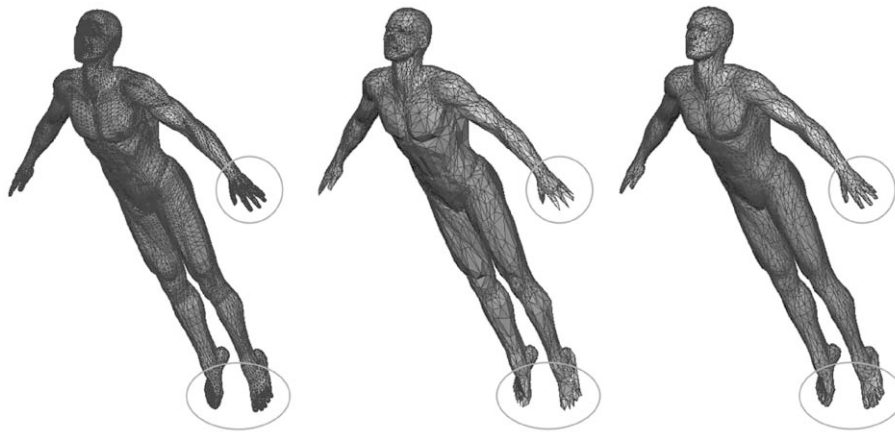


图 3 Horse-to-man 变形动画实验结果对比

简化模型具有 3200 个顶点, 6396 个三角形. 左图为原始模型, 中图为文献 [21] 简化结果, 右图为本文简化结果

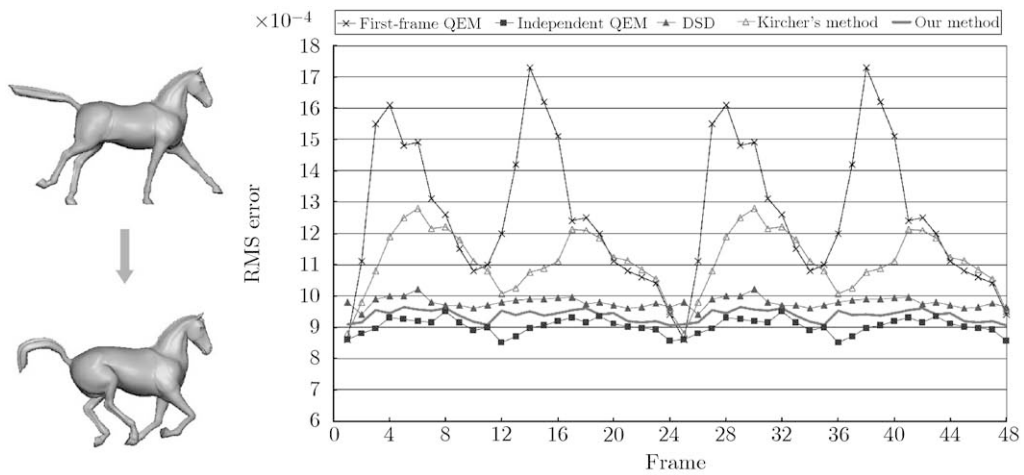


图 4 Horse-gallop 动画 (48 帧) 的误差测量结果

其中水平轴为帧索引值, 垂直轴为误差值

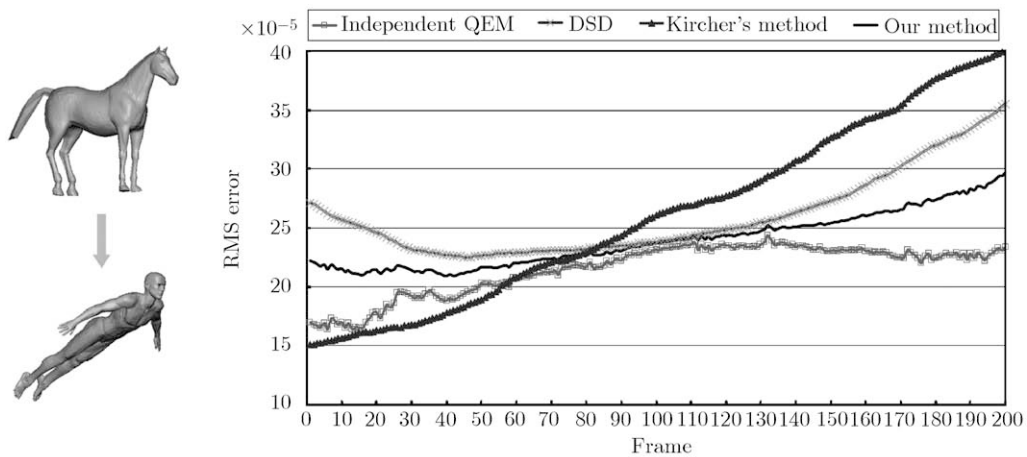


图 5 Horse-to-man 动画 (200 帧) 的误差测量结果

其中水平轴为帧索引值, 垂直轴为误差值

下面对比本文方法与前人方法的实验结果. 图 3 显示了 horse-to-man 变体动画的最后一帧模型, 左图为原始模型, 中图为文献 [21] 的结果, 右图为本文结果. 显然文献 [21] 结果中已经丢失了在人手和脚上的一些轮廓细节特征, 而本文结果则很好地保持了这些特征. 在从 horse 到 man 的变体序列中, 人脚是由马蹄经过非刚性变形转变过来的, 因此该区域的显著性特征与变形程度都较大. 本文方法对此区域的边折叠代价赋予较大的权值, 那么该区域的多数边折叠操作被延迟, 所以本文结果保持了更多的细节特征. 图 4 和 5 显示了在不同方法下, 对 horse-gallop 与 horse-to-man 模型的简化模误差统计. 其中本文结果的误差值都基本保持在独立 QEM 与 DSD 方法之间, 并且在每一帧的误差值都比较均匀, 整体上优于其他方法. 这样, 本文结果在视觉效果与误差统计^[26]上都证明了其优越性.

5 结论

本文提出了一种生成渐进动画网格的改进算法. 给定一组三维时变动画序列, 本方法可以生成任意分辨率下的简化数据. 使用一种改进的二次误差测量来计算边折叠代价, 可以很好地保持局部特征. 同时定义了动画序列的变形程度测量权值, 可以保持变形程度较大的区域的特征. 最后, 提出了网格优化方法, 可以提高动画输出的时间一致性. 本方法高效, 易于实现, 比之前算法能够产生质量更高的输出结果.

参考文献

- 1 李晓勇, 张忠华, 朱伟康. 航天测量船船体变形的数据处理. 光学精密工程, 2009, 17: 445-452
- 2 张麒, 汪源源, 王威琪, 等. 活动轮廓模型和 Contourle 多分辨率分析分割血管内超声图像. 光学精密工程, 2008, 16: 2303-2311
- 3 屈小波, 闫敬文, 杨贵德, 等. 改进拉普拉斯能量和的尖锐频率局部化 Contourle 域多聚焦图像融合方法. 光学精密工程, 2009, 17: 1203-1212
- 4 赵翠莲, 范志坚, 孙亮, 等. 体素连通性的深度图像拉伸特征提取. 光学精密工程, 2009, 17: 930-936
- 5 张玉叶, 周晓东, 王春歆. 应用像素运动模糊特征分割的空间移变降质复原. 光学精密工程, 2009, 17: 1119-1126
- 6 Schroeder W J, Zarge J A, Lorensen W E. Decimation of triangle meshes. In: ACM SIGGRAPH 1992 Conference Proceedings. Chicago, 1992. 65-70
- 7 Low K L, Tan T S. Model simplification using vertex clustering. In: ACM Symposium on Interactive 3D Graphics. New York, 1997. 75-82
- 8 Garland M, Willmott A, Heckbert P S. Hierarchical pace clustering on polygonal surfaces. In: Proceedings of ACM Symposium Interactive 3D Graphics. New York, 2001. 49-58
- 9 Garland M, Heckbert P S. Surface simplification using quadric error metrics. In: ACM SIGGRAPH 1997 Conference Proceedings. New York, 1997. 209-216
- 10 Hoppe H. Progressive meshes. In: ACM SIGGRAPH 1996 Conference Proceedings. New Orleans, 1996. 99-108
- 11 Lee A, Moreton H, Hoppe H. Displaced subdivision surfaces. In: ACM SIGGRAPH 2000 Conference Proceedings. Los Angeles, 2000. 85-94
- 12 Yan J, Shi P, Zhang D. Mesh simplification with hierarchical shape analysis and iterative edge contraction. IEEE Trans Visual Comput Graph, 2004, 10: 142-151
- 13 Luebke D, Reddy M, Cohen J. Level of Detail for 3-D Graphics. Morgan Kaufmann, 2002
- 14 Oliver M. van K, Hlio P. A comparative evaluation of metrics for fast mesh simplification. In: Computer Graphics Forum, 2006. Vol 25. 197-210
- 15 Hoppe H, DeRose T, Dunchamp T, et al. Mesh optimization. In: ACM SIGGRAPH 1993 Conference Proceedings. Anaheim, 1993. 19-25
- 16 Garimella R, Shashkov M. Polygonal surface mesh optimization. Engin Comput, 2004, 20: 265-272
- 17 Liu L, Tai C, Ji Z, et al. Non-iterative approach for global mesh optimization. Comput Aid Des, 2007, 39: 772-782

- 18 Shamir A, Bajaj C, Pascucci V. Multiresolution dynamic meshes with arbitrary deformations. In: IEEE Visualization 2000 Conference Proceedings. Salt Lake City, 2000. 423–430
- 19 Shamir A, Pascucci V. Temporal and spatial level of details for dynamic meshes. In: Proceedings of ACM Symposium on Virtual Reality Software and Technology. Banff, 2001. 77–84
- 20 Mohr A, Gleicher M. Deformation sensitive decimation. Technical Report. University of Wisconsin, 2003
- 21 Kircher S, Garland M. Progressive multiresolution meshes for deforming surfaces. In: Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation. New York, 2005. 191–200
- 22 Huang F C, Chen B Y, Chuang Y Y. Progressive deforming meshes based on deformation oriented decimation and dynamic connectivity updating. In: Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Vienna, 2006. 53–62
- 23 Landreneau E, Schaefer S. Simplification of articulated meshes. In: Proceedings of EUROGRAPHICS. Munich, 2009
- 24 Floater M S. Mean value coordinates. Comput Aid Geometr Des, 2003, 20: 19–27
- 25 Sumner R W, Popovic J. Deformation transfer for triangle meshes. In: ACM SIGGRAPH 2004 Conference Proceedings. Los Angeles, 2004. 399–405
- 26 刘晨, 金施群, 费业泰, 等. 应用空间编码投影法测量三维大物体面形. 光学精密工程, 2008, 16: 2268–2273