

三维网格曲面的可展性优化算法

徐鹏飞 刘利刚

(浙江大学数学系 杭州 310027)

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)

摘 要 可展曲面是每点高斯曲率均为 0 的曲面, 具有许多良好的性质, 因此在工业中具有很多应用. 将一般的曲面用可展曲面来逼近表示具有现实意义. 以此为目的, 文中设计了一个有效的算法来处理一般的曲面, 使得处理后的曲面尽可能满足可展的性质, 同时与初始的曲面尽量地接近. 首先利用最小范数方法来对网格曲面进行处理, 得到初始预测的网格曲面. 初始预测曲面具有良好的可展性, 但是不能较好地保持网格的局部结构. 然后利用尽可能刚性 (As-rigid-as-possible) 的方法, 在初始预测曲面的基础上进行修正得到新的网格曲面. 为了保持局部结构, 作者的方法可以是基于顶点邻域的, 也可以是基于三角形的. 这两个过程可以迭代进行, 直至得到满足要求的结果. 与以往的算法相比, 文中算法能保证结果收敛, 迭代次数更少, 且能得到更好的结果.

关键词 可展曲面; 尽可能刚性; 最小范数法; 最小二乘法; 迭代

中图法分类号 TP391 **DOI 号**: 10.3724/SP.J.1016.2010.00927

Developability Optimization Algorithm for 3D Mesh Surfaces

XU Peng-Fei LIU Li-Gang

¹⁾(Department of Mathematics, Zhejiang University, Hangzhou 310027)

²⁾(State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310027)

Abstract Developable surface has zero Gaussian curvature at every point and has lots of good properties. It has been widely used in various applications in industry. Therefore, it is practically useful and important to approximate mesh surfaces using developable surfaces or near-developable surfaces. This paper presents an efficient algorithm for generating a near developable mesh surface to approximate a given mesh surface as close as possible while preserving the local structures of the surface. First, the initial mesh is obtained by using the least-norm method. The initial mesh has good developability property but it does not preserve the local structure well. Then the as-rigid-as-possible (ARAP) approach is used to optimize the initial mesh by preserving the local structures of the original mesh. This paper proposes two methods, i. e. the cell-based and the triangle-based, to preserve the local structures in the ARAP approach. Both methods work well in above algorithm. The initial mesh and the ARAP optimization can be applied in an iterative way. Experimental results show that the algorithm is convergent has obtained better results with less iterations than the previous methods.

Keywords developable surface; as-rigid-as-possible; least-norm; least-squares method; iteration

1 引言

可展曲面可以无扭曲地展开到一个平面上,其高斯曲率处处为零^[1-2].在工业生产中,许多地方要将二维的材料贴合到三维曲面上.例如,在造船工业中,要将金属条片与船身贴合;在制衣业中,要将二维的布料缝合成完整的衣服;在家具制造业中,要将一些皮革蒙在家具表面;在建筑业中,要将一些板材置于特定的位置,并满足一些特定形状.而可展曲面正好具有这样的性质,即可以等距映射到平面上.如果这些材料所贴合的部位是可展曲面,则将这些材料与此部位贴合时,不用进行任何扭曲或拉伸.这样,材料本身内部所受的应力会大大减少,可以显著提高材料的使用寿命.然而,很多时候,材料所要贴合的部位不具有可展的性质.这就促使人们研究是否可以在保证原物体形状变化不大的情况下,将其表面修改为可展曲面.在运用计算机辅助设计时,人们可以通过对三维模型进行处理,来达到目的.因此,将三维模型表面处理为近似可展曲面,有很大的现实意义.

关于可展曲面的逼近已有一些相关工作.文献[3-5]研究了利用直纹曲面等等距曲面对原模型进行逼近. Kilian 等^[6]研究了如何在平面上生成曲线折痕,使得该平面沿着折痕折叠后生成可展曲面. Rose 等^[7]提出了一种以交互输入的曲线作为边界曲线的可展曲面生成算法.

Wang^[8-9]提出了一系列曲面在离散情况下的可展处理.然而这些方法虽然最终能得到可展曲面的结果,但是对原曲面的破坏较大.本文的方法借鉴了文献[8]中的方法,同时保持原模型的结构,使得生成的可展曲面的结果既能够满足可展性质,又能保持原模型网格的结构.本文及文献[8]的方法都需要通过迭代来生成最终的结果.但是文献[8]中的方法在最后的几步迭代时需要进行特别处理,才能得到最终收敛的结果.而本文的方法在应用时不需要进行这样的特殊处理.

保留原模型的结构的方法,很多工作都有涉及.本文利用了文献[10]中提出的尽可能刚性(as-rigid-as-possible)的方法,对模型形状进行改变时,模型的每一部分都尽量在做旋转或平移变换,而尽量不出现拉伸或扭曲的变换.在满足这个条件时,模型的整体变换能够保持结构尽量不变.另外,本文也借鉴了 Liu 等在文献[11]中将三角面片作为旋转单位的方法.利用这些方法,本文得到的结果能够很好地

保持原模型的结构特征.

问题描述

对于给定的一个模型,需要将其表面处理为可展曲面或近似可展曲面.在计算机图形学中,模型的表面一般可离散表示为三角网格曲面.在本文中,我们只讨论具有单一边界的网格曲面,即曲面与一个圆盘同胚.对于复杂的模型曲面,我们可以将其进行分块,使得每一小块都与圆盘同胚,然后对每一块分别处理.

我们采用文献[9]中提出的网格顶点的可展性度量:如果在某个顶点处,与这个顶点相连的内角之和等于 2π ,则认为在这个顶点处,网格满足可展的性质.对于一个给定的单边界网格曲面,如果每个内点(即非边界点)都满足可展性质,则认为这个网格曲面是可展的.显然,一般的三角网格曲面并不满足可展性质.本文研究的问题为:给定一个三角网格曲面,如何改变网格顶点的位置,使得生成的新的三角网格曲面满足可展性,并且能很好逼近初始给定的网格曲面.

2 基于 ARAP 方法的可展性优化算法

文献[8]提出了一种基于最小范数的方法来将一个三角网格曲面优化成近似可展曲面,其基本思想就是移动顶点坐标使得每个顶点满足可展性质,即每个顶点的内角之和等于 2π .我们利用这个方法对网格进行处理,得到新的网格,称为初始预测.由于初始预测不能保持原始网格的形状,并且顶点较不规则,我们能够在初始预测的基础上进行优化.

为了能在优化过程中尽可能地保持原始曲面,我们采用文献[10]中的尽可能刚性(ARAP)的方法,来对网格进行优化.优化过程保持了原始网格曲面的形状,使得顶点较为规则.同时达到可展的要求.

上述两步,即初始预测和尽可能刚性优化,可以通过迭代不断改进结果,最终得到很好的结果.

下面具体介绍本文的算法.

2.1 利用最小范数方法得到初始预测

对于每次迭代,先利用最小范数方法^[8]得到初始预测网格,可通过以下优化问题求得

$$\begin{aligned} \min_{p \in V_{\text{int}}} J_{\text{pos}} &= \sum_{p \in V_{\text{int}}} \|v_p - v_p^o\|^2 \\ \text{s. t. } \theta(v_p) &\equiv 2\pi \end{aligned} \quad (1)$$

其中, V_{int} 是网格所有内点下标的集合, v_p^o 是网格的

初始顶点, \mathbf{v}_p 是新的顶点位置. 可见未知数只是网格的内点位置.

设 $\theta(\mathbf{v}_p)$ 为 \mathbf{v}_p 以及 \mathbf{v}_p 的邻点的一个函数. 记 $\mathbf{v}_p = \mathbf{v}_p^o + \mathbf{d}_p$, 通过泰勒展开^[8], 可以得到

$$\theta_p(\mathbf{v}_p^o) \cdot \mathbf{d}_p + \sum_{q \in N_V(p)} \theta_q(\mathbf{v}_p^o) \cdot \mathbf{d}_q \approx 2\pi - \theta(\mathbf{v}_p^o) \quad (2)$$

其矩阵形式为

$$\mathbf{A}\mathbf{d} = \mathbf{b} \quad (3)$$

$\theta_p(\mathbf{v}_p^o)$ 以及 $\theta_q(\mathbf{v}_p^o)$ 的求解公式在文献[9]中有描述.

由于式(3)是欠定的, 有无穷多组解. 为使新网格曲面与初始曲面尽量接近, 我们可以求一组解, 使 $\|\mathbf{d}\|^2$ 最小. 即为最小范数问题:

$$\mathbf{d} = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{b} \quad (4)$$

这里矩阵 \mathbf{A} 是稀疏矩阵, 所以可以有效的求得 \mathbf{d} , 进而求得 \mathbf{v}_p . 这样, 就得到了整个过程的第一步, 即得到了初始预测.

2.2 尽可能刚性 (ARAP) 方法

利用上述的最小范数方法, 我们可以得到近似可展的新曲面. 但是, 新曲面与原始曲面相比, 顶点位置变得较为不规则, 整体形状也有较大变化. 因此需要对其进行优化.

尽可能刚性的方法使得网格在变形的时候, 尽量保持其局部性质. 局部在进行变化时, 尽量只做旋转或平移变化, 而不做拉伸和扭曲的变化. 我们根据最小范数求解得到的顶点和初始网格的顶点可得到每个局部的旋转矩阵, 之后再利用这些旋转矩阵, 对初始顶点 \mathbf{v}_p^o 进行处理, 得到优化后的网格曲面.

在文献[10]中, 局部被定义为: 网格曲面的内点及其一环邻域所组成的一块, 称为 cell. 在文献[11]中, 定义每个三角面片为一个局部. 这两种方法各有优点, 最终得到的结果也都十分理想. 下面对这两种方法分别进行讨论.

2.2.1 以 cell 为局部单位

我们已经得到对原始网格的初始预测. 对于每个 cell, 我们希望它们的变化都可以用旋转变换来进行逼近. 假设旋转矩阵为 \mathbf{R}_p , 则有

$$\mathbf{e}_{pq} = \mathbf{R}_p \mathbf{e}_{pq}^o, \quad \forall q \in N_V(p) \quad (5)$$

其中

$$\mathbf{e}_{pq} = \mathbf{v}_p - \mathbf{v}_q, \quad \mathbf{e}_{pq}^o = \mathbf{v}_p^o - \mathbf{v}_q^o \quad (6)$$

实际中, 不可能每个局部的变化都能严格地满足式(5). 因此, 可以定义一个误差函数, 即, 对于每个 cell, 有

$$E(p) = \sum_{q \in N_V(p)} \omega_{pq} \|\mathbf{e}_{pq} - \mathbf{R}_p \mathbf{e}_{pq}^o\|^2 \quad (7)$$

其中 ω_{pq} 是对每条边 \mathbf{e}_{pq} 的权重.

整体上, 总的误差函数为

$$E = \sum_{p \in V_{\text{int}}} \omega_p E(p) \quad (8)$$

在本文中, ω_p 取值均为 1.

我们希望得到一系列的旋转矩阵, 使得式(8)最小. 即, 尽量使初始网格每个局部进行旋转变换来逼近初始预测.

对于每个 cell, 由式(7)可得

$$E(p) = \sum_{q \in N_V(p)} \omega_{pq} (\mathbf{e}_{pq}^T \mathbf{e}_{pq} - 2\mathbf{e}_{pq}^T \mathbf{R}_p \mathbf{e}_{pq}^o + \mathbf{e}_{pq}^{oT} \mathbf{e}_{pq}^o) \quad (9)$$

上式中, 不包含 \mathbf{R}_p 的项是常数项, 可以不用考虑. 要令上式最小, 等价于寻找合适的 \mathbf{R}_p , 使 $\sum_{q \in N_V(p)} \omega_{pq} \mathbf{e}_{pq}^T \mathbf{R}_p \mathbf{e}_{pq}^o$ 最大. 而

$$\begin{aligned} \arg \max_{\mathbf{R}_p} \sum_{q \in N_V(p)} \omega_{pq} \mathbf{e}_{pq}^T \mathbf{R}_p \mathbf{e}_{pq}^o = \\ \arg \max_{\mathbf{R}_p} \text{Tr} \left(\mathbf{R}_p \sum_{q \in N_V(p)} \omega_{pq} \mathbf{e}_{pq}^o \mathbf{e}_{pq}^{oT} \right) \end{aligned} \quad (10)$$

并记

$$\mathbf{S}_p = \sum_{q \in N_V(p)} \omega_{pq} \mathbf{e}_{pq}^o \mathbf{e}_{pq}^{oT} \quad (11)$$

则式(10)可以写作

$$\arg \max_{\mathbf{R}_p} \text{Tr} (\mathbf{R}_p \mathbf{S}_p) \quad (12)$$

当 $\mathbf{R}_p \mathbf{S}_p$ 是对称半正定矩阵时, 上式取到最大值. 为了使之满足对称半正定矩阵的条件, 可以通过以下的方法来得到 \mathbf{R}_p . 首先将 \mathbf{S}_p 进行 SVD 分解^[10]. 则可以得到

$$\mathbf{S}_p = \mathbf{U}_p \boldsymbol{\Sigma}_p \mathbf{V}_p^T \quad (13)$$

其中, \mathbf{U}_p 和 \mathbf{V}_p 是正交矩阵, $\boldsymbol{\Sigma}_p$ 是对角矩阵, 对角线上的值为 \mathbf{S}_p 的奇异值. 同时, 令 \mathbf{R}_p 等于

$$\mathbf{R}_p = \mathbf{V}_p \mathbf{U}_p^T \quad (14)$$

在实际计算时, 适当改变 \mathbf{U}_p 每一行的符号, 使得 \mathbf{R}_p 满足作为旋转矩阵的要求. 此时有

$$\mathbf{R}_p \mathbf{S}_p = \mathbf{V}_p \mathbf{U}_p^T \mathbf{U}_p \boldsymbol{\Sigma}_p \mathbf{V}_p^T = \mathbf{V}_p \mathbf{I} \boldsymbol{\Sigma}_p \mathbf{V}_p^T = \mathbf{V}_p \boldsymbol{\Sigma}_p \mathbf{V}_p^T \quad (15)$$

即此时 $\mathbf{R}_p \mathbf{S}_p$ 是对称半正定矩阵. 在具体求解时, 式(10)中权重 ω_{pq} 采用 cotangent 权系数^[14], 即下式:

$$\omega_{pq} = \frac{1}{2} (\cot \alpha_{pq} + \cot \beta_{pq}) \quad (16)$$

图 1 为角度关系. 可以发现, 有 $\omega_{pq} = \omega_{qp}$ 成立.

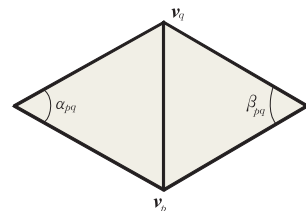


图 1 角度 α_{pq} 和 β_{pq}

得到每个 cell 的旋转矩阵之后, 再将这些旋转矩阵作用到初始网格上来进行优化, 即, 作用到顶点 \mathbf{v}_p^o . 在实际操作时, 由于局部之间的相互影响, 不能保证每个局部都能够进行旋转变换. 可以考虑找到一组解 \mathbf{v}_p , 使得式(8)最小. 对式(8)中的变量 \mathbf{v}_p 求偏导, 并令其值为 0, 则有

$$\frac{\partial E}{\partial \mathbf{v}_p} = \sum_{q \in N_V(p)} 4\omega_{pq} \left((\mathbf{v}_p - \mathbf{v}_q) - \frac{1}{2} (\mathbf{R}_p + \mathbf{R}_q) (\mathbf{v}_p^o - \mathbf{v}_q^o) \right) = 0 \quad (17)$$

变换上式, 可以得到

$$\sum_{q \in N_V(p)} \omega_{pq} (\mathbf{v}_p - \mathbf{v}_q) = \sum_{q \in N_V(p)} \frac{\omega_{pq}}{2} (\mathbf{R}_p + \mathbf{R}_q) (\mathbf{v}_p^o - \mathbf{v}_q^o) \quad (18)$$

改写上式为矩阵形式, 可以得到

$$\mathbf{L}_c \mathbf{V} = \mathbf{g}_c \quad (19)$$

其中 \mathbf{L}_c 是一个矩阵. \mathbf{V} 是网格所有内点组成的向量, \mathbf{g}_c 是通过式(18)右边得到的向量. 由于所有的 ω_{pq} 是从原始网格得到的, 所以每次迭代时, \mathbf{L}_c 都是一样的, 即 \mathbf{L}_c 只需要计算一次. 而 \mathbf{g}_c 需要每次迭代时利用当前的网格信息以及初始预测得到.

2.2.2 以三角面片为局部单位

当以三角面片为局部单位时, 方法和以 cell 为局部单位相同. 但是实际操作时有一些区别. 当以三角面片为局部单位时, 三个顶点以三角形的重心为轴, 做旋转变换. 对于某个三角面片 f_i , 设它的顶点做旋转变换的旋转矩阵为 \mathbf{R}_i , 重心位置为 \mathbf{v}_g , 则可以得到下式:

$$\mathbf{l}_i = \mathbf{R}_i \mathbf{l}_i^o, \quad i=0, 1, 2 \quad (20)$$

其中

$$\mathbf{l}_i = \mathbf{v}_i - \mathbf{v}_g, \quad \mathbf{l}_i^o = \mathbf{v}_i^o - \mathbf{v}_g^o \quad (21)$$

和以 cell 为局部单位类似, 可以对每个三角面片定义一个误差函数:

$$E(t) = \sum_{i=0}^2 \|\mathbf{l}_i - \mathbf{R}_i \mathbf{l}_i^o\|^2 \quad (22)$$

整体误差函数为

$$E = \sum_t \omega_t E(t) \quad (23)$$

其中, ω_t 取值为 1.

同样用 SVD 分解, 类似可以求得旋转矩阵 \mathbf{R}_i . 现在要计算新的顶点. 对式(23)中的变量 \mathbf{v}_p 求偏导, 并令其为 0, 有

$$\frac{\partial E}{\partial \mathbf{v}_p} = \sum_{t \in N_F(p)} \frac{\partial E(t)}{\partial \mathbf{v}_p} = 0 \quad (24)$$

上式中, $N_F(p)$ 表示与顶点 \mathbf{v}_p 相连的三角面片的指

标集. 若 $t \in N_F(p)$, 则三角面片 f_t 中包含 \mathbf{v}_p 作为顶点. 不妨假设 \mathbf{v}_p 是所有三角面片 f_t 的第一个顶点. 则可以得到下式:

$$\frac{\partial E}{\partial \mathbf{v}_p} = \frac{2}{3} \sum_{t \in N_F(p)} (2\mathbf{v}_p - \mathbf{v}_{t1} - \mathbf{v}_{t2} + \mathbf{R}_t (\mathbf{v}_{t1}^o + \mathbf{v}_{t2}^o - 2\mathbf{v}_p^o)) = 0 \quad (25)$$

即

$$\sum_{t \in N_F(p)} (2\mathbf{v}_p - \mathbf{v}_{t1} - \mathbf{v}_{t2}) = \sum_{t \in N_F(p)} \mathbf{R}_t (2\mathbf{v}_p^o - \mathbf{v}_{t1}^o - \mathbf{v}_{t2}^o) \quad (26)$$

改写上式为矩阵形式, 可以得到

$$\mathbf{L}_t \mathbf{V} = \mathbf{g}_t \quad (27)$$

其中 \mathbf{L}_t 是一个矩阵. \mathbf{V} 是网格所有内点组成的向量, \mathbf{g}_t 是通过式(26)右边得到的向量. 与以 cell 为局部单位时类似, 矩阵 \mathbf{L}_t 也是只需要计算一次, \mathbf{g}_t 需要在每次迭代时利用当前网格信息以及初始预测得到.

2.3 利用最小二乘方法对网格进行优化

利用 ARAP 方法, 可以保留原始网格的信息. 结合最小范数方法得到的初始预测, 可以对网格曲面顶点进行优化. 然而在实际操作时发现, 如果直接利用式(19)或(27)进行计算, 则并不能得到很好的结果: 计算时, 迭代次数过多, 对于某些模型, 甚至不能得到收敛的结果. 这是因为在求初始预测时, 泰勒展开式的高次项被忽略了. 当初始网格曲面的可展性很差时, 计算的误差也较大. 而 ARAP 方法只是保留初始网格的结构, 并没有提供可展的限制. 如果利用 ARAP 方法对网格曲面进行优化, 则会使得初始预测的误差进一步加大. 为了加入可展的限制, 可以加入最小范数步骤中的线性方程, 即式(3), 从而得到

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{L} \end{pmatrix} (\mathbf{V} - \mathbf{V}^o) = \begin{pmatrix} \mathbf{b} \\ \mathbf{g} - \mathbf{L}\mathbf{V}^o \end{pmatrix} \quad (28)$$

由于式(28)中的系数矩阵有可能是超定的, 不一定有精确解. 因此我们在最小二乘意义下求得方程的最优解^[12], 即找到一组解, 使下式最小

$$E = \left\| \begin{pmatrix} \mathbf{A} \\ \mathbf{L} \end{pmatrix} \mathbf{d} - \begin{pmatrix} \mathbf{b} \\ \mathbf{g}' \end{pmatrix} \right\|^2 \quad (29)$$

上式中, $\mathbf{d} = \mathbf{V} - \mathbf{V}^o$, $\mathbf{g}' = \mathbf{g} - \mathbf{L}\mathbf{V}^o$. 则可以解得式(28)的最小二乘解为

$$\mathbf{d} = \left((\mathbf{A}^T \quad \mathbf{L}^T)^T \begin{pmatrix} \mathbf{A} \\ \mathbf{L} \end{pmatrix} \right)^{-1} (\mathbf{A}^T \quad \mathbf{L}^T)^T \begin{pmatrix} \mathbf{b} \\ \mathbf{g}' \end{pmatrix} \quad (30)$$

之后结合 $\mathbf{V} = \mathbf{V}^o + \mathbf{d}$, 可以得到优化后的顶点位置.

2.4 最小范数步骤时的一些权重

在实现过程中, 我们还可以将已经很接近可展的顶点和未接近可展的顶点来区分对待, 让未接近

可展的顶点更容易获得较大的变化,则可以更快地令结果收敛.具体只要将不需要更多变化的顶点加上较大的权重,若这个顶点移动很小的距离,则目标函数的值便会显著增加.由此便限制了这些顶点的改变.

具体的权重设置方法如下(这里的数值是实验得到的结果,可以根据需要进行调整):

(1) 得到当前所有内点处,与这个顶点相连的内角之和,与 2π 的差 dif_p .

(2) 设定两个阈值(threshold value),分别为 $tv_l=0.005, tv_u=0.5$. 对于某个顶点,如果 dif_p 大于 tv_u ,则将此顶点处的权重 w_p 设为 1. 如果 dif_p 小于 tv_l ,则将此顶点处的权重 w_p 设为 100. 如果 dif_p 在 tv_u 与 tv_l 之间,则由 $w_p = \frac{tv_u}{dif_p}$ 得到此顶点处的权重.同时,将相应的权重乘到式(3)中 \mathbf{A} 的相应列.

(3) 令 $\mathbf{v}_p = \mathbf{v}_p^o + w_p' \mathbf{d}_p$, 其中 w_p' 为顶点位置改变的权重. 本文在实际应用时,设其上限为 0.9, 下限为 0.1. 设 $dif_p' = (dif_p + \sum_{q \in N_{V_{in}}(p)} dif_q) / |N_{V_{in}}(p)|$,

即 dif_p' 是这个顶点处的 dif_p 及其内部邻点的 dif_p 的均值. 将 dif_p' 与 tv_l 比较,如果前者较大,则设 $w_p' = 0.9$, 否则设 $w_p' = 0.1$.

根据实验结果来看,这样设置权重能够得到不错的结果.

3 结果及比较

我们利用本文提出的算法对许多模型进行了试验和处理. 算法适用于处理一些近似可展的模型曲面,而并不适用于所有的模型,尤其是一些复杂的,含有很多细节信息的模型. 在实际应用时,如果需要处理的模型较复杂,可以先将其分块,然后对各块进行处理,再进行拼接. 各图中的颜色表示曲面的可展程度. 颜色接近蓝色,表示可展程度较好;接近红色,表示可展程度较差. 颜色对应数字是指某顶点处内角和与 2π 的差值.

本文的算法可以处理类似于图 2 的模型. 我们可以先将其分为前后两块,然后分别进行处理.

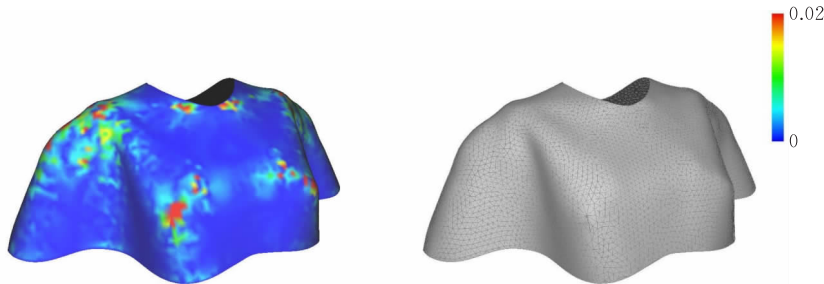


图 2 实验模型

由图 2 中可以看到,虽然 Cloak 模型很多地方接近可展,但仍有一些部分要进行处理. 本文将其分割为前后两部分进行处理. 如果进行更细的分割,则能得到更好的结果. 图中灰度表示模型可展程度. 颜色接近浅色,表示可展程度较好;接近深色,表示可展程度较差. 灰度对应数字是指某顶点处内角和与

2π 的差值.

在图 3 和图 4 中,我们和文献[8]的方法进行了比较. 文献[8]的方法需要对迭代进行一些特别处理才能得到收敛的结果,而我们的方法不需要任何处理都能够收敛. 并且结果显示,本文的方法能得到更优的结果.

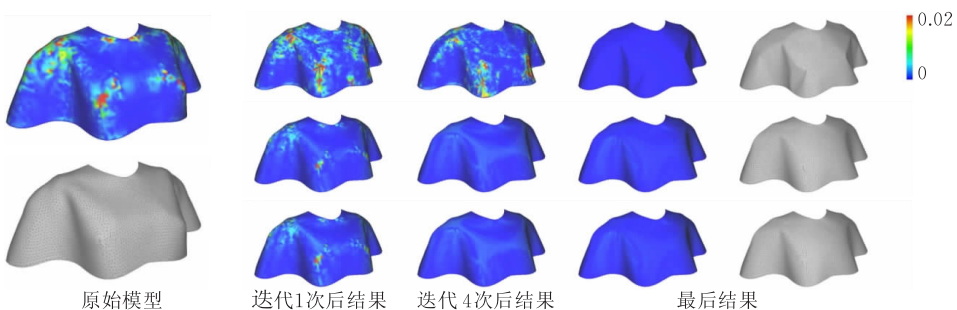


图 3 Cloak 模型前半部分的处理(左边大图是原始模型,左上图的颜色代表了可展程度,左下图为网格模型. 右边小图是不同方法的处理结果,其中第 1 行是文献[8]中方法处理的结果,第 2 行是以 Cell 为局部单位的结果,第 3 行是以三角面片为单位的结果)

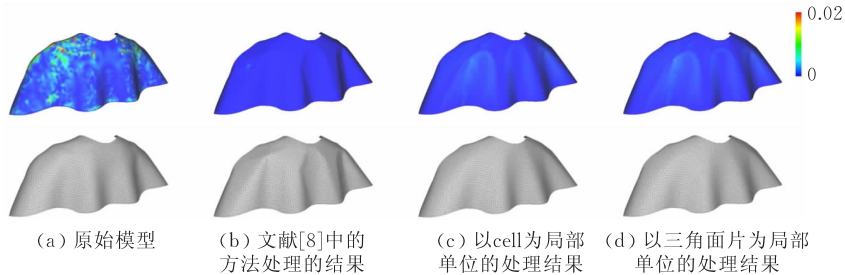


图 4 对 Cloak 模型背部的处理结果

图 3 是 Cloak 模型的前半部分. 从最终的结果来看本文方法得到的结果在保持原模型的形状上要好一些. 而在处理过程中, 可以发现经过 1 次, 以及 4 次迭代后, 本文的结果要明显更接近可展. 而文献[8]中的方法收敛速度却没有那么快. 为了得到最终收敛的结构, 用文献[8]中的方法有时需要采取一些特别的措施. 而本文的方法不需要区别对待, 只要不断的进行迭代, 就会得到收敛的结果. 这也是这个算法的优势之一.

图 4 是 Cloak 模型的后半部分. 这个结果同样显示了本文的结果能更好地保持原模型的形状. 将图 3 与图 4 中的处理结果相结合, 就可以得到对整个 Cloak 模型处理的结果

3.1 计算中的数据

表 1 是本文用到的一些模型的数据. 表 2 是处理模型的一些数据统计. 其中, E_θ 是指所有顶点处内角和与 2π 差值的最大值. E_v 是指所有顶点偏离初始位置的最大值. 表 2 中, 最小范数方法想要得到收敛的结果, 需要在迭代若干步后, 进行一些特殊处理

(具体请参见文献[8]). 而本文的方法没有这样的情况. 在表 2 中, 处理模型并没有设置终止值, 而仅仅是不断地迭代. 当结果相对稳定时终止迭代. 根据表 2 中的数据, 可以发现, 用最小范数方法得到的结果具有较好的可展性, 一般 E_θ 比本文的结果要小. 但是考虑实际情况, 当 E_θ 达到 10^{-3} 数量级时, 已经没有必要再进行迭代, 因为此时已经足够可展. 事实上, 只要迭代足够多的次数, 本文的方法也是可以达到更小的 E_θ 的. 如果考虑到与原始模型的接近程度, 就可以看到本文方法的优势了. 如表 2 所示, 用本文的方法, 最后的 E_v 要比最小范数方法小得多. 这说明了这个方法在保持模型初始位置上要优于最小范数方法. 通过对图 3~图 6 等图的观察也可以发现, 本文的方法在保留原模型的形状上要更好一些.

表 1 所用模型的数据

模型	面片数	顶点数
Patch	1224	656
Cloak front	7070	3758
Cloak back	5504	2964
Skirt	21415	10991

表 2 处理模型的计算数据(PC 数据: Intel Core Duo processor T2300E, 1.66GHz, 512MB RAM)

(其中 E_θ 指所有顶点处内角和与 2π 差值的最大值, E_v 是指所有顶点偏离初始位置的最大值)

模型	方法	时间(每次迭代)/ms	步数	E_θ	E_v
Patch	Least-norm	78	20	6.557×10^{-7}	3.093×10^{-2}
	LS with ARAP(cell)	281	20	1.852×10^{-3}	3.056×10^{-4}
	LS with ARAP(triangle)	297	20	1.990×10^{-3}	4.195×10^{-4}
Cloak front	Least-norm	672	50	1.958×10^{-3}	1.131×10^{-1}
	LS with ARAP(cell)	2375	5	2.330×10^{-3}	6.534×10^{-3}
	LS with ARAP(triangle)	2485	6	2.705×10^{-3}	5.483×10^{-3}
Cloak back	Least-norm	531	24	5.901×10^{-5}	2.014×10^{-2}
	LS with ARAP(cell)	1656	10	1.745×10^{-3}	1.555×10^{-3}
	LS with ARAP(triangle)	1719	15	1.670×10^{-3}	1.054×10^{-3}
Skirt	Least-norm	1797	19	3.920×10^{-4}	1.423×10^{-2}
	LS with ARAP(cell)	7547	4	1.294×10^{-3}	2.077×10^{-3}
	LS with ARAP(triangle)	8015	4	1.559×10^{-3}	2.505×10^{-3}

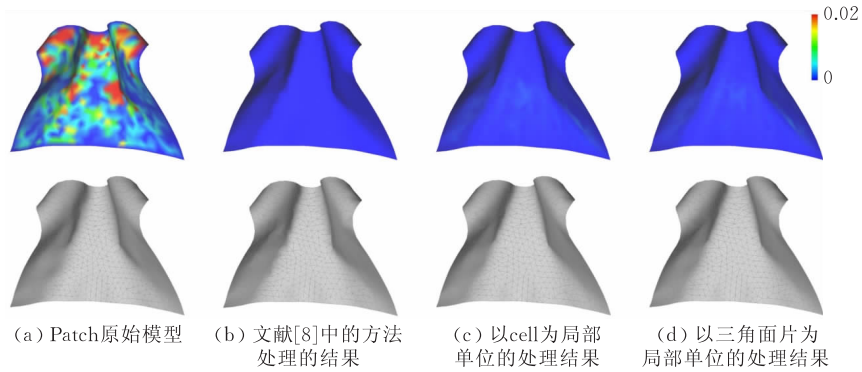


图 5 对 Patch 模型的处理结果

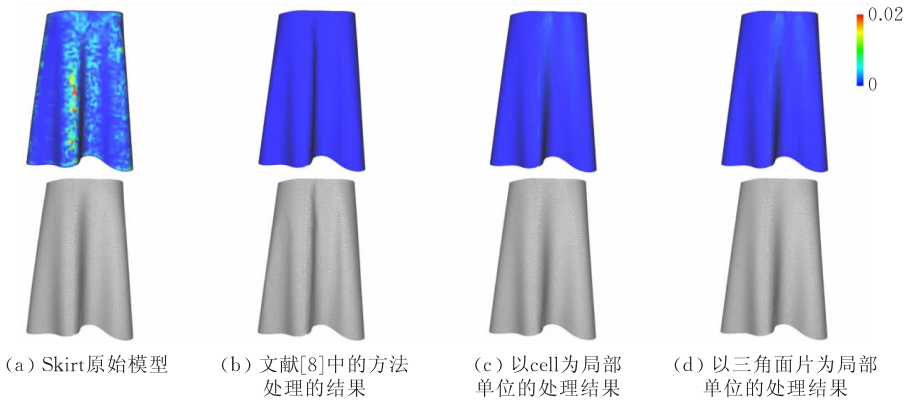


图 6 对 Skirt 模型的处理结果

4 总结与结论

本文给出的算法在得到理想的可展曲面的同时,还保证了原模型的网格结构尽量不被破坏,这是算法的优势所在.从这篇文章中一些例子也可以看到,最后得到的结果也确实在保持原模型结构上较优.本文的算法在实际应用时,只需设定一个可展性的目标值,然后不断地进行迭代即可.这使得这个方法更易被应用.另外,从实验的数据来看,被处理的模型不会出现与原形状相差过大的情况,这使得本文的算法能够胜任处理各种模型.

本文算法也存在着一些不足.由于本文算法结合了 ARAP 方法,使得计算量增加较多,因此每次迭代所需要的时间较长.

本文的方法虽然取得了一些结果,但是仍有改进的余地.首先,可以考虑对顶点赋予不同的权重,以得到更好的结果.其次,这个方法并没有加入人工交互的成分,只是单纯的对网格进行处理.但是在实际应用时,要处理的模型表面会有一些特殊情况,比如某些部分不需要处理,或者某些顶点位置不能移动.这个时候,可以利用交互的方法来选出这些顶

点,然后将其固定或赋以特殊的权重,使得处理后能够得到人们需要的结果.

参 考 文 献

- [1] Opera J. *Differential Geometry and Its Applications*. Beijing: China Machine Press, 2005
- [2] Do Carmo M P. *Differential Geometry of Curves and Surfaces*. Englewood Cliffs, NJ: Prentice-Hall, 1976
- [3] Chu C, Séquin C. Developable bézier patches: properties and design. *Computer-Aided Design*, 2002, 34(7): 511-527
- [4] Leopoldseder S, Pottmann H. Approximation of developable surfaces with cone spline surfaces. *Computer-Aided Design*, 1998, 30(7): 571-582
- [5] Chen H, Lee I, Leopoldseder S, Pottmann H, Randrup T, Wallner J. On surface approximation using developable surfaces. *Graphical Models and Image Processing*, 1999, 61(2): 110-124
- [6] Kilian M, Floery S, Chen Z, Mitra N J, Sheffer A, Pottmann H. Curved folding. *ACM Transactions on Graphics*, 2008, 27(3): 1-9
- [7] Rose K, Sheffer A, Wither J, Cani M, Thibert B. Developable surfaces from arbitrary sketched boundaries//*Proceedings of Eurographics Symposium on Geometry Processing*. Barrvelona, Spain, 2007: 163-172

- [8] Wang C. A least-norm approach to flattenable mesh surface processing//Proceedings of the IEEE International Conference on Shape Modeling and Applications. Stony Brook, New York, USA, 2008: 131-138
- [9] Wang C. Towards flattenable mesh surfaces. *Computer-Aided Design*, 2008, 40(1): 109-122
- [10] Sorkine O, Alexa M. As-rigid-as-possible surface modeling//Proceedings of the Eurographics Symposium on Geometry Processing. Barcelona, Spain, 2006: 109-116
- [11] Liu L, Zhang L, Xu Y, Gotsman G, Gortner S J. A local/global approach to mesh parameterization. *Computer Graphics Forum*, 2008, 27(5): 1495-1504
- [12] Sorkine O, Cohen-Or D. Least-squares mesh//Proceedings of Shape Modeling International. Genova, Italy, 2004: 191-194
- [13] Desbrun M, Meyer M, Alliez P. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum*, 2002, 21(3): 209-218
- [14] Pinkall U, Polthier K. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 1993, 2: 15-36



XU Peng-Fei, born in 1986. His research interests include computer graphics, geometric processing, etc.

LIU Li-Gang, born in 1975, associate professor. His research interests include computer graphics, geometric processing, digital city and image processing, etc.

Background

Developable surface has zero Gaussian curvature at every point and has lots of good properties. They are widely used in industrial applications, including shipping industry, making cloths, and architecture etc., due to their nice capability of being made from a flat sheet without any stretching and tearing. Hence, developable surfaces have attained a lot of research interests in the field of computer aided geometric design and computer graphics. With the development of 3D acquisition technologies and hardware devices, discrete polygonal mesh surfaces have become increasingly popular. Therefore, it is practically useful and important to approximate mesh surfaces using developable surfaces or near-developable surfaces in digital geometry processing.

Many researches on approximating mesh surfaces using developable or nearly-developable surfaces have been done in the literature. These researches focused on different aspects of the developable surfaces. Some works were proposed to use special kind of developable surfaces, like ruled surfaces, to approximate the original surfaces. The other works proposed tools to deal with the piecewise planar surfaces and the developable surfaces can be easily constructed. All these works have been successfully used in practical applications. However, these approaches have some disadvantages in some aspects either in approximation quality or in robustness.

Most of the approaches often fail to preserve the original shapes well.

This paper proposes a new and efficient algorithm for generating a near developable mesh surface to approximate a given mesh surface as close as possible. Different from the previous approaches, our algorithm can preserve the local structures of the surface as much as possible. Therefore, our algorithm can obtain high quality mesh surface with good developability properties. This is especially important for applications like texture mapping and painting.

The basic idea is to use the as-rigid-as-possible (ARAP) approach to improve the results obtained by the least-norm method. The ARAP approach is a good approach that can preserve the local structures of the mesh. The authors propose two methods, i. e., the cell-based and the triangle-based, to preserve the local structures in the ARAP approach. Both methods work well in our algorithm. They have done lots of experiments on various mesh surfaces. Experimental results show that the algorithm has obtained better results with less iterations than the previous methods.

This work is supported by the joint grant of the National Natural Science Foundation of China and Microsoft Research Asia (60776799) and the 973 National Key Basic Research Foundation of China (No. 2009CB320801).