

基于 GPU 并行优化的网格参数化算法

吴璇, 张举勇

(中国科学技术大学 数学科学学院, 安徽 合肥 230026)

摘要: 网格参数化是计算机图形学、数字几何处理领域的研究热点, 在动画、医疗、工业设计等领域中都发挥着重要作用。现有参数化方法主要思路是构造一个高度非线性的全局优化问题, 因此计算效率低, 难以并行。提出了一种可并行、可扩展的参数化算法。该算法通过引入辅助变量。然后使用交替方向乘子算法 (Alternating Direction Method of Multipliers, ADMM), 迭代优化每个面和每条边上的子问题得到参数化映射。为了验证算法模型的高效性, 使用 GPU 加速, 相比于现存单线程算法, 本文算法因为高度并行化运行时间缩短了至少百倍以上。

关键词: 并行计算; 交替方法乘子法; 网格参数化; 优化算法

中图分类号: TP391

文献标识码: A

DOI: 10.19358/j.issn.2096-5133.2020.09.004

引用格式: 吴璇, 张举勇. 基于 GPU 并行优化的网格参数化算法[J]. 信息技术与网络安全, 2020, 39(9): 16-23.

Mesh parameterization based on GPU parallel optimization

Wu Xuan, Zhang Juyong

(School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, China)

Abstract: Mesh parameterization is a research hotspot in the field of computer graphics and digital geometry processing. It plays an important role in animation, medical treatment, industrial design and other fields. Existing methods formulate this problem as a global optimization problem. Due to its high nonlinearity and global optimization of the model, it is very difficult to solve efficiently and parallelize. This paper presents a parallel and scalable algorithm for mesh parameterization. The proposed method solves this problem by introducing a set of auxiliary variables. Then using ADMM (Alternating Direction Method of Multipliers), this problem can be easily solved by optimizing small problems for each face and each edge iteratively. To verify the efficiency of the proposed method, we implement the proposed algorithm via GPU, reduce the running time by at least 100 times compared with single thread implementation due to the high parallelism of the proposed algorithm.

Key words: parallel computing; ADMM; mesh parameterization; optimization algorithm

0 引言

三维模型是一种使用三维曲面来表述物体的三维数据, 网格是三维模型中一种应用广泛的表达方式。随着数字几何处理技术的发展以及扫描技术的进步, 网格模型得以广泛应用于动画、游戏、建筑、医疗、工业设计等行业。网格曲面参数化是流形曲面和参数域之间的一一映射, 是网格处理领域中不可或缺的基础工具, 在网格变形、纹理映射、网格压缩中都发挥着重要作用。通常网格是在 3D 空间中的二维曲面, 直接对于 3D 模型进行网格处理非常复杂, 通过一一映射到简单的参数域, 得到

的参数化结果与原始网格有相同的拓扑结构以及尽可能小的失真, 然后在参数域上进行网格处理, 极大地降低了处理难度。

一个高质量的参数化映射 f 有以下性质: 无翻转、低失真度量。无翻转意味着 $\det J(f) > 0$, 这里 $J(f)$ 是 f 的雅各比矩阵。理想中的映射是在映射后网格与初始网格之间没有形变, 但这只是理想情况, 一个高质量的网格需要尽量减少形变, 而失真度量就是用于衡量映射形变的数值。

经典的参数化方法主要分为线性方法与非线性方法两种。线性方法计算简单, 可扩展性强, 因为

线性方法通过计算一个线性系统来得到参数化结果。虽然线性方法在计算效率上占据优势,但是有许多方法都必须固定边界,无法获得自由边界的参数化结果,比如针对拓扑圆盘, FLOATER M^[1-2]通过把边界固定到一个凸多边形上,同时所有权重都保证为正数,得到一个无翻转的参数化结果。自由边界的方法可以通过虚拟边界、增添线性方程来实现。自由边界方法通常可以减少固定边界造成大的变形扭曲,却不一定确保得到的映射是无翻转的。非线性方法通常构造出一个以变形能量为目标式,包含无翻转硬约束的全局优化问题^[3-4],使用牛顿法、高斯牛顿法等优化算法降低参数化网格的变形能量,这些能量函数描述了参数化映射后网格的变形、失真程度,通常是高度非线性、非凸的,所以这些方法计算效率低,而且在处理大型网格时,非线性方法通常会随着所处理网格的增大,收敛速度极大地降低。

为了解决以往参数化方法运算消耗大、运算效率低、非并行、可扩展性差的缺陷,本文提出了一种可并行、可扩展计算无翻转、高质量参数化网格的算法。不同于以往算法构造出一个无法并行的全局优化问题,本文算法通过引入辅助变量,把参数化问题分解为每个面上、每条内边上的局部子问题。该算法的空间复杂度与网格模型规模成线性关系,也就是 $4N+2|\mathcal{E}_{\text{int}}|$, 其中 N 是网格的面数, $|\mathcal{E}_{\text{int}}|$ 是网格内边条数。相比于现存算法不可并行性,本文算法最大创新点在于每次迭代都可以并行处理 N 个关于三角面片上映射的子问题以及 $|\mathcal{E}_{\text{int}}|$ 个关于内边相容性约束的子问题。实验显示相比于现存算法,本文算法最终得到相同甚至更好质量网格所需运算时间缩短了至少百倍以上。随着扫描技术的飞快发展,3D 网格模型的规模越来越大,可扩展的网格参数化算法意义重大。但计算大规模网格的无翻转映射是一个具有挑战性的难题,该算法可扩展,长于处理大型网格模型。

1 相关工作

1.1 无翻转映射

良好的参数化结果需要具有无翻转的性质。TUTTE 嵌入及其变体在理论上保证参数化映射无翻转,但基本上都会引入大的失真变形。基于表示的方法^[4-5]无法确保消除所有翻转,固定边界方法总是可以产生无翻转参数化映射,但在边界不适

宜的情况下往往带来大的失真变形。基于维护的方法^[6-7]开始优化失真能量的同时保持无翻转。然而,这些数值算法往往需要解决一个全局优化问题,以至于不可并行以及可扩展性差。比如 HORMANN K 等人提出了 MIPS 方法^[7],可以获得自由边界的参数化结果,但是 MIPS 能量计算耗时久且容易陷入局部最小点。有界失真方法,比如 LIPMAN Y 等提出的方法^[8-9]直接限制了失真度的大小,但计算效率低,而且如何设置一个保证无翻转的合适边界成为难题。不同于以上方法,采用了一种局部推进策略来解决复杂的网格参数化问题,算法是可并行和可扩展的。为了产生无翻转映射,一个直接的方法是从一个无翻转的初始值开始优化,然后使用一个防翻转能量,通常防翻转能量在翻转出现时将会变为无穷大。这样优化问题的结果会保证属于可行域。比如之前的一些研究工作^[10-13],采用的方法中也使用一个无翻转的映射作为初始值^[12],如果存在翻转,则会将其映射到无翻转的可行域中。在使用 ADMM 算法优化的过程中,用矩阵 A_i 来表示三角曲面 τ_i 的映射矩阵,在 ADMM 算法的 A 子问题中,使用牛顿法来迭代 A ,在迭代中使用线搜索来保证每个三角曲面上的映射矩阵的行列式大于 0。这样保证了算法结果始终处于无翻转的可行域中。

1.2 ADMM 算法

ADMM 是一种强大的优化算法,因为其适合处理大型问题以及可以高度并行化,通过与 GPU 结合,在信号处理、计算机视觉、图像处理、自动控制等领域有着广泛的应用^[14-18]。ADMM 算法一开始用来处理凸问题,实际上,它在处理非凸问题上也有很好的表现^[19-21]。该算法把一个全局的优化问题分解为小的局部问题。通常处理以下形式的问题:

$$\begin{aligned} \min \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz = c \end{aligned} \quad (1)$$

就像 SA 算法,在分散网格后,必须保证网格可以拼接起来,以及如果想要处理网格变形,还得保证满足控制点约束。在本文中,通过引入辅助变量^[22],全局的优化问题被分解成了每个三角曲面上包含两个子问题的优化问题。这两个子问题分别对应着网格拼接,能量优化两个方面。

2 可并行参数化算法

2.1 问题描述

考虑 3D 网格 M 上的参数化映射 $\mathcal{M} = \{A_1, A_2,$

$\dots, A_N\}$ 。映射的原始网格是 3D 三角网格 M , 该三角网格包含 N_v 个顶点 $\{v_i, i=1, \dots, N_v\}$, N_e 条边 $\{e_i, i=1, \dots, N_e\}$, N 个三角面片 $\{s_i, i=1, \dots, N\}$, $\mathcal{E}_{\text{int}}$ 是包含三角网格 M 所有内边的集合。在每一个面片 s_i 上构建局部坐标系, 面片上的点在局部坐标系中表示为 $v=(x_1, x_2) \in \mathbb{R}^2$, 参数域上的点表示为 $u=(u_1, u_2) \in \mathbb{R}^2$ 。用 $\{A_i, T_i\}$ 来表示面片 s_i 上的仿射变换 $\mathcal{A}_i$, 其中 $A_i \in \mathbb{R}^{2 \times 2}$ 是线性变换矩阵, $T_i \in \mathbb{R}^2$ 是平移向量。面片 s_i 上的点 v 映射后的结果 u 为 $A_i v + T_i$ 。三角网格 M 上的映射是无翻转的当且只当 $\det(A_i) > 0, i=1, \dots, N$ 。使用 MIPS 能量来度量映射误差。对于每个三角形曲面, 映射被定义成了一个仿射变换:

$$\phi(x) = Ax + T \quad (2)$$

对于每个三角曲面, MIPS 能量有以下表示:

$$E(\phi) = \left(\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1} \right) = \frac{\text{trace}(A^T A)}{\det(A)} \quad (3)$$

其中 σ_1, σ_2 是线性变换矩阵 A 有符号的奇异值。优化问题被构造成为以下形式:

$$\begin{aligned} \min_{\{A_i, i \in [1, N]\}} & \sum_{i=1}^N E(A_i) \\ \text{s.t.} & \det(A_i) > 0, \forall i=1, \dots, N \\ & (A_{e_1} - A_{e_2})e = 0, \forall e \in \mathcal{E}_{\text{int}} \end{aligned} \quad (4)$$

这里的约束 $(A_{e_1} - A_{e_2})e = 0, \forall e \in \mathcal{E}_{\text{int}}$ 是任意两个共享同一条边 e 的三角曲面 s_{e_1}, s_{e_2} 上的变换矩阵的相容性条件, 其中 A_{e_1} 是 s_{e_1} 上的变换矩阵, A_{e_2} 是 s_{e_2} 上的变换矩阵。 e 是边 e 的单位向量, $\mathcal{E}_{\text{int}}$ 是包含三角网格 M 所有内边的集合。该约束要求三角网格任意内边经过相邻两个三角形上的变换矩阵映射后, 得到的两条边方向平行, 长度相等, 这样共享的边仍然能够无缝拼接起来。如图 1 中只有 (3) 中的映射才能拼接起来, (1) 与 (2) 中或者方向不平行或者长度不相等, 得到的参数化结果无法在参数域中拼接起来。

2.2 ADMM 算法

对于该优化问题使用 ADMM 算法来解决。首先,

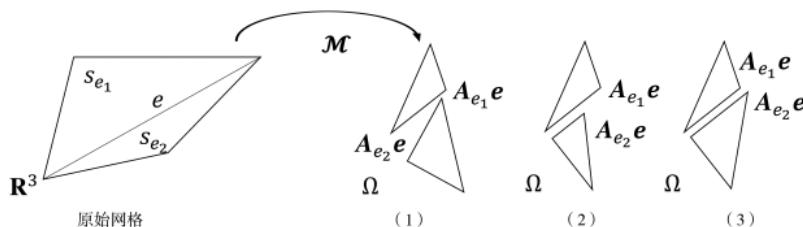


图 1 相容性约束示意图

把所有三角曲面上的变换矩阵 A_i 连接成一个矩阵 $A \in \mathbb{R}^{2 \times 2N}$ 。然后引入辅助变量: $Y \in \mathbb{R}^{2 \times 4|\mathcal{E}_{\text{int}}|}$, 它对应于相邻三角曲面上的线性映射的相容性约束 $(A_{e_1} - A_{e_2})e = 0, \forall e \in \mathcal{E}_{\text{int}}$ 。常见符号如表 1 所示。

表 1 常见符号表

符号	定义
A	$(A_0, A_1, \dots, A_N)$
Y	$(Y_{e_0,1}, Y_{e_0,2}, \dots, Y_{e_1,1}, Y_{e_1,2}, \dots, Y_{e_{ \mathcal{E}_{\text{int}} },1}, Y_{e_{ \mathcal{E}_{\text{int}} },2}), i \in [1, \mathcal{E}_{\text{int}}]$
S_Y	选择矩阵, 可以选择出 Y 的子矩阵所在三角曲面上的变换矩阵 A_i
λ_Y	对偶变量, 更新方式: $\lambda_Y = \lambda_Y + \mu(Y - AS_Y)$

可以把该问题重写为以下形式:

$$\begin{aligned} \min_{A, Y} & \sum_{i=1}^N E(A_i) + \sigma_1(Y) \\ \text{s.t.} & Y = AS_Y \end{aligned} \quad (5)$$

这里 σ_1 是相容性约束的示性函数。 $S_Y \in \mathbb{R}^{2N \times 4|\mathcal{E}_{\text{int}}|}$ 是稀疏的选择矩阵, 则有:

$$\begin{aligned} L(A, Y, \lambda_Y) = & \sum_{i=1}^N E(A_i) + \sigma_1(Y) + \lambda_Y \odot (Y - AS_Y) + \frac{\mu}{2} \|Y - AS_Y\|^2 \end{aligned} \quad (6)$$

其中 λ_Y 是对偶变量, $\odot$ 是系数乘法, μ 是惩罚系数。算法的主流程见图 2。

本算法的初始参数化映射 $\mathcal{M}_0$ 通过使用简单的参数化算法 Linear ABF^[12] 获得, 相对应的初始参数化网格为 $M_0 := \mathcal{M}_0(M)$ 。

把每个面片上的仿射变换作为优化变量, 构造优化问题, 引入辅助变量, 应用 ADMM 算法处理优化问题, 获取最终的参数化映射 $\mathcal{M}_{\text{final}}$ 。

等到算法收敛后, 从 $\mathcal{M}_{\text{final}}$ 中恢复出相对应的参数化网格 $\mathcal{M}_{\text{final}}$ 。

2.2.1 固定 A , 更新 Y

固定 A , 去掉无关的项, $\min_Y L(A, Y, \lambda_Y)$ 简化为:

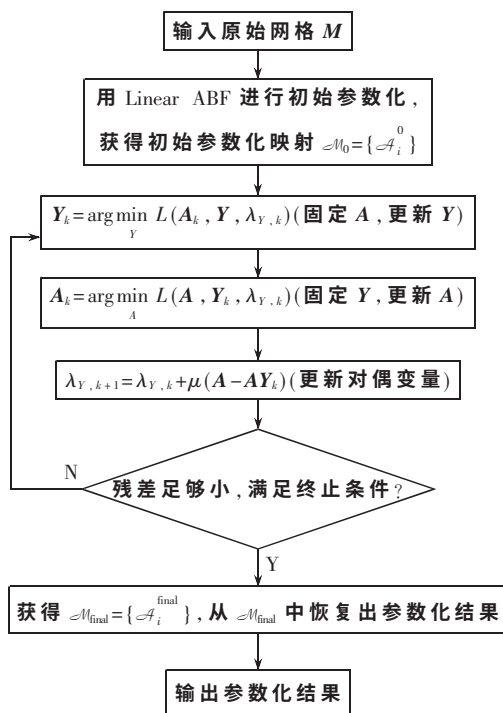


图2 算法流程图

$$\min_Y \sigma_1(Y) + \frac{\mu}{2} \left\| Y - AS_Y + \frac{\lambda_Y}{\mu} \right\|^2 \quad (7)$$

对于每个内边 e , 可以分解成 $|\mathcal{E}_{\text{int}}|$ 个独立的局部子问题:

$$\min_{Y_{e_1}, Y_{e_2}} \sigma_1(Y_{e_1}, Y_{e_2}) + \frac{\mu}{2} \left(\left\| Y_{e_1} - A_{e_1} + \frac{\lambda_Y^{e_1}}{\mu} \right\| + \left\| Y_{e_2} - A_{e_2} + \frac{\lambda_Y^{e_2}}{\mu} \right\| \right) \quad (8)$$

其中 σ_1 有以下形式:

$$\sigma_1(Y_{e_1}, Y_{e_2}) = \begin{cases} 0, & (Y_{e_1} - Y_{e_2})e = 0 \\ \infty, & \text{其他} \end{cases} \quad (9)$$

其中 e_1, e_2 是内边 e 相邻的两个三角曲面, 令 Y_{e_1}, Y_{e_2} 分别为对应于三角曲面 e_1, e_2 的用于解决相容性问题辅助矩阵, A_{e_1}, A_{e_2} 为上一轮更新的三角曲面 e_1, e_2 上的变换矩阵。而 $\lambda_Y^{e_1}, \lambda_Y^{e_2}$ 是上一轮更新的三角曲面 e_1, e_2 上的对偶变量。用拉格朗日乘子法来解决该子问题, 问题的解 Y 相对应于边 e 的组成部分是:

$$\begin{cases} Y_{e_1} = q_{e_1} - d_e \\ Y_{e_2} = q_{e_2} + d_e \end{cases} \quad (10)$$

这里:

$$d_e = \frac{1}{2} (q_{e_2} - q_{e_1}) e e^T \quad (11)$$

且:

$$\begin{cases} q_{e_1} = A_{e_1} - \frac{\lambda_Y^{e_1}}{\mu} \\ q_{e_2} = A_{e_2} - \frac{\lambda_Y^{e_2}}{\mu} \end{cases} \quad (12)$$

2.2.2 固定 Y , 更新 A

固定 Y , $\min_A L(A, Y, \lambda_Y)$ 分解为 N 个独立的局部子问题。对于每一个三角曲面 i :

$$\min_{A_i} E(A_i) + \frac{\mu}{2} \left(\sum_{t \in y(i)} \left\| A_i - \left(Y_t + \frac{\lambda_Y^t}{\mu} \right) \right\|^2 \right) \quad (13)$$

其中 A_i 表示每个三角曲面上的变换矩阵。 $y(i)$ 分别是辅助矩阵 Y 中对应于三角曲面 i 的矩阵的索引的集合。 λ_Y^t 是上轮更新的相对应的对偶变量。 B_{Y_t} 是矩阵 $Y_t + \frac{\lambda_Y^t}{\mu}$ 的元素列成一列。使用 $f(A_i)$ 来表示目标函数, 对于 MIPS 能量, 目标函数的梯度和黑塞矩阵可以写成以下形式:

$$\begin{aligned} \nabla f &= \frac{2V_1}{\det(A_i)} - \frac{\text{sign}(\det(A_i)) \text{trace}(A_i^T A_i) V_2}{\det(A_i)^2} + \\ &\mu \left[\sum_{t \in y(i)} (V_1 - B_{Y_t}) \right] \end{aligned} \quad (14)$$

$$\begin{aligned} H &= \frac{2I}{|\det(A_i)|} - \frac{2\text{sign}(\det(A_i))(V_1 V_2^T + V_2 V_1^T)}{\det(A_i)^2} - \\ &\frac{\text{sign}(\det(A_i)) \text{trace}(A_i^T A_i) I'}{\det(A_i)^2} + \\ &\frac{2\text{sign}(\det(A_i)) \text{trace}(A_i^T A_i) V_2 V_2^T}{\det(A_i)^2} + \beta I \end{aligned} \quad (15)$$

其中 $\beta = \mu(|y(i)|)$, $V_1 = (a, b, c, d)^T$ 是将 A_i 的元素列成一列, $V_2 = (d, -c, -b, a)^T$, sign 是符号函数, I 是一个 4×4 的单位矩阵, I' 有以下表示形式:

$$I' = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (16)$$

使用牛顿法来解决这个问题, 更新流程如下, 对于三角曲面 i 上的变换矩阵有:

$$\begin{cases} \delta A_i^{k+1} = -H_i^{k-1} \cdot \nabla f(A_i^k) \\ A_i^{k+1} = A_i^k + \alpha_k \cdot \delta A_i^{k+1} \end{cases} \quad (17)$$

令 $\alpha_0 = 1$, α_k 用 α_{k-1} 初始化, 如果 α_k 使目标式 $f_i(A_i)$

上升,则乘以 0.5 来更新步长,直到目标式下降。为了确保无翻转,在线搜索的过程中必须保证 $\det(A_i) > 0$,在线搜索中目标式 $f_s(A_i)$ 在 $\det(A_i) < 0$ 时设置无穷大, $\det(A_i) > 0$ 时 $f_s(A_i) = f(A_i)$ 。

2.2.3 更新对偶变量

用该公式更新 λ_Y :

$$\lambda_{Y, k+1} = \lambda_{Y, k} + \mu(Y - AS_Y) \quad (18)$$

2.2.4 初始化与终止条件

上述算法需要给出每个三角曲面上的初始变换矩阵。使用一些参数化方法来计算原始网格的参数化结果来得到初始的参数化结果,本次实验中主要采用的方法是 Linear ABF 参数化方法,除此以外还可以使用 TUTTE 嵌入。认定当 $\|r_{\text{primal}}\|^2$ 小于 0.000 3 且 $\|r_{\text{primal}}\|^2$ 小于 $1e^{-5}$ 的时候,ADMM 算法收敛。

主残差:

$$r_{\text{primal}} = [Y - AS_Y] \quad (19)$$

对偶残差:

$$r_{\text{primal}} = \mu \begin{bmatrix} \delta AS_Y \\ \delta A \end{bmatrix} \quad (20)$$

2.2.5 网格恢复

在优化算法收敛后,将得到每个三角曲面上的变换矩阵 A_i ,而网格顶点在参数域上的坐标可以从变换矩阵中恢复出来。为进行这一系列过程,从顶点 v_0 出发,进行广度优先遍历构造以下点集:

$$\begin{aligned} D_0 &= \{v_0\} \\ D_1 &= N(D_0) \setminus D_0 \\ D_2 &= N(D_1) \setminus (D_0 \cup D_1) \\ D_3 &= N(D_2) \setminus (D_0 \cup D_1 \cup D_2) \\ &\dots \\ D_i &= N(D_{i-1}) \setminus \bigcup_{k=0}^{i-1} D_k \end{aligned} \quad (21)$$

其中 $N(\cdot)$ 是一环相邻顶点的集合, D_0 表示包含顶点 v_0 的集合, D_i 表示距离 D_0 中任意一个顶点最短路径正好包含 $i-1$ 条边的顶点的集合。所有的集合都根据从 D_0 中顶点出发的广度优先搜索来决定。构造过程为从 D_0 中取出点 v_0 作为根节点,找到它未遍历过的相邻点,放入 D_1 中,成为 v_0 的子节点,然后依次取出 D_1 中的顶点,遍历未访问的顶点,构造 D_2 ,这样依次构造点集,直到网格中所有的顶点都被遍历过。

依次计算 $D_0, D_1 \dots$ 中顶点的在参数域中的位置,当已经计算出来 D_i 中顶点 v_a 在参数域中的位

置时 $u(v_a)$,依照以下公式恢复出 D_{i+1} 中 v_a 的子节点——顶点 v_b 在参数域中的位置 $u(v_b)$:

$$u(v_b) = u(v_a) + A^*(v_b - v_a) \quad (22)$$

其中 A^* 是连接两个顶点 v_a, v_b 的边任一相邻三角面片上的变换矩阵。当 D_i 中所有顶点都被计算出参数域中位置之后,再依次计算 D_{i+1} 中顶点在参数域中的位置。

3 实验结果

网格参数化上的实验环境是一台 2.50 GHz 的 Intel Core i5 的笔记本电脑, RAM 为 8 GB, GPU 是 GeForce 940MX。利用 CUDA 加速时,使用的是 cuda 9.0。核函数中每个块中有 64 个线程数,块的个数则由该子问题总共有多少并行计算的元素来决定的,例如在 A 子问题中,三角曲面的个数 N 将分散到 $\lfloor N/64 \rfloor$ 个块上。牛顿法中,线搜索中 $\alpha=0.5$,同时当梯度的范数 ∇f 小于 $1e^{-3}$ 时,终止牛顿法。为了更加公正地对比不同方法得到的映射的质量,使用文献[24]中的方法来统一质量度量。对于每一个三角曲面 s_j ,采用保角损失度量 $\delta_{s_j}^{\text{con}} = \max\{\sigma_{j,1}, \sigma_{j,2}\} / \min\{\sigma_{j,1}, \sigma_{j,2}\}$,其中 $\sigma_{j,1}, \sigma_{j,2}$ 是三角曲面 j 上的映射 A_j 的奇异值。用 $\sigma_{\text{avg}}^{\text{con}}$ 来表示所有三角曲面上的保角损失度量的平均值。

3.1 网格参数化

图 3 展示了三角网格参数化结果。图中原始网格之后紧跟的是该网格的参数化结果,第一个是 SA 算法的结果,第二个是本文算法的结果,使用的初始值都是由 Linear ABF 算法得到的。在参数化结果下面标注着算法计算时间以及得到的参数化结果的平均保角度损失度量。可以看出,本文的算法在 bunny 与 bimba 这两个例子上展现了极大的优势,这是因为这两个网格相比于 male, disk 是大型网格(bunny 有 1 251 046 个顶点, bimba 有 30 268 个顶点),可以看到本文的算法因为高度并行化的缘故,长于处理大型网格,在计算速度上占有极大的优势。同时,将本文算法同 SLIM[23]、AMIPS[24]算法进行比较,结果展示在表 2 中,可以看出本文算法得到高质量网格的同时,运算时间基本缩短了至少百倍以上,比如 tooth, SLIM 算法所用时间是本文算法的 23 726 倍。图 4 展示了网格大小递进的 gargoye 模型 MIPS 能量下降趋势,证明本文算法的可扩展性,大多数算法在模型点数超过 150k 时,收敛

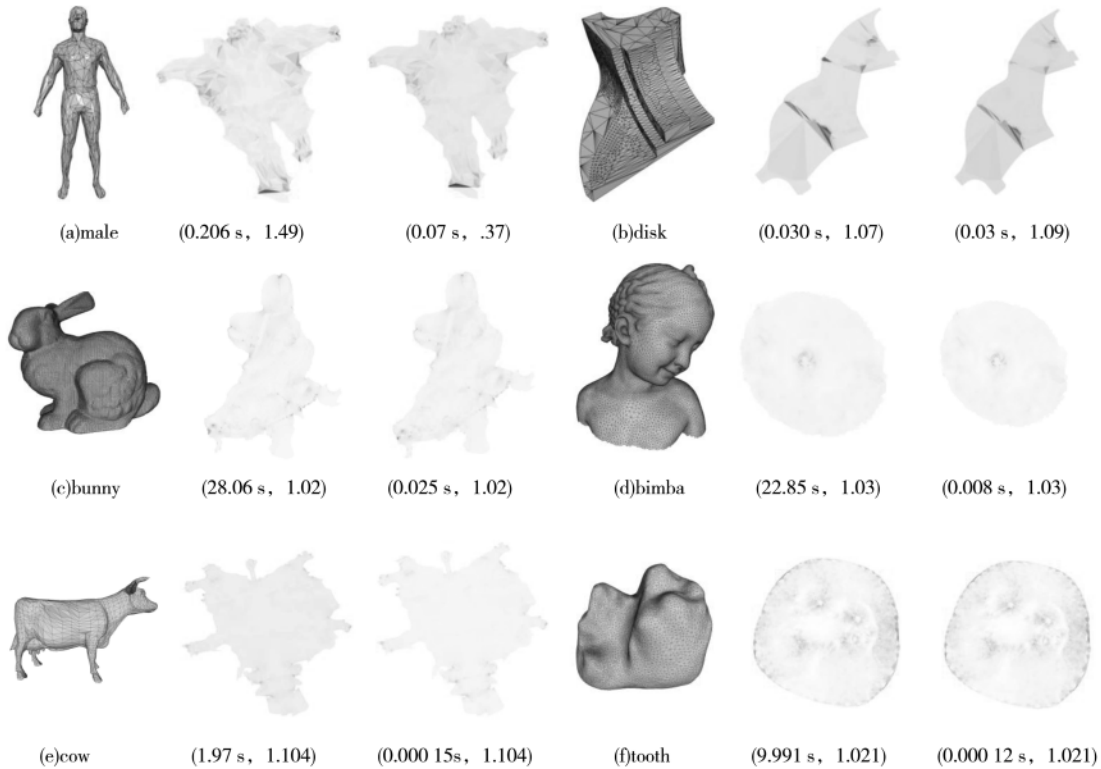


图 3 网格参数化结果展示

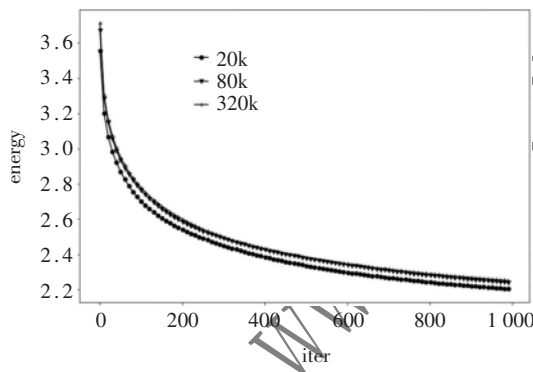


图 4 不同大小网格迭代趋势图

会变慢,本文算法却在网格模型变大为 320k 个面时,保持着近乎一致的收敛速度。

3.2 不同惩罚系数

图 5 为不同惩罚系数下网格参数化的结果。对于 camel,在惩罚系数为 90 000 时的最终收敛结果的平均保角度损失度量为 1.32,收敛时间为 0.025 s,在惩罚系数为 30 000 时的最终收敛结果平均保角度损失度量为 1.25,收敛时间为 0.064 s;对于 isis,在惩罚系数为 50 000 时的最终平均保角度损失度量为 1.118,收敛时间为 0.003 5 s,在惩罚系数为 10 000 时的最终平均保角度损失度量为 1.099,收敛时间

表 2 参数化算法运行时间比较

网格	算法	N_e/N	δ^{con}	T/s
cow	SA	3 195/5 804	1.104	1.97
	AMIPS	3 195/5 804	1.098	1.055
	SLIM	3 195/5 804	1.102	0.529
	ADMM	3 195/5 804	1.104	0.000 145 6
bunnyhead	SA	2 929/5 792	1.119	5.205
	AMIPS	2 929/5 792	1.116	1.059
	SLIM	2 929/5 792	1.119	0.692
	ADMM	2 929/5 792	1.105	0.000 334 1
bone	SA	2 529/5 000	1.087	4.253
	AMIPS	2 529/5 000	1.085	0.958
	SLIM	2 529/5 000	1.087	0.644
	ADMM	2 529/5 000	1.086	0.000 153 1
tooth	SA	8 670/17 263	1.021	9.991
	AMIPS	8 670/17 263	1.021	3.475
	SLIM	8 670/17 263	1.021	2.757
	ADMM	8 670/17 263	1.021	0.000 116 2
bunny	SA	35 190/69 451	1.020	28.06
	AMIPS	35 190/69 451	1.019	12.702
	SLIM	35 190/69 451	1.020	17.438
	ADMM	35 190/69 451	1.020	0.025
bimba	SA	30 268/60 290	1.030	22.85
	AMIPS	30 268/60 290	1.033	9.472
	SLIM	30 268/60 290	1.034	17.998
	ADMM	30 268/60 290	1.030	0.008

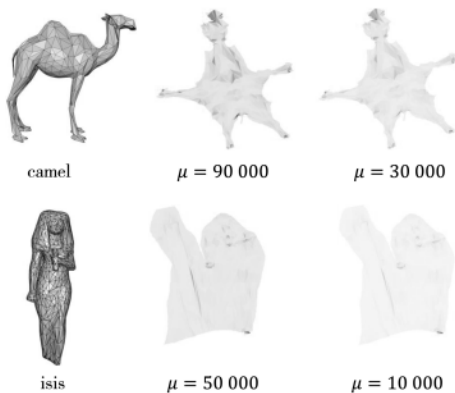


图 5 不同惩罚系数参数化结果图

为 0.058 s。注意当惩罚系数过低的时候,将出现不收敛的情况。但参考表 3,惩罚系数整体来说有相当大的范围都会收敛,比如对 tooth 进行保角映射,在惩罚系数 1 000 到 50 000 都是收敛的。对于 MIPS 能量,通常可以选择 10 000。对于对称狄利克雷能量,惩罚系数通常可以选择 500。

表 3 不同惩罚系数比较

网格	惩罚系数	N_v/N	T/s	δ^{con}
cow	10 000	3 195/5 804	0.000 145 6	1.104 18
	3 000	3 195/5 804	0.001 441 2	1.103 798
	50 000	3 195/5 804	0.000 15	1.104 23
bunnyhead	50 000	2 929/5 792	0.000 334 1	1.105 279
	10 000	2 929/5 792	0.002 754 4	1.108 617
	90 000	2 929/5 792	0.000 204 9	1.100 203
bone	10 000	2 529/5 000	0.000 153 1	1.086 329
	50 000	2 529/5 000	0.000 279 4	1.086 738
	2 000	2 529/5 000	0.000 678 5	1.086 3
tooth	10 000	8 670/17 263	0.000 116 2	1.021 082
	90 000	8 670/17 263	0.000 179 2	1.021 086
	1 000	8 670/17 263	0.000 126 1	1.021 032

4 结论

随着参数化领域研究的发展,各种参数化算法被陆续提出。但快速产出无翻转、高质量的参数化映射一直是一个强烈需求。在本文中,提出一种针对三角网格,计算三角网格的无翻转、高质量参数化结果的可并行、可扩展方法。该网格参数化 ADMM 算法可以高效并行地计算无翻转、高质量的三角网格映射,在处理大型网格上尤其占有优势。本文从不同角度、不同例子展示了该算法的优势。在处理大型网格或者希望快速处理网格时,都可以从该

算法中得到益处。

参考文献

- [1] BOYD S, PARIKH N, CHU E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations and Trends in Machine learning, 2011, 3(1): 1-122.
- [2] FLOATER M. One-to-one piecewise linear mappings over triangulations[J/OL]. Mathematics of Computation, 2002, 72.
- [3] SHEFFER A, DE STURLER E. Parameterization of faceted surfaces for meshing using angle-based flattening[J/OL]. Engineering with Computers, 2001, 17: 326-337.
- [4] FU X M, LIU Y. Computing inversion-free mappings by simplex assembly[J]. ACM Transactions on Graphics(TOG), 2016, 35(6): 216.
- [5] SHEFFER A, LEVY B, MOGILNITSKY M, et al. ABF++: fast and robust angle based flattening[J/OL]. ACM Trans. Graph., 2005, 24: 311-330.
- [6] KOVALSKY S Z, AIGERMAN N, BASRI R, et al. Large-scale bounded distortion mappings[J]. ACM Trans. Graph., 2015, 34(6): 1-10.
- [7] HORMANN K, GREINER G. MIPS: an efficient global parametrization method[R]. Erlangen-Nuernberg UNIV (Germany) Computer Graphics Group, 2000.
- [8] LIPMAN Y. Bounded distortion mapping spaces for triangular meshes[J/OL]. ACM Transactions on Graphics-TOG, 2012, 31.
- [9] AIGERMAN N, LIPMAN Y. Injective and bounded distortion mappings in 3d[J/OL]. ACM Transactions on Graphics(TOG), 2013, 32.
- [10] SMITH J, SCHAEFER S. Bijective parameterization with free boundaries[J]. ACM Transactions on Graphics(TOG), 2015, 34(4): 70.
- [11] SCHÜLLER C, KAVAN L, PANOZZO D, et al. Locally injective mappings[C]. Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing. Eurographics Association, 2013.
- [12] ZAYER R, LÉVY B, SEIDEL H P. Linear angle based parameterization[C]. Barcelona: Fifth SGP, 2007.
- [13] RABINOVICH M, PORANNE R, PANOZZO D, et al. Scalable locally injective mappings[J]. ACM Transactions on Graphics, 2017, 36(2): 1-16.
- [14] TAO J, ZHANG J, DENG B, et al. Parallel and scalable

- heat methods for geodesic distance computation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.
- [15] ZHANG J, PENG Y, OUYANG W, et al. Accelerating admm for efficient simulation and optimization[J]. ACM Transactions on Graphics(TOG), 2019, 38(6): 1–21.
- [16] NEUMANN T, VARANASI K, WENGER S, et al. Sparse localized deformation components[J]. ACM Transactions on Graphics(TOG), 2013, 32(6): 179.
- [17] PAN Z, MANOCHA D. Efficient solver for spacetime control of smoke[J]. ACM Transactions on Graphics(TOG), 2017, 36(5): 162.
- [18] OVERBY M, BROWN G E, LI J, et al. Admm $\supseteq$ projective dynamics: fast simulation hyperelastic models with dynamic constraints[J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23(10): 2222–2234.
- [19] WEN Z, YANG C, LIU X, et al. Alternating direction methods for classical and ptychographic phase retrieval[J]. Inverse Problems, 2012, 28(11): 115010.
- [20] MIKSIK O, VINEET V, P? REZ P, et al. Distributed non-convex admm-inference in largescale random fields[C]. British Machine Vision Conference(BMVC), 2014.
- [21] LIAVAS A P, SIDIROPOULOS N D. Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers[J]. IEEE Transactions on Signal Processing, 2015, 63(20): 5450–5463.
- [22] COMBETTES P L, PESQUET J C. Proximal splitting methods in signal processing[J]. Hnz H Bauschke, 2011, 49: 185–212.
- [23] RABINOVICH M, PORANNE R, PANOZZO D, et al. Scalable locally injective mappings[J]. ACM Transactions on Graphics, 2017, 36(2): 1–16.
- [24] FU X M, LIU Y, GUO B. Computing locally injective mappings by advanced MIPS[J]. ACM Transactions on Graphics(TOG), 2015, 34(4): 71.

(收稿日期: 2020-06-28)

作者简介:

吴璇(1994–), 通信作者, 女, 硕士, 主要研究方向: 计算机图形学、优化算法。E-mail: bx315@mail.ustc.edu.cn。

张举勇(1985–), 男, 博士, 副教授, 主要研究方向: 三维视觉、计算机图形学、数值最优化。

(上接第 15 页)

作者简介:

杜兵(1988–), 通信作者, 男, 硕士研究生, 工程师, 主要研究方向: 计算机应用。E-mail: sdnydubing@163.

com。

杨仁宝(1975–), 男, 硕士研究生, 工程师, 主要研究方向: 航天测控。

李林峰(1989–), 男, 硕士研究生, 工程师, 主要研究方向: 计算机应用。

版权声明

经作者授权，本论文版权和信息网络传播权归属于《信息技术与网络安全》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《信息技术与网络安全》编辑部
中国电子信息产业集团有限公司第六研究所