

基于边折叠网格简化的三维形状变形

华顺刚*, 钟庆, 李绍帅

(大连理工大学 精密与特种加工教育部重点实验室, 辽宁 大连 116024)

摘要: 提出一种基于边折叠网格简化的交互式三维形状变形算法. 通过折叠三角形网格的边来提取原始三维模型中的形体关键特征点生成简化模型, 以交互方式对三维模型添加固定、移动等位置约束, 建立并求解非线性优化方程, 获得简化模型上各关键点仿射变换的最优值. 然后基于关键点的仿射变换, 计算原始模型中各节点的变形映射. 实例表明边折叠网格简化可有效提取三维形状模型的关键特征点, 减少模型的多边形数量和计算量, 结合非线性优化求解, 实现了直接交互、实时显示和细节保持的三维形状变形.

关键词: 边折叠网格简化; 三维形状变形; 仿射变换; 非线性优化

中图分类号: TP391 **文献标志码:** A

0 引言

三维形状变形技术可以用来改变三维实体模型的局部及整体形状, 使之产生实时、逼真的变形效果, 因此被广泛应用于三维影视动画、医学图像处理 and 虚拟现实等.

早期的关于三维实体变形的工作主要集中在空间变形, Barr^[1] 提出对空间的全局映射, 实现三维实体全局或局部变形; Sederberg 等^[2] 提出了基于网格的自由变形技术, 该技术用一个三维网格对三维实体进行参数化, 是一种有效的获得复杂形状的粗糙变形方法, 但该方法在得到精确的变形时需要一个精细的、手工设计的控制网格和过多的用户操作. 通过直接操作控制网格, Hsu 等^[3] 实现了对三维模型的更多的直观控制, 但用户仍然受限于自由变形算法的可表现性. 这些变形方法主要关注的是模型的变形效果, 而变形过程中采用间接操作控制网格代替三维实体本身的方式, 细节信息不能得到很好的保持.

近几年, 高细节扫描网格技术得到了广泛的应用, 如何通过直观的用户交互来编辑高质量的网格, 同时保持变形细节等问题成为三维形状变形研究的重点. 保持变形表面的细节可以利用多分辨率分解和变分弯曲能最小化方法, 或者通过

操纵微分坐标、求解泊松系统来完成. 多分辨率方法^[4] 可以产生一个简化网格层次和与之相关的细节系数, 很好地保持了细节信息. 基于微分表示的算法^[5,6] 通过利用局部形状属性, 如尺寸、方向和曲率等来表述模型, 将网格编辑转化成为求解能量最小化问题. Botsch 等^[7] 分析了以上两种方法的外在联系和内在限制, 将二者融合成一种具有有效性和灵活性的网格编辑技术. 对于细分平面, Zhou 等^[8] 提出了一种交互变形的算法, 用户可以在三维网格模型表面上任意拾取一个点作为句柄, 通过鼠标移动操作句柄, 使模型的其他部位产生相应动作. Sumner 等^[9] 利用点采样方法在原始模型上均匀抽取特征点, 创建一个与抽样点紧密相关的简化模型, 称为“变形图表”, 通过一系列的刚性变换, 达到模型局部或整体变形的目的.

本文对交互式三维形状变形进行研究, 提出一种基于边折叠网格简化的交互式三维形状变形算法. 文献^[9] 中采用的点采样方法是在形体上均匀采样来生成简化模型, 而本文运用的边折叠三角网格简化算法可有效提取形体的关键特征点, 以期只用较少关键特征点的简化模型就能表述复杂形体, 通过非线性优化快速实现形体变形, 获得高质量的细节保持效果.

1 边折叠三角形网格简化

复杂的三维形状模型通常包含大量的顶点和三角面片,如果直接对原始模型进行变形操作,计算过程复杂且耗时,因此需要先简化网格模型. 三维模型网格简化是指在保持原模型几何形状不变的前提下,采用适当的算法减少该模型的面片数、边数和顶点数^[10]. 根据简化过程中所删除元素的不同,网格简化方法大致可以分为 3 类:基于顶点移除的网格简化、基于面折叠的网格简化和基于边折叠的网格简化^[11]. 其中边折叠三角形网格简化算法通过选定一条有向边以及它所连接的两个顶点,将其中较不重要的一个顶点“折叠”至另一个顶点,最后修改网格拓扑关系来完成网格简化.

边折叠算法原理如图 1 所示. 当图 1(a)中有向边 e 的起点 u “折叠”至终点 v 时,有向边 e 、起点 u 以及三角形 f_1 和 f_2 被删除,形成图 1(b)所示的简化拓扑结构. 在简化过程中,如何找到一条有向边,使得删除该边对三维模型整体的影响最小是该算法的重点. 这里可引入边折叠代价这一概念,边折叠代价评价了有向边在拓扑结构中的重要性,在本文中定义为边的长度与其起点重要度的乘积. 顶点重要度反映该顶点的局部区域表面变化情况,若多个多边形共面度越高,采用一个平面取代它们对三维模型整体所产生的影响会越小. 本文利用被折叠三角形和保留三角形的面法线来定义顶点重要度,从而反映顶点 u 附近表面折叠前后的变化情况. 顶点重要度定义如下:

$$\lambda_u = \sum_{t_n \in (T(u)-T(uv))} \min_{f_n \in T(uv)} \{1 - N_{t_n} \cdot N_{f_n}\} \quad (1)$$

式中: $T(u)$ 和 $T(uv)$ 分别为包含顶点 u 的三角形集合和同时包含顶点 u 、 v 的三角形集合; N_{t_n} 为三角形 t_n 的面法线; N_{f_n} 为三角形 f_n 的面法线.

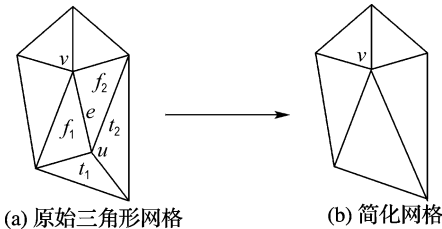


图 1 边折叠三角形网格简化示意图

Fig. 1 Schematic diagram of edge collapse triangle mesh simplification

点积 $N_{t_n} \cdot N_{f_n}$ 定义了三角形 t_n 和 f_n 的共面度,其值越大,两三角形越共面, λ_u 值越小. 这两个

三角形被另一平面代替对整个模型的影响越小,边折叠代价越小,移除 e_{uv} 对整个三维模型的影响越小, e_{uv} 越可能被删除;反之,则越应当保留此边.

本文中有向边 e_{uv} 的边折叠代价为

$$C_{e_{uv}} = \|v - u\| \cdot \lambda_u; v \in A(u) \quad (2)$$

式中: $\|v - u\|$ 为 u 、 v 两顶点的欧式距离; $A(u)$ 为与顶点 u 相连的顶点的集合.

考虑到边的长度和起点的重要度这两方面因素,上述边折叠代价既反映了两顶点之间的距离,又结合了模型局部表面几何变化. 在操作过程中,每次计算所有边的折叠代价,将边折叠代价最小的那条边删除,直到达到要求的顶点数. 这样可以有效地保留模型的关键信息,从而使原始模型的特征得到较好的保持. 图 2 为利用边折叠三角形网格简化算法对三维马模型进行简化的示例,原始模型图 2(a)的顶点数为 48 485,简化后图 2(b)的顶点数为 200. 由图 2(b)可以看出,马模型的外部轮廓和内部细节特征均得到了有效的保持.

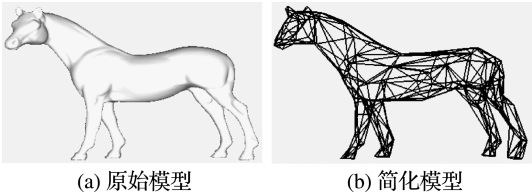


图 2 三维形状模型网格简化

Fig. 2 3D shape mesh simplification

简化模型中节点的布局应当基本符合变形前形体的形状,其中节点的数量、是否为关键特征点等因素决定了模型的可表现性. 一般而言,形体的顶点数越多,为表达细节变形,所需的简化模型节点数量也越多,这样未知变量的个数也越多,计算代价会增加,因此关键特征点对形体变形至关重要. 若尽量使用有意义的关键特征点来表征简化模型,则用较少的节点就能近似表达复杂的形体,在获得好的变形效果的同时,变形计算的成本代价减小.

2 形状模型仿射变换

在简化模型的基础上,对三维模型实施整体变形. 本文称简化模型中的顶点为关键点,并为每个关键点指定一个仿射变换. 设第 i 个关键点的仿射变换矩阵是由一个 3×3 的旋转矩阵 R_i 和一个 3×1 的平移向量 T_i 组成. 原始模型的各项点根据仿射变换矩阵映射到新的位置,完成整体变形. 如图 3 所示,设 v 是原始模型中的顶点,根据第 i

个关键点的仿射变换,按照如下公式映射到新位置 $\tilde{\mathbf{v}}$:

$$\tilde{\mathbf{v}} = \mathbf{R}_i(\mathbf{v} - \mathbf{P}_i) + \mathbf{P}_i + \mathbf{T}_i; i \in \{1, \dots, m\} \quad (3)$$

式中: $\mathbf{P}_i$ 为简化模型中第 i 个关键点位置向量; m 为简化模型中关键点个数。

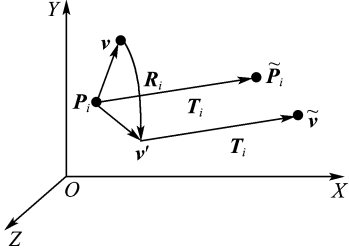


图3 节点仿射变换

Fig. 3 Nodes affine transformation

首先在旋转矩阵 $\mathbf{R}_i$ 作用下,原始模型的顶点 $\mathbf{v}$ 根据关键点 $\mathbf{P}_i$ 变换到点 $\mathbf{v}'$,其次在平移向量 $\mathbf{T}_i$ 作用下变换到 $\tilde{\mathbf{v}}$;对于关键点 $\mathbf{P}_i$ 本身, $(\mathbf{v} - \mathbf{P}_i)$ 为 $\mathbf{0}$ 向量,只根据 $\mathbf{T}_i$ 做平移运动到 $\tilde{\mathbf{P}}_i$ 。

由于式(3)只需要已知模型顶点坐标即可进行,对任意以顶点信息给出的三维形体均适用。对于原始模型中的各顶点,所有关键点的仿射变换对其均有影响。然而为了提高计算速度,保证变形效果流畅,只考虑将离该顶点最近的 K 个关键点的影响进行平滑融合,式(3)修改为式(4):

$$\tilde{\mathbf{v}}_j = \frac{1}{K} \sum_{i \in A(j)} (\mathbf{R}_i(\mathbf{v}_j - \mathbf{P}_i) + \mathbf{P}_i + \mathbf{T}_i); j \in \{1, \dots, V\} \quad (4)$$

式中: $A(j)$ 为离顶点 $\mathbf{v}_j$ 最近的 K 个关键点集合; V 为三维形状顶点个数。

3 形状变形的优化求解

上述原始模型中各顶点的映射计算均基于关键点的旋转矩阵 $\mathbf{R}_i$ 和平移向量 $\mathbf{T}_i$,因此本章将讨论关键点仿射变换矩阵的求解问题。采用交互方式,被选中的关键点直接根据鼠标拖动移动到指定位置,其余关键点的仿射变换矩阵可通过构造优化目标函数进行求解来获得。参考文献[9],本文的优化目标函数形式如下:

$$\min_{\mathbf{R}_1, \mathbf{T}_1, \dots, \mathbf{R}_m, \mathbf{T}_m} \omega_{\text{reg}} E_{\text{reg}} + \omega_{\text{rot}} E_{\text{rot}}$$

式中: $E_{\text{reg}} = \sum_{i=1}^m \sum_{l \in N(i)} \|\mathbf{R}_i(\mathbf{P}_l - \mathbf{P}_i) + \mathbf{P}_i + \mathbf{T}_i - (\mathbf{P}_l + \mathbf{T}_i)\|_2^2$, $E_{\text{rot}} = \sum_{i=1}^m ((\mathbf{c}_1^i \cdot \mathbf{c}_2^i)^2 + (\mathbf{c}_1^i \cdot \mathbf{c}_3^i)^2 +$

$(\mathbf{c}_2^i \cdot \mathbf{c}_3^i)^2 + (\mathbf{c}_1^i \cdot \mathbf{c}_1^i - 1)^2 + (\mathbf{c}_2^i \cdot \mathbf{c}_2^i - 1)^2 + (\mathbf{c}_3^i \cdot \mathbf{c}_3^i - 1)^2)$; ω_{reg} 为规则化项误差 E_{reg} 的权重系数, ω_{rot} 为旋转项误差 E_{rot} 的权重系数; $N(i)$ 为与关键点 i 相连的所有关键点的集合; $\mathbf{c}_1^i$ 、 $\mathbf{c}_2^i$ 和 $\mathbf{c}_3^i$ 分别是 $\mathbf{R}_i$ 的3个列向量, $\mathbf{c}_1^i = (r_{11}^i \ r_{21}^i \ r_{31}^i)^T$, $\mathbf{c}_2^i = (r_{12}^i \ r_{22}^i \ r_{32}^i)^T$, $\mathbf{c}_3^i = (r_{13}^i \ r_{23}^i \ r_{33}^i)^T$, $i \in \{1, \dots, m\}$ 。

利用规则化项误差 E_{reg} 将第 i 个关键点对第 l 个关键点的影响与第 l 个关键点自身的仿射变换相协调,使变形更加平滑、柔和;为使简化模型中各关键点尽量作纯旋转变换,引入旋转项误差 E_{rot} ,从而可有效地保留三维形状的细节信息。

优化目标函数中包含变量 $\mathbf{R}_i =$

$$\begin{pmatrix} r_{11}^i & r_{12}^i & r_{13}^i \\ r_{21}^i & r_{22}^i & r_{23}^i \\ r_{31}^i & r_{32}^i & r_{33}^i \end{pmatrix}, \mathbf{T}_i = (t_1^i \ t_2^i \ t_3^i)^T \text{ 和 } \mathbf{P}_i = (p_1^i \ p_2^i \ p_3^i)^T. \text{ 设函数 } f(\mathbf{x}) \text{ 满足 } f^T(\mathbf{x})f(\mathbf{x}) = \mathbf{F}(\mathbf{x}) = \omega_{\text{reg}} E_{\text{reg}} + \omega_{\text{rot}} E_{\text{rot}}. \text{ 其中所求变量 } \mathbf{x} = (r_{11}^i \ r_{12}^i \ r_{13}^i \ r_{21}^i \ r_{22}^i \ r_{23}^i \ r_{31}^i \ r_{32}^i \ r_{33}^i \ t_1^i \ t_2^i \ t_3^i)^T, \text{ 共 } 12m \text{ 项. 设集合 } N(i) \text{ 元素个数为 } n_i, \text{ 对第 } i \text{ 个关键点, } E_{\text{reg}} \text{ 有 } 3 \times n_i \text{ 个平方项, } E_{\text{rot}} \text{ 有 } 6 \text{ 个平方项. 于是 } f(\mathbf{x}) \text{ 共有 } 3 \times \sum_{i=1}^m n_i + 6 \times m \text{ 项, 可表示为 } f(\mathbf{x}) = (\phi_1^i(\mathbf{x}) \ \phi_2^i(\mathbf{x}) \ \phi_3^i(\mathbf{x}) \ \dots \ \phi_1^i(\mathbf{x}) \ \phi_2^i(\mathbf{x}) \ \phi_3^i(\mathbf{x}) \ \phi_4^i(\mathbf{x}) \ \phi_5^i(\mathbf{x}) \ \phi_6^i(\mathbf{x}) \ \dots)^T, l \in N(i).$$

指定规则化项误差为 $\phi_q^i(\mathbf{x}) = \mathbf{R}_i(\mathbf{P}_l - \mathbf{P}_i) + \mathbf{P}_l + \mathbf{T}_l - (\mathbf{P}_l + \mathbf{T}_l)$, 其中 $q \in \{1, 2, 3\}$, 表示空间坐标的3个分量; 旋转项误差为 $\phi_1^i(\mathbf{x}) = \mathbf{c}_1^i \cdot \mathbf{c}_2^i$, $\phi_2^i(\mathbf{x}) = \mathbf{c}_1^i \cdot \mathbf{c}_3^i$, $\phi_3^i(\mathbf{x}) = \mathbf{c}_2^i \cdot \mathbf{c}_3^i$, $\phi_4^i(\mathbf{x}) = \mathbf{c}_1^i \cdot \mathbf{c}_1^i - 1$, $\phi_5^i(\mathbf{x}) = \mathbf{c}_2^i \cdot \mathbf{c}_2^i - 1$, $\phi_6^i(\mathbf{x}) = \mathbf{c}_3^i \cdot \mathbf{c}_3^i - 1$ 。

这样目标函数转化为 $\min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) = f^T(\mathbf{x})f(\mathbf{x})$, 此函数为非线性最小二乘优化问题, 可利用高斯牛顿法进行迭代求解。高斯牛顿法需要首先求解 $f(\mathbf{x})$ 的 Jacobian 矩阵 $\mathbf{J}$, 然后通过多次迭代求解 $\mathbf{x}_k$ 的改变量 $\mathbf{d}_k$, 进而计算新的 $\mathbf{x}_k$ 值, 未知变形的当前估计为

$$\mathbf{d}_k = \underset{\mathbf{d}}{\operatorname{argmin}} \|\mathbf{f}_k + \mathbf{J}_k \mathbf{d}\|_2^2 \quad (5)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k$$

在每次迭代过程中, $\mathbf{J}_k \mathbf{d}_k + \mathbf{f}_k = \mathbf{0}$ 为超定方程组, 不能利用一般的求矩阵逆的方法求解, 需要将其通过以下方式转化为正规方程组的解:

$$\mathbf{J}_k^T \mathbf{J}_k \mathbf{d}_k = \mathbf{J}_k^T (-\mathbf{f}_k) \quad (6)$$

如果系数矩阵 $J_k^T J_k$ 可逆,则该方程组有唯一解.此时 x_k 的改变量 d_k 可由下式求出:

$$d_k = (J_k^T J_k)^{-1} J_k^T (-f_k)$$

此方程在稀疏矩阵 J 满秩,并且 $J_k^T J_k$ 可逆时有解.通过合理调整 $f(x)$ 的结构,可使 J 和 $J_k^T J_k$ 满足求解要求.同时注意到 $J_k^T J_k$ 为对称矩阵,进一步通过调节 w_{reg} 和 w_{rot} 的取值,使 $J_k^T J_k$ 为对称正定矩阵,利用 Cholesky 分解^[12,13] 求解可以提高计算速度,得到实时显示的变形效果.

重复上述迭代过程直到收敛,若收敛精度为 ϵ ,则利用目标函数的梯度: $\|\nabla F\|_\infty < \sqrt[3]{\epsilon}(1 + F_k)$;或目标函数的改变量: $|F_k - F_{k-1}| < \epsilon(1 + F_k)$;或者 d_k 的模: $\|d_k\|_\infty < \sqrt[2]{\epsilon}(1 + \|d_k\|_\infty)$,来作为迭代终止条件.

4 实验验证

根据本文提出的算法,进行了三维形状变形实验和效果比较.图 4 所示为三维狮子形状模型的交互变形操作实例.狮子局部变形时,其身体的其他部位相应随之变形.其头部抬起后,身体其余部分产生连贯的动作.该实例表明,经过优化迭代和计算,狮子形体上其余顶点随鼠标的移动而实时调整坐标位置,使整个变形过程自然流畅,并有效地保持了形体的细节信息.

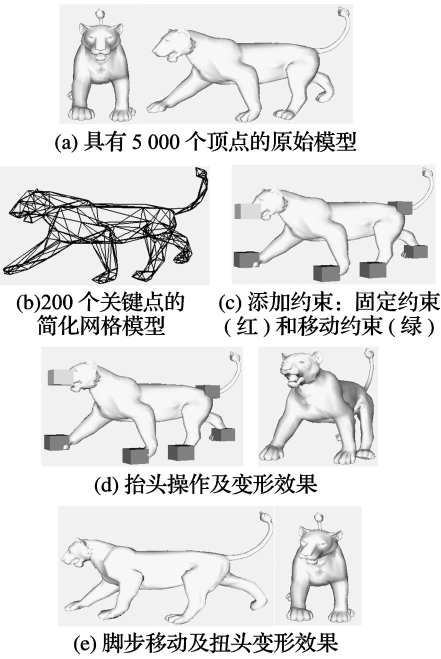


图 4 三维狮子模型交互变形

Fig. 4 3D lion shape interactive deformation

由于边折叠网格简化算法可有效地提取形体关键特征点作为其简化模型的节点,可以充分表征原始三维模型的形状特性.图 5 所示的是顶点数为 48 485 的马模型变形效果,其简化网格模型见图 2.图 5(b)是拉伸脖子变形效果,从图中可以看出脖子与身体结合处的变形自然流畅;图 5(c)是拉伸马腿的效果,马增高的同时,其局部细节信息得到了有效保护.与图 4 相比,此模型原始顶点数为狮子模型的 10 倍左右,但简化模型的关键点个数仍为 200,同样获得了满意的形状变形效果.

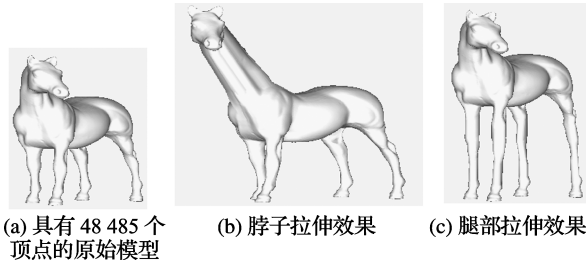


图 5 三维马模型交互变形

Fig. 5 3D horse shape interactive deformation

图 6 所示为骆驼模型的变形效果.原始模型的顶点数为 21 887,图 6(b)是其简化网格模型,关键点个数为 200,图 6(c)和(d)为其变形操作后的效果.

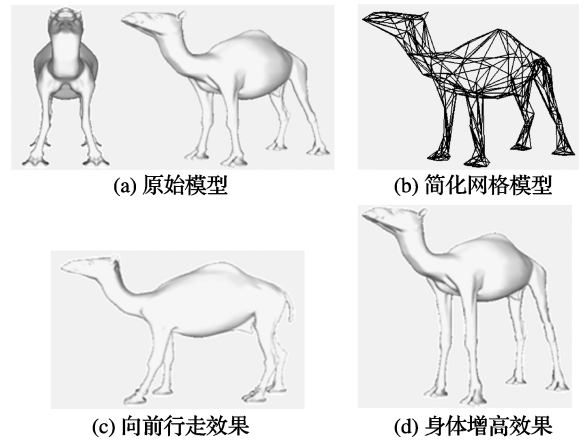


图 6 三维骆驼模型交互变形

Fig. 6 3D camel shape interactive deformation

5 结 语

本文基于三角形网格的边折叠简化,对三维形状变形操作算法进行了研究.三角形边折叠简化在保持三维形状轮廓信息的同时,可以有效提取形体的关键特征点,减少了模型的表征节点数.

基于简化模型,通过求解非线性目标函数,实现形状模型的变形映射,既保证了变形效果,又减少了非线性目标函数中的未知变量个数.算法求解中,通过使大型稀疏矩阵正定,利用 Cholesky 分解加速优化迭代过程.实例表明,本文算法可用于以顶点定义、具有大量细节信息的三维自然形体的变形操作,在变形过程中可有效保持形状细节信息,获得实时的显示效果.

参考文献:

- [1] BARR A H. Global and local deformations of solid primitives [J]. **Computer Graphics**, 1984, **18**(3):21-30
- [2] SEDERBERG T W, PARRY S R. Free-form deformation of solid geometric models [J]. **Computer Graphics**, 1986, **20**(4):151-160
- [3] HSU W M, HUGHES J F, KAUFMAN H. Direct manipulation of free-form deformations [J]. **Computer Graphics (ACM)**, 1992, **26**(2):177-184
- [4] BOTSCH M, KOBELT L. An intuitive framework for real-time freeform modeling [J]. **ACM Transactions on Graphics**, 2004, **23**(4):630-634
- [5] SORKINE O. Laplacian mesh processing [D]. Tel Aviv: School of Computer Science, Tel Aviv University, 2006
- [6] COHEN-OR D. Space deformations, surface

- deformations and the opportunities in-between [J]. **Journal of Computer Science and Technology**, 2009, **24**(1):2-5
- [7] BOTSCH M, SUMNER R W, PAULY M, *et al.* Deformation transfer for detail preserving surface editing [C] // **Proceedings of Vision, Modeling & Visualization**. Aachen: Bielefeld Graphics & Geometry Group, 2006
 - [8] ZHOU Kun, HUANG Xin, XU Wei-wei, *et al.* Direct manipulation of subdivision surfaces on GPUs [J]. **ACM Transactions on Graphics**, 2007, **26**(3):1276491
 - [9] SUMNER R W, SCHMID J, PAULY M. Embedded deformation for shape manipulation [J]. **ACM Transactions on Graphics**, 2007, **26**(3):1276478
 - [10] 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
 - [11] 潘志庚,庞明勇. 几何网格简化研究及进展[J]. 江苏大学学报, 2005, **26**(1):67-71
 - [12] DAVIS T A, HAGER W W. Dynamic supernodes in sparse Cholesky update/downdate and triangular solves [J]. **ACM Transactions on Mathematical Software**, 2009, **35**(4):1-23
 - [13] TOLEDO S. TAUCS: A library of sparse linear solvers [EB/OL]. (2009-09-01) <http://www.tau.ac.il/~stoledo/taucs>.

3D shape deformation based on edge collapse mesh simplification

HUA Shun-gang*, ZHONG Qing, LI Shao-shuai

(Key Laboratory for Precision & Non-traditional Machining Technology of Ministry of Education,
Dalian University of Technology, Dalian 116024, China)

Abstract: Based on edge collapse mesh simplification, an interactive 3D shape deformation algorithm is proposed. By collapsing edges of triangle mesh to extract key points from original 3D model, the reduced 3D model is generated, and then fixed or movement constraints are forced to the reduced model interactively. By constructing and solving nonlinear optimization equation, optimal solutions of affine transformations are found for all key points. On the basis of affine transformations of key points, deformation mappings for all nodes in original model are calculated. Examples show that edge collapse mesh simplification can be effectively used to reduce the number of polygons and computation, and combining it with nonlinear optimization solution, a real-time display and detail-preserved 3D shape deformation can be realized interactively.

Key words: edge collapse mesh simplification; 3D shape deformation; affine transformation; nonlinear optimization