

中图法分类号: TP301.6 文献标志码: A 文章编号: 1006-8961(2011)10-1918-08

论文索引信息: 王勋, 王启超. 四面体胞的尽可能刚性网格变形算法 [J]. 中国图象图形学报, 2011, 16(10): 1918-1925

# 四面体胞的尽可能刚性网格变形算法

王勋, 王启超

(浙江工商大学计算机与信息工程学院, 杭州 310018)

**摘要:** 提出一种基于四面体胞的尽可能刚性三角形网格变形算法。用户通过操作网格上的若干顶点以得到所需的模型变形结果。首先, 算法对网格模型内部进行稀疏四面体化, 以产生一个贴合模型表面的四面体胞集。在模型变形过程中, 算法通过最小化相应的变形能量函数, 以保持网格模型表面局部区域的刚性以及每个四面体胞的刚性, 从而有效避免模型表面及其内部的扭曲。同时, 针对大尺度编辑可能造成的模型局部塌陷, 提出一种简单的四面体胞自适应剖分方法, 根据模型局部体积的剧烈变化, 自动剖分对应的四面体胞以增加模型内部的局部变形自由度, 进而消除不正确的变形效果。此外, 自适应的四面体胞剖分允许算法在初始时只需对网格模型进行稀疏的四面体化, 而在变形过程中根据需要进一步提高四面体胞的局部稠密度, 因而保证了算法的鲁棒性及其效率。实验结果表明, 该变形算法可以有效保持模型的表面细节以及模型的内部体积, 并能够有效避免模型形状在大尺度变形时的局部退化。

**关键词:** 模型变形; 刚性变形; 体积约束; 能量函数; 自适应

## As-rigid-as-possible mesh deformation based on tetrahedral cells

Wang Xun, Wang Qichao

(College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018 China)

**Abstract:** In this paper, we introduce a novel approach to as-rigid-as-possible mesh deformation based on tetrahedral cells. The most distinctive feature of this approach is that users can modify the original mesh by selecting a couple of vertices on the mesh. First, sets of tetrahedral cells are produced, which fit well to the mesh surface, through sparse voxelization. Preserving the rigidity of local transformations of the mesh surface and these tetrahedra during deformation is achieved by minimizing the corresponding energy formulation that prevents unnatural artifacts both on the surface and in the interior of the model. Meanwhile, in order to avoid the possible collapse emerged under large-scale deformation, we present a simple adaptive tetrahedron decomposition method, which brings more local deformation freedom to the interior of the mesh to eliminate the implausible deformation effects based on these dramatic volume changes in the local area. In addition, a few tetrahedra are generated by this method, and then more tetrahedral cells would be densely involved into the deformation process if necessary, there by resulting in highly efficient and robust results. In our experiments, we show that the volume and the surface details of the whole mesh can be approximately preserved and the local degeneration inside the mesh under the large-scale deformation could be efficiently avoided.

**Keywords:** shape deformation; rigid deformation; volume constraint; energy minimization; adaptive

收稿日期: 2010-12-22; 修回日期: 2011-05-06

基金项目: 国家自然科学基金项目(60873218, 61003189); 国家重点基础研究计划(973)基金项目(2009CB320801); 浙江省自然科学基金重点项目(Z1080232, Z1101340); 浙江省教育厅重点项目(Z201018041)。

第一作者简介: 王勋(1967—), 男, 教授。2006年于浙江大学CAD&CG国家重点实验室获计算机科学技术专业博士学位, 主要研究领域为虚拟现实, 地理信息系统和智能信息处理。E-mail: wx@mail.zjgsu.edu.cn。

## 0 引言

随着 3 维数据采集技术的不断发展,3 维网格模型在模拟仿真、逆向工程、工业设计、影视特效和互动娱乐等领域得到了广泛应用。进而,基于 3 维网格模型的数字几何处理技术成为计算机图形学领域的一个研究热点。作为数字几何处理的核心技术之一,网格模型的变形近年来受到了国内外研究者的极大关注。最近,众多研究者提出基于能量优化的几何模型变形方法<sup>[1-13]</sup>。其中,基于微分域的网格曲面变形方法通过优化相应的变形能量函数来保持曲面的局部微分量,从而保证模型变形后不丢失细节,最终获得视觉上自然的变形结果。例如,Sorkine 等人使用顶点拉普拉斯坐标(Laplacian coordinate)作为局部微分量<sup>[1]</sup>,俞益洲等人使用曲面上的梯度坐标作为局部微分量<sup>[2]</sup>,Lipman 等人使用局部标架作为局部微分量<sup>[3]</sup>,Zayer 等人<sup>[4]</sup>通过调和函数插值把约束点处的局部变换传播到网格模型的其余部分。此外,由于待优化的变形能量函数是非线性能量函数,一些方法通过线性化把非线性优化问题转为线性优化问题<sup>[1]</sup>,从而提高优化求解速度。但是,线性化方法常会产生带有收缩和扭曲的子优化变形结果。为了避免线性子优化变形结果,可以采用非线性优化方法。黄劲等人<sup>[5]</sup>提出子空间变形技术,他们把高分辨率的网格模型嵌入在一个低分辨率的多边形网格中,然后通过均值坐标<sup>[6]</sup>把模型上的顶点表示为多边形网格顶点的线性组合,并代入到变形能量函数中,这样大大减少了变形能量函数中的变量个数并且线性地光滑了原始变量之间的非线性关系。与此同时,Au 等人<sup>[7]</sup>和 Sumner 等人<sup>[8]</sup>分别提出各自的简化变形模型,通过有选择性地构建若干个有代表性的局部变换来表示曲面的局部变形,以加快非线性优化方法的求解速度并提高求解的鲁棒性。

但是,以上这些方法主要考虑如何保持模型表面的细节,而对于模型的其他几何属性,如体积等。由于它们的高度的非线性性,则缺乏有效的解决策略,从而导致模型在大尺度变形时常会产生不自然的变形结果。其中,Sorkine 等人<sup>[9]</sup>提出一种表面尽可能刚性的 3 维模型变形方法。他们为模型表面上的每个点的一环邻域定义了一个局部刚性区域,并在变形过程中通过最小化相应的变形能量函数尽量

避免这些局部刚性区域发生扭曲,从而达到保持模型表面局部细节的目的。但是,由于他们的方法只考虑了模型表面的信息,因而模型在大尺度变形情况下会产生严重的细节扭曲和明显的体积变化。周昆等人<sup>[11]</sup>为 3 维模型建立相应的体图,并把 Laplacian 微分坐标从曲面推广到体。进而,通过保持模型体图的 Laplacian 坐标,以避免模型在大尺度变形时发生显著的体积变化。但是,为 3 维网格模型构建相应的体图较为复杂;并且,体图中的几何元素可能是四面体、六面体,甚至更多面体,不便于显式计算网格模型的当前体积,因而难以精确地控制网格模型变形时的体积变化。Botsch 等人<sup>[12]</sup>将网格模型嵌入到一个空间六面体网格中,并通过保持空间六面体的刚性来避免内嵌模型发生严重的体积扭曲。同时,他们根据空间六面体网格的局部变形扭曲,自适应地对其进行局部剖分,以增加它的局部变形自由度,从而有效避免了内嵌模型在变形时发生显著扭曲。但是,在他们方法中,模型的变形结果受制于空间六面体网格的初始朝向,由于空间六面体网格并不与模型表面贴合,当两者朝向不一致时(甚至是轻微的不一致),变形模型的表面细节容易产生无向扭曲现象。Rivers 等人<sup>[13]</sup>把物体嵌入在规则网格中,通过基于几何匹配的方法<sup>[14]</sup>模拟规则网格的变形得到嵌入其中的物体变形。他们通过邻近网格单元来构建相互覆盖的粒子几何,并且粒子几何的数目也与网格单元数量差不多,从而得到光滑的网格变形效果。虽然基于几何匹配的模拟方法快速鲁棒,但对于大尺度的变形,它们的收敛速度仍无法满足实时交互需求。

提出一种基于四面体胞的尽可能刚性格网变形算法。首先,算法对网格模型内部进行稀疏四面体化,以产生一个贴合模型表面的四面体胞集。在模型变形过程中,算法通过最小化相应的变形能量函数,以保持网格模型表面局部区域的刚性以及每个四面体胞的刚性,从而有效地保持了模型的表面细节及其内部体积。同时,针对大尺度编辑可能造成的模型局部塌陷,提出一种简单的四面体胞自适应剖分方法,根据模型局部体积的剧烈变化,自动剖分对应的四面体胞以增加模型内部的局部变形自由度,进而消除不正确的变形效果。此外,自适应的四面体胞剖分允许算法在初始时只需对网格模型进行稀疏的四面体化,因此,四面体化时新增的模型内部顶点个数远小于初始的三角形网格顶点个数。同

时,在变形过程中,算法仅有针对性地对四面体胞进行局部剖分。最终,算法的求解复杂性与 Sorkine 等人的基于模型表面的尽可能刚性变形算法<sup>[9]</sup>基本一致。

### 1 基于四面体胞的尽可能刚性变形

为了实现基于四面体胞的网格模型变形,算法首先把模型内部离散成一个四面体胞集,如图 1 所示。为了演示,该模型内部加入了大量的内部点。在实际变形例子中,仅对模型进行稀疏四面体化,只在模型内部加入少量的内部点。在当前算法中,使用 Tetgen 自动对输入模型进行稀疏四面体化<sup>[10]</sup>,以生成所需的四面体胞集。在详细阐述变形算法前,先给出若干定义。令四面体胞集为一二元组  $M = (V, E)$ 。其中,  $V = \{v_i\}$ 、 $E = \{e_{ij} = \langle v_i, v_j \rangle\}$  分别为四面体胞集中顶点和边的集合。显然,输入网格模型的顶点即为顶点集合  $V$  中在模型表面的顶点,记  $V'$ 。此外,令  $p_i, q_i$  分别为四面体胞集中顶点  $v_i$  在初始以及变形时的位置。

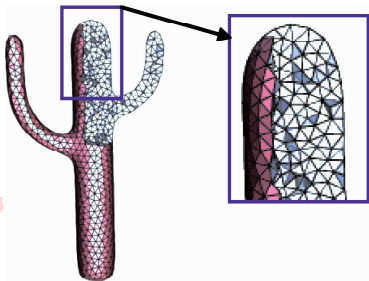
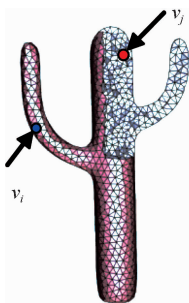


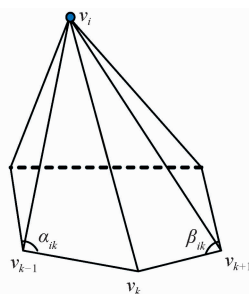
图 1 网格模型的四面体剖分

Fig. 1 Tetrahedral voxelization of the mesh

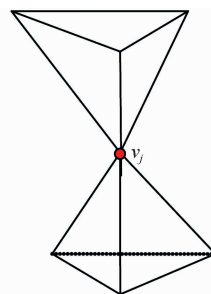
介绍的基于四面体胞的三角网格模型变形方法,不仅考虑模型表面的几何细节,并且把模型内部离散成一个四面体胞集,以在其内部增加相应的扭



(a) 网格模型



(b) 网格表面顶点  $v_i$  的一环邻域  $NS(i)$



(c) 四面体胞集中内部顶点  $v_j$  的一环邻域  $NV(j)$

图 2 模型顶点的一环邻域定义

Fig. 2 Definition of one-ring neighborhood of mesh vertex

曲抑制约束,从而达到同时保持模型表面几何特征以及体积的变形效果,最终生成视觉上自然的变形结果。在变形过程中,算法最小化相应的变形能量函数,以保证产生所需的变形结果。该变形能量函数包括如下 3 部分:

1) 网格模型表面的保细节能量  $E_{surf}$ ,通过该能量在变形过程中保持模型表面的几何细节。该能量类似于 Sorkine 等人<sup>[9]</sup>介绍的基于网格表面的尽可能刚性能量,其具体定义为

$$E_{surf(i)} = \sum_{k \in NS(i \in V')} \omega_{ik} \| (q_i - q_k) - R_i(p_i - p_k) \|^2 \quad (1)$$

其中,对于网格表面上的每个顶点  $v_i \in V'$ ,  $NS(i)$  表示该顶点在网格表面上的一环邻域(如图 2(b)所示);  $R_i$  表示该一环邻域的旋转变换,通过约束网格表面的局部邻域进行刚性变换,有效保持网格表面的几何局部细节<sup>[7]</sup>;  $w_{ik}$  代表顶点  $v_i$  与  $v_k$  之间的权重且满足  $\sum_{k \in NS(i)} w_{ik} = 1$ 。这里,定义  $w_{ik}$  为<sup>[9]</sup>:  $w_{ik} = A_i(\cot \alpha_{ik} + \cot \beta_{ik})/2$ 。其中,  $\alpha_{ik}$  和  $\beta_{ik}$  的定义如图 2(b)所示;  $A_i$  则代表顶点  $v_i$  的近似 Voronoi 区域的面积。显然,  $w_{ik}$  由网格模型的局部拓扑以及几何(角度与面积)确定,因而能够更加精确地描述模型的局部几何特征。

2) 网格模型内部四面体胞的抗扭曲能量  $E_{tet}$ 。在变形过程中,如果只保持模型的表面细节,模型在大尺度变形下常会产生严重的失真,如图 3(b)所示。抗扭曲能量保证网格模型内部在变形过程不发生扭曲,从而有效保持模型的体积,避免了模型在大尺度变形时产生的严重细节失真和明显体积变化,如图 3(c)所示。该能量具体定义为

$$E_{tet(j)} = \sum_{k \in NV(j \in (V-V'))} \omega_{jk} \| (q_j - q_k) - R_j(p_j - p_k) \|^2 \quad (2)$$

式中,对于四面体胞集在模型内部的任一顶点  $v_j \in (V - V')$ ,定义与该顶点相邻的所有顶点为其一环邻域,用  $NV(j)$  表示(如图 2(b)所示); $R_j$  表示该环邻域的旋转变换; $w_{jk}$  的定义与公式(1)中的  $w_{ik}$  类似。

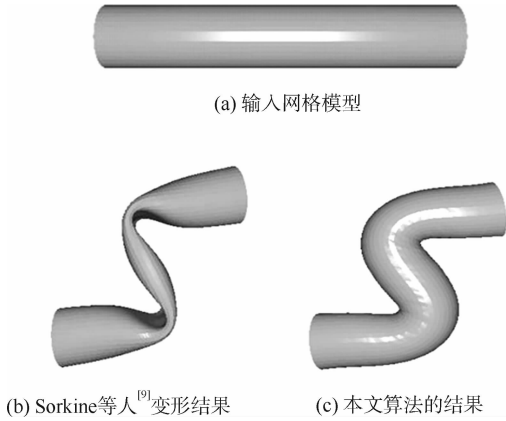


图 3 Cylinder 的变形结果

Fig. 3 Cylinder blending deformation

3) 用户控制顶点约束能量  $E_{ctrl}$ 。该能量保证产生的网格模型变形结果满足用户指定的约束,其具体定义为

$$E_{ctrl} = \sum_k \|P_k - c_k\|^2 \quad (3)$$

其中,对于用户在网格模型表面上选择的每个顶点  $P_k, c_k$  为用户编辑后该顶点的新位置。

最终,算法最小化如下的变形能量函数,以产生满足用户约束的自然的模型变形结果

$$E_{tot} = \lambda_1 \sum_{i \in V'} E_{surf(i)} + \lambda_2 \sum_{j \in (V-V')} E_{tet(j)} + \lambda_3 E_{ctrl} \quad (4)$$

式中,  $E_{surf(i)}, E_{tet(j)}, E_{ctrl}$  分别为上述的 3 个能量子项,  $\lambda_1, \lambda_2$  和  $\lambda_3$  分别为它们的权因子。

在本文的所有实验例子中,  $\lambda_1, \lambda_2$  和  $\lambda_3$  取为:  $\lambda_1 = \lambda_2 = 0.7, \lambda_3 = 1$ , 均取得了令人满意的变形结果。对于式(4)中能量函数中包含的式(1)(2)中的能量子项,由于它们当中的旋转变换  $R_i$  与  $R_j$  均与式(4)中的未知变量:  $q_i$  (四面体胞集中顶点的变形位置) 相关,并且是一种非线性关系,因此,最小化式(4)中的能量函数是一个非线性优化问题。在第 2 节中,介绍一种迭代的方式来求解该优化问题。

## 2 变形能量函数的优化

采用一种交替最小化的方式迭代地优化式(4)

中的能量函数,以计算四面体胞集的顶点变形位置  $\{q_i\}$ , 也即获得了网格模型的变形位置。该交替最小化方法与 Sorkine<sup>[9]</sup> 以及杨文武等人<sup>[15-16]</sup> 采用的方法类似,其具体步骤如下:

首先,给定  $\{q_i\}$  的一个初始估计值。在实现过程中,本文算法采用上一次操作后四面体胞集的顶点位置作为初始估计值(实验发现,该初始估计值非常适合 3 维网格模型的交互式编辑)。然后,通过如下迭代过程进行求解:

- 1) 给定初始值:  $\{q_i\}^{(0)}$ ;
- 2) 在第  $l$  次迭代时,基于上次变形结果  $\{q_i\}^{(l-1)}$ ,最小化式(4)中的能量函数,计算得到每个一环邻域在当前的最优旋转变换  $\{R_i\}^{(l-1)}$  和  $\{R_j\}^{(l-1)}$ ;
- 3) 把计算得到的最优旋转变换  $\{R_i\}^{(l-1)}, \{R_j\}^{(l-1)}$  和用户编辑的网格约束顶点新位置代入式(4)。此时,最小化式(4)中的能量函数即转化为一个线性最小二乘问题,通过求解一个线性系统,即得到当前四面体胞集的顶点位置:  $\{q_i\}^{(l)}$ ;
- 4) 当能量函数式(4)局部收敛时,迭代结束,否则转到步骤 2),继续迭代。

在上述迭代步骤 2) 中,每个一环邻域的当前最优旋转变换  $\{R_i\}^{(l-1)}$  和  $\{R_j\}^{(l-1)}$  可以通过最小化式(1)(2)定义的能量子项  $E_{surf(i)}$  和  $E_{tet(j)}$  得到。以求解  $E_{tet(j)}$  中的第  $l$  步当前最优旋转变换为例,假设  $\{q_i\}^{(l-1)}$  已知,那么求解相应的最优旋转变换  $\{R_i\}^{(l)}$  可以使用如下变换矩阵来逼近地表示  $\{R_i\}^{(l)}$  在式(1)下的最小二乘解<sup>[15]</sup> 为

$$A_l = \sum_{k \in NS(i)} (q_i - q_k)(p_i - p_k)^T \quad (5)$$

对  $A_l$  进行 QS 分解,得到  $A_l = RS$ 。其中,  $R$  为一正交单位矩阵,  $S$  为一对角矩阵<sup>[11]</sup>。因为对角矩阵  $S$  仅包含伸缩变换,因此  $R$  即为所求的最优旋转变换。

在步骤 3) 中,为描述方便,将该线性系统表示成紧凑的矩阵形式

$$MQ = b \quad (6)$$

式中,  $M$  为该线性系统的系统矩阵,  $Q$  为待求未知量组成的列向量,  $b$  为一已知列向量。注意到,在该线性系统中,元素是稀疏的,并且系统矩阵  $M$  仅与四面体胞集中边以及网格模型中边的连接关系相关,不会随着模型的位置改变而发生变化,因此,采用具有预分解功能的直接求解方法求解该稀疏线性

系统<sup>[17]</sup>。其中,系统矩阵  $M$  只需要分解一次,之后每次求解,只需进行若干次回代操作 (back substitution)。通过这种方式,线性系统式 (6) 可以被高效求解。

### 3 自适应的四面体剖分

为了保证算法效率,算法仅对输入的网格模型进行稀疏的四面体化,从而避免了在模型内部加入

过多的内部点而增加算法的求解复杂度。但是,当模型内部中的四面体胞数量较少时,又可能造成模型内部的变形自由度不足而导致变形扭曲。如图 4 (a) 方框中所示, Cactus 模型在大尺度编辑后出现了局部塌陷的现象。为了解决该问题,提出一种简单的四面体胞自适应剖分方法,它根据模型局部体积的剧烈变化,自动剖分对应的四面体胞以增加模型内部的局部变形自由度,进而消除不正确的变形效果(如图 4(b) 所示)。

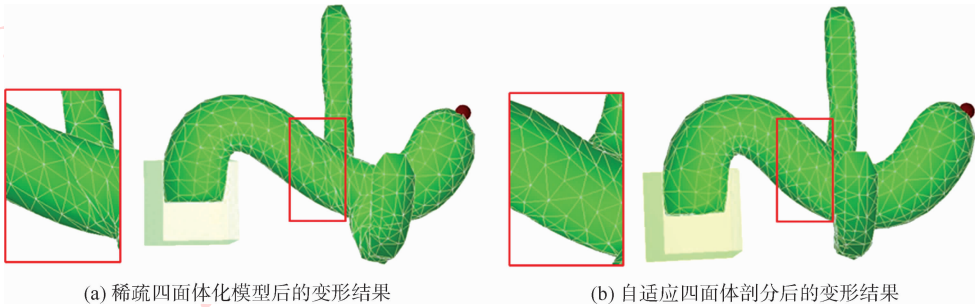


图 4 Cactus 交互式变形结果的比较

Fig. 4 Comparison of Cactus blending deformation

在实验中,  $r_d = 2/3$ , 它表示在变形过程中,当四面体胞体积减少或增大了  $1/3$  倍时,则剖分该四面体胞,以增加该处的变形自由度。为了剖分四面体胞,算法在四面体胞的中心位置增加一个新的顶点  $v_{new}$ , 并结合原有的 4 个顶点将该四面体胞一分为四。新顶点在初始以及当前四面体胞集中的位置为

$$p_{new} = \frac{3p_{k4} + 2(p_{k1} + p_{k2} + p_{k3})}{9}$$

$$q_{new} = \frac{3q_{k4} + 2(q_{k1} + q_{k2} + q_{k3})}{9}$$

在每次完成对四面体胞集的剖分后,需要升级第 3 节介绍的网格模型内部四面体胞的抗扭曲能量  $E_{tet}$ , 并对变形能量函数进行重新求解。由于算法仅自动在四面体胞集中体积剧烈变化的区域进行剖分,因此四面体胞集中的体胞数量不会显著增加。此外,由于新增体胞的局部性,升级后的变形能量函数在重新求解时通常只需  $1 \sim 2$  次迭代。

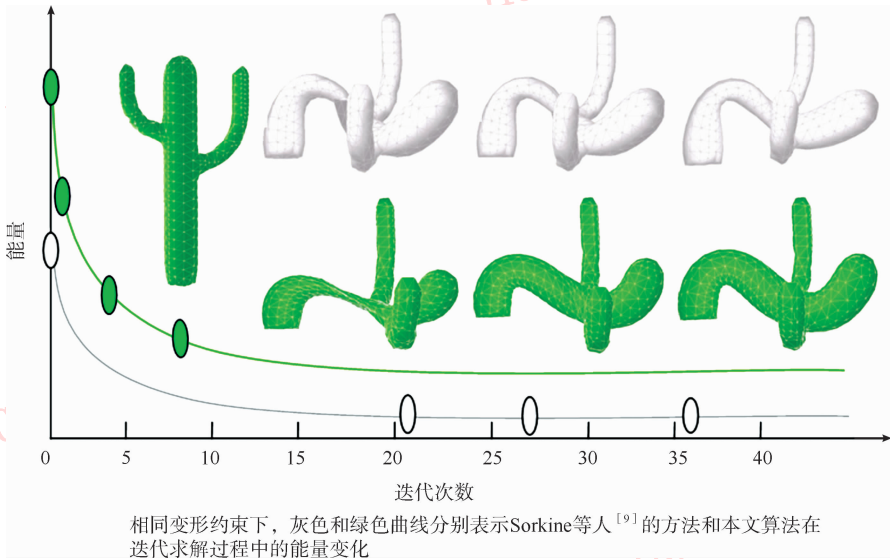
### 4 实验结果与讨论

在具有 Intel 双核 2.4GHz CPU、2GB 内存、

Windows XP 运行环境的 PC 机上,以单线程方式实现了介绍的基于四面体胞的尽可能刚性网格变形算法。为了编辑网格模型,用户在网格表面上选取若干顶点并对它们进行平移、旋转或缩放,然后算法通过最小化变形能量函数式 (4) 自动计算得到模型的变形结果。

在图 5 中,给出了在相同变形约束下,本文算法与 Sorkine 等人<sup>[9]</sup> 基于模型曲面的变形方法在迭代求解过程中的能量收敛示意图。由该图可以看出,本文算法具有更好的收敛性,通常只需要  $7 \sim 13$  次求解迭代。此外,自适应的四面体胞剖分允许算法在初始时只需对网格模型进行稀疏的四面体化,因此,四面体化时新增的模型内部顶点个数远小于初始的三角形网格顶点个数。

同时,在变形过程中,算法仅有针对性地对四面体胞进行局部剖分,不会显著增加模型内部的体胞顶点数(如表 1 所示)。最终,算法的求解时间与 Sorkine 等人的基于模型表面的尽可能刚性变形算法<sup>[9]</sup> 基本一致(如表 1 所示)。实验例子中,对于中等规模大小的网格模型(顶点数  $\leq 4\ 000$ ),算法可以达到  $4 \sim 15$  帧/s 的交互率,基本能够满足用户的实时交互需要。



相同变形约束下,灰色和绿色曲线分别表示Sorkine等人<sup>[9]</sup>的方法和本文算法在迭代求解过程中的能量变化

图 5 Cactus 模型变形时的能量曲线

Fig. 5 The convergence curve of the deformation energy

表 1 本文所使用模型、体积及变形时间统计信息

Tab. 1 The statistics of demos shown in the paper

| 输入模型      | 原始模型 |           |           | 未剖分四面体胞集 |          | 自适应剖分四面体胞集 |           | 预计算时间/s                      |        | 平均回代时间/s                     |       |
|-----------|------|-----------|-----------|----------|----------|------------|-----------|------------------------------|--------|------------------------------|-------|
|           | 顶点数  | 体积        | 体网格剖分时间/s | 顶点数      | 体积       | 顶点数        | 体积        | Sorkine 等人 <sup>[9]</sup> 算法 | 本文算法   | Sorkine 等人 <sup>[9]</sup> 算法 | 本文算法  |
| Cylinder  | 236  | 35.300    | 0.188     | 236      | 29.564   | 260        | 31.527    | 0.044                        | 0.374  | 0.009                        | 0.012 |
| Cactus    | 620  | 0.01186   | 0.14      | 625      | 0.01062  | 670        | 0.01151   | 0.182                        | 2.219  | 0.024                        | 0.032 |
| Dinosaur  | 5621 | 24039.88  | 1.172     | 6660     | 21984.56 | 6863       | 22368.20  | 2.327                        | 12.531 | 0.171                        | 0.251 |
| Armadillo | 4325 | 235810.62 | 0.672     | 4629     | 205722.4 | 4835       | 210475.12 | 1.901                        | 10.562 | 0.152                        | 0.203 |
|           | 4325 | 235810.62 | 0.672     | 4629     | 214732.4 | 4967       | 219841.23 | 1.901                        | 10.562 | 0.152                        | 0.203 |

通过保持模型内部四面体胞的刚性,本文算法能够有效避免大尺度变形引起的扭曲效果。如图 6 所示,对于圆柱模型的一个较大角度的弯曲变形, Sorkine 等人的方法<sup>[9]</sup>只考虑圆柱的表面细节,产生的变形结果整体产生了明显的塌陷(图 6(b)),而本文算法在保持网格表面细节的同时,很好地避免了模型塌陷(图 6(c))。

图 7 给出了另一个例子,表明本文算法产生的变形结果优于仅考虑模型表面的变形方法<sup>[9]</sup>。特别地,在本文算法中,当初初始化的四面体胞较稀疏,具备的变形自由度不足以完全避免变形塌陷时(图 4(a)),介绍的四面体剖分方法会自动侦测出模型局部体积的剧烈变化,并自动剖分对应的四面体胞

以增加模型内部的局部变形自由度,进而消除不正确的变形效果(图 4(b))。

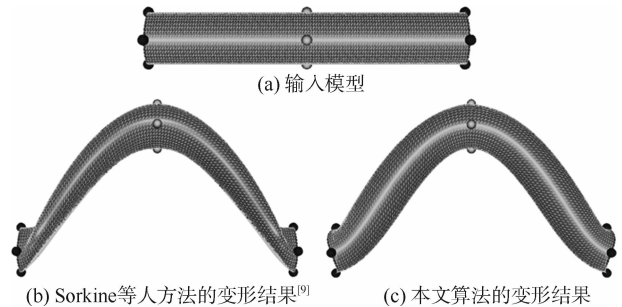


图 6 Cylinder 扭曲变形

Fig. 6 Cylinder bending deformation

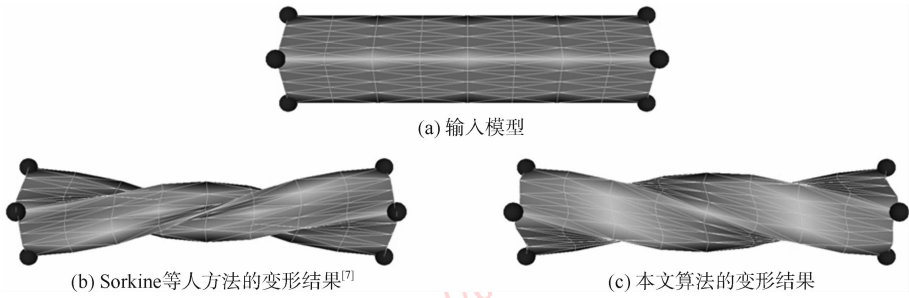


图 7 Bar 拉伸变形

Fig. 7 Bar twisting deformation

在图 8 和图 9 中,给出了本文算法为另外两个复杂模型产生的变形结果。从结果可以看出,尽管模型发生了较大尺度的变形,本文算法也能保

持它们之上的丰富表面细节特征,并且能够有效保持模型的几何体积,进而得到视觉上自然的变形结果。

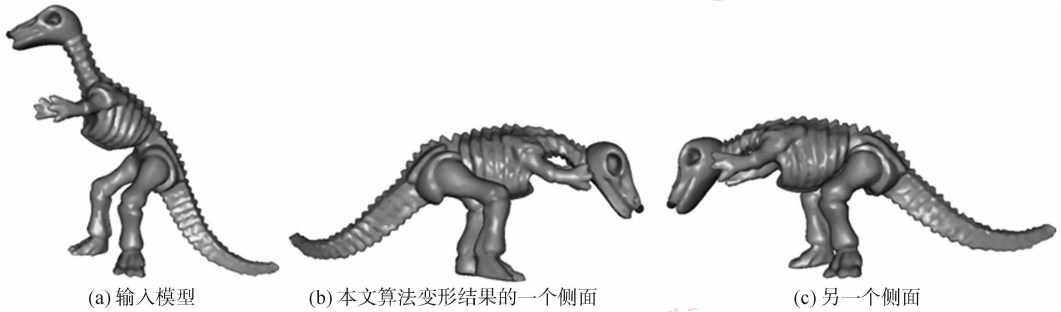


图 8 Dinosaur 变形

Fig. 8 Dinosaur deformation

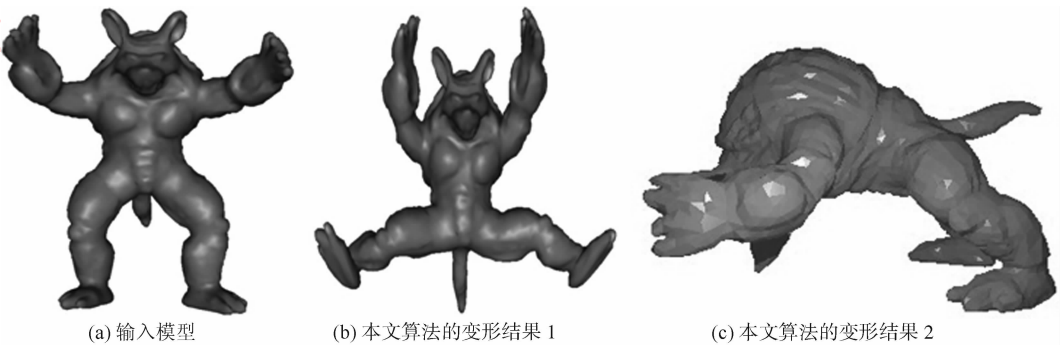


图 9 Armadillo 的变形

Fig. 9 Armadillo deformation

## 5 结 论

提出一种基于四面体胞的三角网格模型变形算法。首先,算法对网格模型内部进行稀疏四面体化,在模型变形过程中,通过最小化相应的变形能量函数,以保持网格模型表面局部区域的刚性,以及每个

四面体胞的刚性。最终,算法可以有效避免模型在大尺度变形时发生几何细节扭曲,以及体积变化等退化现象。同时,针对模型内部中的四面体胞数量较少时可能造成的变形扭曲,提出一种简单的四面体胞自适应剖分方法,根据模型局部体积的剧烈变化,自动剖分对应的四面体胞以增加模型内部的局部变形自由度,进而消除不正确的变形效果。实验

结果表明,算法直观、鲁棒,能够生成视觉上自然的变形结果。缺点是本文算法求解时间与模型表面顶点和内部顶点相关,对于具有一定几何细节的网格数据,很难做到实时变形。

希望把变形过程中的求解过程从 CPU 上迁移到 GPU 上,以进一步提高算法效率。此外,拟研究相应的多分辨率方法以使新方法适用于顶点数巨大的复杂模型。最后,拟把基于四面体胞的变形方法,进一步推广到其他应用,如变形迁移、形状插值等。

## 参考文献 (References)

- [ 1 ] Sorkine O, Cohen-Or D, Lipman Y, et al. Laplacian surface editing [ C ] // Proceedings of Eurographics Symposium on Geometry Processing. Nice, France: Eurographics Association, 2004: 175-184.
- [ 2 ] Yu Y Z, Zhou K, Xu D, et al. Mesh editing with poisson-based gradient field manipulation [ J ]. ACM Transactions on Graphics, 2004, 23 ( 3 ) : 644-651.
- [ 3 ] Lipman Y, Sorkine Y, Cohen-Or D, et al. Linear rotation-invariant coordinates for meshes [ J ]. ACM Transactions on Graphics, 2005, 24 ( 3 ) : 479-487.
- [ 4 ] Zayer R, Rössl C, Kami Z, et al. Harmonic guidance for surface deformation [ J ]. Computer Graphics Forum, 2005, 24 ( 3 ) : 601-609.
- [ 5 ] Huang J, Shi X H, Liu X G, et al. Subspace gradient domain mesh deformation [ J ]. ACM Transactions on Graphics, 2006, 25 ( 3 ) : 1126-1134.
- [ 6 ] Ju T, Schaefer S, Warren J. Mean value coordinates for closed triangular meshes [ J ]. ACM Transactions on Graphics, 2005, 24 ( 3 ) : 561-566.
- [ 7 ] Au O K C, Fu H, Tai C L, et al. Handle-aware isolines for scalable shape editing [ J ]. ACM Transactions on Graphics, 2007, 26 ( 3 ) : 83-88.
- [ 8 ] Sumner R W, Schmid J, Pauly M. Embedded deformation for shape manipulation [ J ]. ACM Transactions on Graphics, 2007, 26 ( 3 ) : 80-87.
- [ 9 ] Sorkine O, Alexa M. As-rigid-as-possible surface modeling [ C ] // Proceedings of the fifth Eurographics Symposium on Geometry Processing. Barcelona, Spain: Eurographics Association, 2007: 109-116.
- [ 10 ] Si H, Tetgen. A Quality Tetrahedral Mesh Generator and 3-Dimensional Delaunay Triangulator [ CP/OL ]. ( 2009-01-24 ) [ 2010-06-07 ]. <http://tetgen.berlios.de/>.
- [ 11 ] Zhou K, Huang J, Snyder J, et al. Large mesh deformation using the volumetric graph Laplacian [ J ]. ACM Transactions on Graphics, 2005, 24 ( 3 ) : 496-503.
- [ 12 ] Botsch M, Pauly M, Wicke M, et al. Adaptive space deformations based on rigid cells [ J ]. Computer Graphics Forum, 2007, 26 ( 3 ) : 339-347.
- [ 13 ] Rivers A R, James D L. Fastlsm: fast lattice shape matching for robust real-time deformation [ J ]. ACM Transactions on Graphics, 2007, 26 ( 3 ) : 82.
- [ 14 ] Müller Matthias, Heidelberger Bruno, Teschner Matthias, et al. Meshless deformations based on shape matching [ J ]. ACM Transactions on Graphics, 2005, 24 ( 3 ) : 471-478.
- [ 15 ] Yang W W, Feng J Q, Jin X G. Shape deformation with tunable stiffness [ J ]. The Visual Computer, 2008, 24 ( 7 ) : 495-503.
- [ 16 ] Yang Wenwu. Computer Assisted Cartoon Character Pose Editing and Interpolation [ D ]. Hangzhou: Zhejiang University, 2009. [ 杨文武. 计算机辅助卡通角色动作姿势编辑与插值 [ D ]. 杭州:浙江大学, 2009. ]
- [ 17 ] Davis T A. UMFPAK: Unsymmetric Multifrontal Sparse LU Factorization Package [ CP/OL ]. ( 2004-06-16 ) [ 2010-06-07 ]. <http://www.cise.ufl.edu/research/sparse/umfpack/>.