基于 B 样条的细分曲面参数化

王智巍, 陈仁杰*

(中国科学技术大学数学科学学院 合肥 230022)
(renjiec@ustc.edu.cn)

摘 要: 四边形细分曲面参数化算法在游戏影视领域具有广泛应用, 但现有算法无法兼顾参数化速度和质量. 为此, 提出一种基于 B 样条的细分曲面参数化算法, 首先将细分曲面转化为分片 B 样条曲面, 然后通过最优化几何扭曲对样条曲面逐片进行参数化; 对于一些细分曲面分片较多的情况, 文中通过拼接相邻面片, 对合并曲面进行参数化. 不同于定义在分片线性空间上的网格类参数化算法, 文中以数量更少的控制网格为优化变量, 得到的定义在样条空间上的参数化映射在分片内部具有 C^2 连续性. 与现有算法相比, 收敛速度更快, 参数化结果的扭曲更低, 采取的合并策略能有效地减少割缝数量, 提升参数化结果的质量.

关键词: 细分曲面; B 样条函数; 网格参数化; 计算机图形学; 四边形网格
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2024.20186

B-Splines Based Subdivision Surfaces Parameterization

Wang Zhiwei and Chen Renjie*

(*School of Mathematical Sciences, University of Science and Technology of China, Hefei* 230022)

Abstract: The quadrilateral subdivision surface parameterization algorithm is widely used in the gaming and film industry. However, current algorithms do not effectively balance the speed and quality of parameterization. To tackle this problem, a B-spline-based subdivision surface parameterization algorithm is proposed. In this algorithm, the input subdivision surface is first transformed into a piecewise B-spline surface. Then, the spline surface is parameterized section by section using optimized geometric distortion. In cases where the subdivision surface has numerous subdivisions, the algorithm utilizes patch concatenation to parameterize the merged surface. Unlike grid-based parameterization algorithms that are defined in the piecewise linear space, the proposed algorithm reduces the number of control grids used as optimization variables. This approach results in a parameterization mapping defined in the spline space, ensuring C^2 continuity within each patch. Compared to existing algorithms, the algorithm demonstrates faster convergence, lower overall distortion in parameterization results, effective reduction of seam count through merging strategies, and improved quality of the parameterization outcomes.

Key words: subdivision surfaces; B-splines; mesh parameterization; computer graphics; quadrilateral mesh

细分曲面具有良好的几何性质, 可以表示复杂的几何形状, 在电影、动画、计算机辅助设计等行业具有广泛应用. 对输入网格无限次重复细分操作如 Catmull-Clark(CC)细分算法[1], 得到的极限曲面即为细分曲面. 细分曲面的优点是能表示任意拓扑的光滑曲面, 并可以通过直接操纵其控制

收稿日期: 2023-03-23; 修回日期: 2023-06-20. **基金项目**: 国家自然科学基金(62072422); 安徽省自然科学基金(2008085MF195).
王智巍(1998—), 男, 硕士研究生, 主要研究方向为数字几何处理; **陈仁杰**(1983—), 男, 博士生导师, CCF 会员, 论文通信作者, 主要研究方向为几何建模与处理、计算机图形学.

网格来改变曲面的形状. 除了在奇异点附近, 细分曲面整体是 C^2 连续的, 并可以用分片双三次 B 样条曲面表示.

网格参数化是对网格几何、拓扑等信息进行进一步处理的基础, 在图形渲染和数字几何处理等领域具有广泛的应用. 在实际应用中, 参数化需满足低扭曲、无翻转等条件. 例如, 纹理映射将平面上的图案贴到 3D 物体的表面上, 更小的扭曲意味着更小的失真; 而翻转会导致参数化网格的局部折叠, 进而图案在贴到 3D 物体上时会发生局部断裂或者没有图案对应的现象.

之前有关细分曲面参数化的工作[2]流程如下.

输入. 细分曲面的控制网格.

输出. 细分曲面的参数化结果.

Step1. 对输入网格 CC 细分一定次数并三角化, 得到低次细分的控制网格.

Step2. 对低次细分的控制网格进行 As-rigid-as-possible(ARAP)参数化[3], 得到映射结果.

Step3. 对低次细分的控制网格和映射结果同时细分多次并三角化, 得到高次细分的控制网格的参数化结果, 以此来近似细分曲面的参数化结果.

这种做法[2]存在 2 个问题: (1) 需要多次细分输入网格, 才能减少高次细分的控制网格参数化结果的扭曲, 但这会增加低次细分控制网格的规模, 影响参数化速度, 而较少的细分次数又会导致高次细分控制网格参数化结果的扭曲变大. (2) 细分曲面是对控制网格无限次重复细分操作的结果, 以高次细分的控制网格近似细分曲面在理论上会有逼近误差.

为了解决上述问题, 本文提出了基于 B 样条的细分曲面参数化算法, 对于输入的细分曲面, 计算对应的分片 B 样条曲面和各片对应的 B 样条控制网格, 然后对样条曲面逐片进行曲面参数化, 得到细分曲面的参数化映射. 针对一些细分曲面分片较多的情况, 本文设计了合并曲面参数化算法来减少割缝, 提升参数化质量.

与之前的算法[2]相比, 本文将优化变量从低次细分的控制网格替换为变量数更少的 B 样条曲面控制网格, 提升了优化速度; 将优化空间从分片线性空间替换为样条空间并进行连续参数化映射, 然后对连续映射采样, 此时仍能较好地保持三角面片之间的连续性, 能够更好地降低参数化结果的扭曲. 此外, 与逐片曲面参数化相比, 合并曲面参数化虽然以增加最终扭曲为代价, 并且参数化映射在拼接处只有 C^0 连续, 但能够有效地减少割缝数量, 提升参数化结果的质量.

1 相关工作

网格参数化的目标是得到输入网格与平面之间的映射. 除了要求尽可能小的扭曲, 参数化还需要满足内部无翻转和边界无相交的约束. 这样在纹理映射时, 才能得到相对理想的结果, 图 1 展示了 ARAP 参数化算法[3]对 cow 模型的参数化结果. 左图为棋盘格纹理下的模型贴图, 右图为参数化结果的扭曲, 越接近红色扭曲越大, 越接近白色扭曲越小.

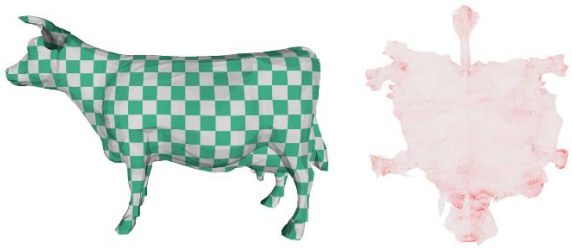


图 1 ARAP 参数化算法[3]的参数化结果

在参数化的研究中, 为了减小参数化扭曲, 学者提出了很多优化能量, 保角扭曲度量如保角能量[4], MIPS 能量[5], As-similar-as-possible 能量[3,6]; 等距扭曲度量如 Symmetric dirichlet 能量[7-8], AMIPS 能量[9], ARAP 能量[3,6]. 绝大多数参数化算法需要一个无翻转的初始参数化网格, Tutte 参数化算法[10]能提供一个固定边界的无翻转的初始化, 这为很多参数化算法[8,11-13]的成功奠定了基础. 对于线性能量[3,6], 一般采用最小二乘法将优化问题转化为求解线性方程组; 对于非线性能量, 优化算法分为 3 类: 只用一阶梯度矩阵而不直接用二阶海森矩阵的一阶算法, 如文献[11,14-16]; 使用一阶梯度矩阵的差异去迭代近似二阶 Hessian 矩阵的柯西牛顿法, 如 L-BFGS[17-18]; 使用二阶 Hessian 矩阵信息的拟牛顿法, 如文献[13,19]. 在 3 类算法中, 一阶算法每步的迭代速度快但收敛所需迭代次数最多, 拟牛顿法可以用最少的迭代次数收敛, 但每一步获得二阶 Hessian 矩阵的代价是非常高昂的.

针对四边形细分曲面的参数化工作较少, He 等[2]提出了 2 种算法, 一种是控制网格参数化算法, 以低次细分的控制为优化变量, 计算其在 3D 空间和 UV 参数域之间的参数化映射, 然后对映射细分多次得到高次细分的控制的参数化结果, 以此来逼近细分曲面的参数化结果. 另一种算法是基函数参数化算法, 通过控制网格的基函数计算参数化映射, 得到光滑的细分曲面参数化映射. 相较于第 1 种算法, 基函数参数化算法得到映射有更好的

连续性质, 但收敛速度较慢. Gong 等[20]针对同胚于圆盘和球体的四边形网格, 设计了以角度扭曲为优化目标的纯四边形网格的参数化算法, 但是这种算法仅在简单拓扑上有较好表现, 在亏格更高的四边形网格上有很大局限性. He 等[21]针对细分曲面奇异点处的参数化结果有较大扭曲的现象, 设计了重新参数化算法来降低奇异点附近的参数化扭曲.

本文关注的是细分曲面参数化的快速算法, 在此基础上保证参数化结果有较低的扭曲. 虽然根据定义, CC 细分曲面是由初始四边形网格经无限次 CC 细分而得到, 但是 Patching Catmull-Clark meshes(PCCM)算法[22]证明, 除了在少量奇异点处, 细分曲面可以分片转化为 B 样条张量积曲面. 因此本文以 B 样条曲面的控制网格为变量, 直接对分片 B 样条曲面进行参数化.

2　本文算法

本文算法主要流程如下.

输入. 细分曲面的控制网格.

输出. 细分曲面的参数化结果.

Step1. 通过网格分片算法[23]得到分片网格, 逐片使用PCCM算法[22]得到分片双三次B样条曲面, 以此表示对应分片在 CC 细分下的细分曲面.

Step2. 对每片 B 样条曲面, 先在其控制网格的参数域上粗采样并三角化得到粗采样网格; 然后以控制网格为变量, 对粗采样网格逐片参数化得到每片细分曲面在 3D 空间和 UV 域之间的保距映射关系[3].

Step3. 指定细采样分辨率, 对 3D 空间和 UV 域上的 B 样条曲面同时采样并三角化, 得到细分曲面在所求分辨率下的参数化结果.

针对一些细分曲面分片较多的情况, 由于相邻 B 样条曲面的控制网格在公共边界处重合, 因此本文拼接相邻曲面来减少分片数量, 然后对合并曲面参数化, 提升参数化质量.

2.1　细分曲面到分片双三次 B 样条曲面

PCCM 算法[22]将细分曲面转化为分片的双三次 B 样条面片, 其中, 双三次 B 样条面片整体是 C^2 连续的, 奇异点处是 C^0 连续, 并可以通过平滑奇异点的方式使奇异点处 G^1 连续, 但平滑后得到的双三次 B 样条曲面表示和细分曲面在奇异点处存在近似误差. 除此之外, PCCM 算法[22]要求分片满足 2 个条件: (1) 分片的形状与矩形类似; (2) 分片的内部点的度均为 4, 且只有片的 4 个角点可以是奇异点.

本文使用分片算法[23], 它能得到满足条件的网格分片, 并且是片数较少的算法. 图 2 展示了分片算法在一些模型上的分片效果, 其中红色连线标注了分片边界. 该算法计算所有从细分曲面奇异点出发的粒子轨迹, 如果粒子相撞, 则按照相应规则停止或继续前进, 最终实现网格分片. 得到的片均为四边形片, 内部点的度为 4, 边界点的度为 3, 角点的度为 2 且仅有角点可能是分片网格的奇异点. 由此得到的网格分片的数目和细分曲面奇异点的个数有关. 在输入的细分曲面奇异点较多的情况下, 最终结果的纹理贴图会有较多割缝.

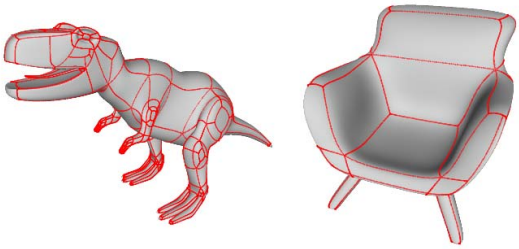


图 2　分片算法[23]在一些模型上的分片效果

需要注意的是, 网格分片算法允许一些片的角点落在另一些片的边上, 即出现 T 型拼接的情况. 然而, PCCM 算法无法保证 T 型拼接处分片曲面之间的 G^1 连续性, 此时分片样条曲面和细分曲面之间存在近似误差. 针对这种现象, 可行的解决方法是修改分片规则, 让粒子撞到分片边界时继续前行, 避免出现 T 型拼接, 但这会大幅增加分片数量, 削弱了网格分片算法分片较少的优势. 因此在实践中, 本文容许近似误差的存在.

2.2　细分曲面参数化

本文将参数化的优化空间设置在样条空间, 以分片 B 样条曲面的控制网格作为优化变量, 进行细分曲面参数化.

首先在样条空间上对 B 样条曲面进行均匀采样, 然后对采样得到的点集三角化, 得到细分曲面的粗采样网格 $M(\boldsymbol{V},\boldsymbol{T})$. 其中, $\boldsymbol{V}$ 是 M 所有顶点的集合, $\boldsymbol{T}$ 是 M 所有面的集合. 值得注意的是, $\boldsymbol{V}$ 可以由 B 样条控制网格的控制顶点 $\boldsymbol{V}_{\mathrm{cp}}$ 线性表出, 即 $\boldsymbol{V}=\boldsymbol{G}\boldsymbol{V}_{\mathrm{cp}}$. 其中, $\boldsymbol{V}\in\mathbb{R}^{n\times 3}$; $\boldsymbol{V}_{\mathrm{cp}}\in\mathbb{R}^{m\times 3}$; n 是 M 的顶点数; m 是 B 样条控制网格的控制顶点数; $\boldsymbol{G}\in\mathbb{R}^{n\times m}$ 是在 B 样条空间上均匀采样时唯一确定的系数矩阵.

对 M 中的三角面片 t, 设其顶点在局部坐标系下的坐标为 $x_t=\{x_t^0,x_t^1,x_t^2\}$, 并且对应的参数化坐标为 $u_t=\{u_t^0,u_t^1,u_t^2\}$, 从而有 x_t 与 u_t 间的雅可比

矩阵 $\boldsymbol{J}_t(u)$，以及 x_t 到 u_t 的单位旋转矩阵 $\boldsymbol{L}_t(u)$. 进而优化能量为 $E(u,\boldsymbol{L}_t)=\sum_{t=1}^{\#T}A_t\left\|\boldsymbol{J}_t(u)-\boldsymbol{L}_t(u)\right\|_{\mathrm{F}}^2$. 其中，$\#T$ 表示网格 M 的面数；A_t 表示三角面片 t 的面积.

本文使用 ARAP 参数化算法[3]对这一优化问题进行求解. 在求解的局部阶段，首先对三角面片 t 对应的雅可比矩阵 $\boldsymbol{J}_t$ 进行 SVD 分解，即 $\boldsymbol{J}_t=\boldsymbol{U}_t\boldsymbol{\Sigma}_t\boldsymbol{V}_t$；得到单位旋转矩阵 $\boldsymbol{L}_t=\boldsymbol{U}_t\boldsymbol{V}_t^{\mathrm{T}}$. 在求解的全局阶段，采取和 ARAP 算法[3]中相同的变换算法，对能量 $E(u,\boldsymbol{L}_t)$ 变形可得

$$E(u,\boldsymbol{L}_t)=\frac{1}{2}\sum_{(i,j)\in e_{\mathrm{h}}}\cot(\theta_{ij})\left\|(u_i-u_j)-\boldsymbol{L}_{t(i,j)}(x_i-x_j)\right\|^2.$$

其中，e_{h} 是 M 所有半边的集合；(i,j) 表示 M 中从 v_i 指向 v_j 的半边；$t(i,j)$ 是包含半边 (i,j) 的三角面片；θ_{ij} 是 $t(i,j)$ 中半边 (i,j) 的对角.

设 $\boldsymbol{U}=(u_i)_{n\times1}$，即粗采样网格参数化结果对应的列向量，可将 $E(u,\boldsymbol{L}_t)$ 化为以 $\boldsymbol{U}$ 为变量的形式

$$\min_{\boldsymbol{U}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{U}-\boldsymbol{b}\boldsymbol{U}+c \tag{1}$$

这里 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，$\boldsymbol{b}\in\mathbb{R}^{n\times1}$，$c\in\mathbb{R}$. 对应表达式如下.

对顶点 v_i 的一邻域集合 $N(i)$，

$$\boldsymbol{A}_{ii}=\sum_{j\in N(i)}\left(\cot(\theta_{ij})+\cot(\theta_{ji})\right).$$

对于 v_i 邻域中的顶点 v_j，

$$\boldsymbol{A}_{ij}=-\left(\cot(\theta_{ij})+\cot(\theta_{ji})\right),$$

其余位置为 0;

$$\boldsymbol{b}_i=\sum_{j\in N(i)}\left(\cot(\theta_{ij})\boldsymbol{L}_{t(i,j)}+\cot(\theta_{ji})\boldsymbol{L}_{t(j,i)}\right)(x_i-x_j).$$

c 的值在后续计算中没有用到，故不展开.

为了使细分曲面的粗采样网格参数化结果 $\boldsymbol{U}$ 可由参数域上细分曲面的 B 样条控制网格 $\boldsymbol{U}_{\mathrm{cp}}$ 线性表出，令 $\boldsymbol{U}=\boldsymbol{G}\boldsymbol{U}_{\mathrm{cp}}$，从而式(1)可以化为 $\min_{\boldsymbol{U}_{\mathrm{cp}}}(\boldsymbol{G}\boldsymbol{U}_{\mathrm{cp}})^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{G}\boldsymbol{U}_{\mathrm{cp}})-\boldsymbol{b}(\boldsymbol{G}\boldsymbol{U}_{\mathrm{cp}})+c$. 通过最小二乘法转为矩阵方程 $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{G}\boldsymbol{U}_{\mathrm{cp}}=\boldsymbol{G}^{\mathrm{T}}\boldsymbol{b}$.

与 He 等[2]在分片线性空间优化不同，本文在样条空间优化，将优化变量从低次细分控制网格替换为数量更少的 B 样条曲面控制网格，提升了算法的求解速度.

2.3 合并曲面参数化

一些例子中，细分曲面有较多奇异点，导致分片算法[23]分片较多，这会使参数化结果的割缝较多，纹理不连贯，降低了参数化质量. 为此，本文提出了合并曲面参数化算法.

本文注意到，相邻的分片 B 样条曲面的控制网格在公共边界处是重合的，因此本文将相邻曲面拼接，并在第 2.2 节的基础上对合并曲面进行参数化. 设 $\{M_i\}_{i\in I}$ 为分片 B 样条曲面对应的粗采样网格块集合，I 为细分曲面分片对应标号的集合. 该集合满足如下性质：若 M_i 和 M_j 相邻，则它们有重合的边界. 本文要求 I 中至少有一组相邻的分片曲面. 设 $M_i(\boldsymbol{V}_i,\boldsymbol{F}_i)$ 对应的控制顶点为 $\boldsymbol{V}_{\mathrm{cp}_i}$，则有对应关系 $\boldsymbol{V}_i=\boldsymbol{G}_i\boldsymbol{V}_{\mathrm{cp}_i}$，$i\in I$. 其中，$n_i$ 是 M_i 的顶点数；m_i 是 M_i 对应的 B 样条控制网格的控制顶点数；$\boldsymbol{V}_i\in\mathbb{R}^{n_i\times3}$，$\boldsymbol{V}_{\mathrm{cp}_i}\in\mathbb{R}^{m_i\times3}$，$\boldsymbol{G}_i\in\mathbb{R}^{n_i\times m_i}$ 是在 B 样条空间上均匀采样时唯一确定的；相应有参数域上的粗采样网格参数化结果 $\boldsymbol{U}_i$ 和对应的 B 样条控制网格 $\boldsymbol{U}_{\mathrm{cp}_i}$.

设 $M_I(V_I,F_I)$ 是对 $\{M_i\}_{i\in I}$ 网格拼接得到的合并网格. 拼接时将 $\{M_i\}_{i\in I}$ 中相邻网格片的重合边和位置重合的顶点合并. M_I 的顶点数小于 $\{M_i\}_{i\in I}$ 的顶点数之和，面数等于 $\{M_i\}_{i\in I}$ 的面数之和. 与之对应有参数域上 M_I 的合并网格参数化结果 $\tilde{\boldsymbol{U}}$ 和对应的合并 B 样条控制网格 $\tilde{\boldsymbol{U}}_{\mathrm{cp}}$，这里 $\tilde{\boldsymbol{U}}$ 和 $\tilde{\boldsymbol{U}}_{\mathrm{cp}}$ 也是将 $\{\boldsymbol{U}_i\}_{i\in I}$ 和 $\{\boldsymbol{U}_{\mathrm{cp}_i}\}_{i\in I}$ 中相邻分片的重合边拼接所得. 其中，对每一个 $\boldsymbol{U}_{\mathrm{cp}_i}$，存在唯一的矩阵 $\boldsymbol{S}_i$ 满足

$$\boldsymbol{U}_{\mathrm{cp}_i}=\boldsymbol{S}_i\tilde{\boldsymbol{U}}_{\mathrm{cp}} \tag{2}$$

其中，$\boldsymbol{S}_i$ 是位置矩阵，表示 M_i 的每个顶点在合并网格 M_I 顶点中的对应位置.

对合并网格 M_I 的优化能量，可以视为 $\{M_i\}_{i\in I}$ 中每片的优化能量的累和

$$\min_{\{\boldsymbol{U}_{\mathrm{cp}_i}\}_{i\in I}}\sum_{i\in I}(\boldsymbol{G}_i\boldsymbol{U}_{\mathrm{cp}_i})^{\mathrm{T}}\boldsymbol{A}_i(\boldsymbol{G}_i\boldsymbol{U}_{\mathrm{cp}_i})-\boldsymbol{b}_i(\boldsymbol{G}_i\boldsymbol{U}_{\mathrm{cp}_i})+c_i;$$

将式(2)代入可得

$$\min_{\tilde{\boldsymbol{U}}_{\mathrm{cp}}}\sum_{i\in I}(\boldsymbol{G}_i\boldsymbol{S}_i\tilde{\boldsymbol{U}}_{\mathrm{cp}})^{\mathrm{T}}\boldsymbol{A}_i(\boldsymbol{G}_i\boldsymbol{S}_i\tilde{\boldsymbol{U}}_{\mathrm{cp}})-\boldsymbol{b}_i(\boldsymbol{G}_i\boldsymbol{S}_i\tilde{\boldsymbol{U}}_{\mathrm{cp}})+c_i.$$

通过最小二乘法转化为矩阵方程

$$\left(\sum_{i\in I}(\boldsymbol{G}_i\boldsymbol{S}_i)^{\mathrm{T}}\boldsymbol{A}_i(\boldsymbol{G}_i\boldsymbol{S}_i)\right)\tilde{\boldsymbol{U}}_{\mathrm{cp}}=\sum_{i\in I}(\boldsymbol{G}_i\boldsymbol{S}_i)^{\mathrm{T}}\boldsymbol{b}_i \tag{3}$$

求解合并曲面参数化映射时，局部阶段对每个三角面片计算 $\boldsymbol{L}_t$，全局阶段求解式(3)中的矩阵方程.

3　实　验

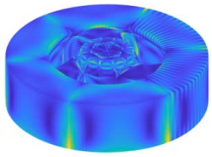
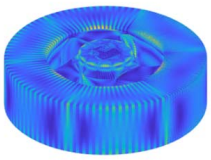
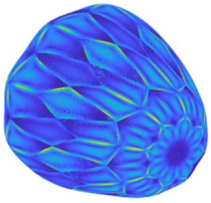
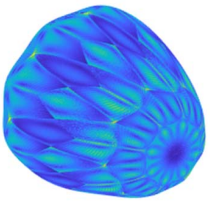
实验使用的 CPU 为 i9-10900KF，内存容量为 64 GB. 本文关注的是参数化的效率，选取 He 等[2]的控制网格参数化算法进行对比，分别比较在每片细分曲面上参数化的速度和最终扭曲，分析合并曲面参数化算法的缝合效果，并讨论合并分片的数量选取对最终结果质量的影响. 最后展示了本文算法在一些例子上的纹理贴图结果，并讨论了算法存在的不足.

3.1　参数化结果扭曲的比较

为确保本文和 He 等[2]得到的最终网格的规模相同，在细采样时，本文对 B 样条控制网格每个单位内 4 等分采样，然后三角化得到最终网格. 与之对应 He 等[2]在细采样时，对分片线性映射做 2 次 CC 细分得到最终网格. 这样处理能确保 2 种算法得到的最终网格的顶点数和面数相同，从而确保对比实验的公平性.

本文计算了最终结果的扭曲，即所有三角面片的 ARAP 能量[3]之和，和 He 等[2]算法进行比较. 表 1 展示了 2 种算法参数化结果扭曲的对比，数字为相应算法的扭曲. 分片的颜色越接近黄色，扭曲越大，越接近深蓝，扭曲越小. 可以看出，本文算法的参数化结果扭曲更小，这是因为 He 等[2]算法在分片线性空间进行参数化，并对分片线性映射进行细分，面片间的扭曲随着细分变大，最终结果扭曲更大. 而本文基于 B 样条空间进行连续的参数化映射，对连续映射采样并三角化，能较好地保持三角面片之间的连续性，结果有更低的扭曲.

表 1　2 种算法对模型进行曲面参数化得到的最终结果的 ARAP[3]能量和对比

模型	本文		He 等[2]	
	参数化结果	ARAP[3]能量和	参数化结果	ARAP[3]能量和
模型 1		69.150 7		77.973 8
模型 2		48.575 7		56.120 4
模型 3		5.710 4		6.329 3

3.2　参数化速度的比较

本文设置 ARAP 算法[3]的收敛条件为

$$\left|\frac{E_{k+1}-E_k}{E_k}\right|<10^{-6}.$$

其中，E_k 是第 k 次迭代时的扭曲. 首先比较本文算法和 He 等[2]的算法在分片网格上收敛时的参数化结果和收敛时的扭曲. 从表 2 可以看出，本文算法收敛时的扭曲会更大，这是因为本文在自由度更小的 B 样条空间进行参数化，自由度降低限制了 ARAP 算法[3]在全局阶段找到更精确的局部最优解；而 He 等[2]算法在分片线性空间进行参数化，更大的自由度有助于找到更精确的局部最优解. 此外，比较扭曲的数量级可以看出本文算法和 He 等[2]算法的参数化扭曲在同一数量级，2 个扭曲的差值相对扭曲而言较小，说明自由度降低并没有导致扭曲大幅增加.

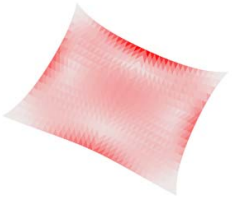
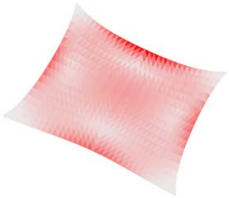
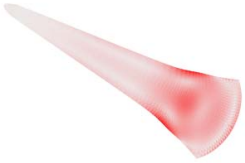
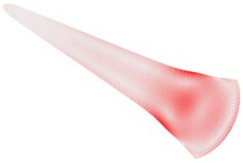
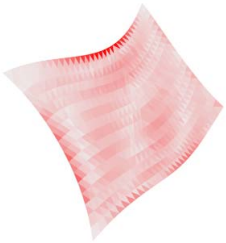
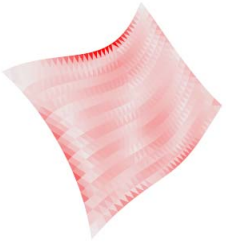
表 3 统计了本文算法和 He 等[2]算法达到收敛条件时所需的时间，可以看到本文算法的收敛速度更快，且随着细分曲面的采样网格规模变大，参数化的速度优势会更明显. 这是因为分片线性空间的变量维数随着细分指数增加，而样条空间的变量维数没有变化，从而本文求解的线性系统的复杂度会显著小于 He 等[2]算法，收敛用时会显著降低.

3.3 合并曲面参数化效果分析

图 3 展示了算法对细分曲面参数化结果的影响，红色连线标注分片曲面边界. 第 1 行是对分片曲面逐片参数化的结果，第 2 行是合并曲面参数化的结果. 合并后曲面内的纹理更连贯，说明合并算法能有效地减少割缝数量，提升参数化质量.

合并分片的数量由输入网格的片数决定. 合并数过小，不能有效地减少割缝数量，反之则会导致最终结果出现纹理扭曲过大的区域. 表 4 中，当合并数过大时，虽然能合并区域，但红圈标示的合并区域内部有明显的扭曲，这是因为更多的面片拼接，虽然减少了求解的变量数量，但更少的自由度也会限制求解式(3)时找到更精确的局部最优解. 所以合并数的大小和参数化结果的质量之间存在着平衡. 从经验来看，对一般的输入网格，合并片数选取输入网格分片数的 $\frac{1}{4}\sim\frac{1}{6}$ 时能得到质量较好且扭曲增长较少的结果.

表 2　2 种算法在分片曲面上参数化收敛时的 ARAP[3]能量和对比

模型	本文		He 等[2]	
	参数化结果	ARAP[3]能量和	参数化结果	ARAP[3]能量和
分片 1		$2.755\ 1\times10^{-6}$		$2.746\ 8\times10^{-6}$
分片 2		$8.932\ 6\times10^{-5}$		$8.931\ 0\times10^{-5}$
分片 3		$6.996\ 4\times10^{-4}$		$6.995\ 4\times10^{-4}$
分片 4		$2.539\ 5\times10^{-6}$		$2.521\ 7\times10^{-6}$

3.4 结果展示

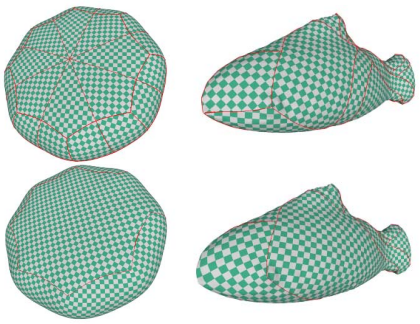
图 4 展示了合并曲面参数化算法在一些模型上的结果. 可以看到参数化结果的割缝较少，纹理区域连贯，这证明了合并策略的有效性. 同时注意到，一些模型上出现了较为明显的扭曲，比如图 4 中第 1 个模型的下方有明显的失真，这说明合并分片会一定程度增加扭曲.

3.5 算法的不足

本文用 B 样条曲面去表示分片细分曲面，通过求解 B 样条曲面的参数化映射来得到细分曲面的参数化映射，同 He 等[2]算法相比，有更高的效率，但分片算法[23]分块较多，会让参数化结果的割缝较多，纹理不连贯. 为此，本文通过合并相邻曲面来减少参数化时的非光滑过渡瑕疵. 其代价是，合并曲面在拼接处只有 C^0 连续，这要比 PCCM 算法[22]分片曲面之间的 G^1 连续性差；同时，相应的参数化映射在拼接处也是 C^0 连续的，这也比 He 等[2]算法的参数化映射的连续性差.

表 3　2 种算法参数化收敛时所需用时对比

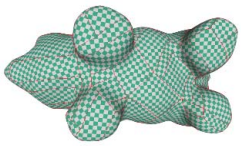
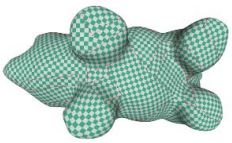
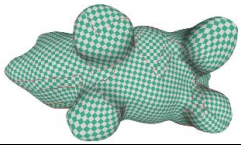
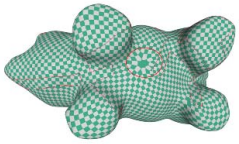
模型	顶点数	面数	控制点数	t/s		He 等[2]/本文
				He 等[2]	本文	
表 1 模型 1	160 531	292 352	78 869	52.798	16.399	3.219
	613 067	1 169 408		180.800	52.876	3.419
	2 395 195	4 677 632		1 183.241	252.853	4.679
表 1 模型 2	36 540	66 048	18 396	3.183	1.264	2.518
	139 086	264 192		11.782	4.358	2.703
	542 268	1 056 768		86.074	16.862	5.104
表 1 模型 3	27 744	49 152	16 224	4.606	1.794	2.567
	104 544	196 608		18.106	5.345	3.387
	405 600	786 432		84.587	22.036	3.838
图 3 模型 2	31 178	58 880	11 712	7.680	2.299	3.340
	121 210	235 520		32.082	6.890	4.656
	477 914	942 080		148.190	27.501	5.388
表 4 模型	66 359	124 160	26 439	7.961	5.063	1.572
	256 799	496 640		50.771	14.674	3.459
	1 010 159	1 986 560		359.633	64.533	5.572



a. 模型 1　　b. 模型 2

图 3　合并算法对细分曲面参数化结果质量的影响

表 4　对于分片总数为 79 片的模型，合并片数的选取对合并曲面参数化结果质量的影响

合并片数	合并曲面参数化结果	ARAP[3]能量和
0		32.529
10		34.482
20		38.021
30		41.299

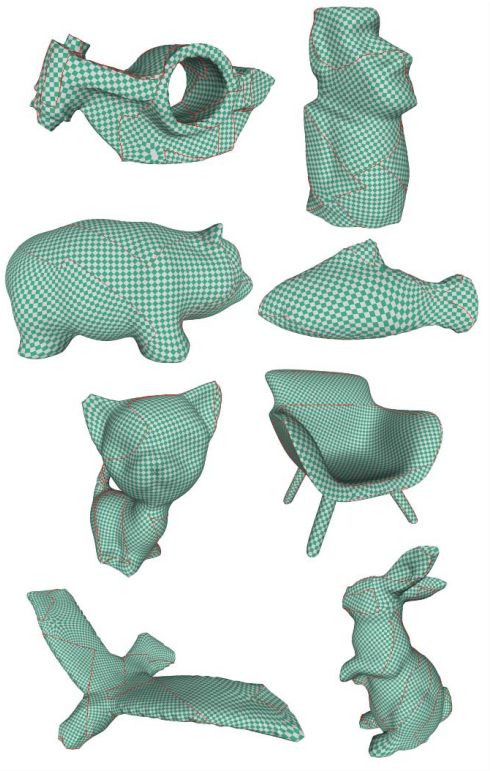


图 4　一些模型上的参数化结果展示

其次，本文使用 PCCM 算法[22]得到细分曲面的双三次 B 样条曲面表示，对网格分片有严格的要求，而 He 等[2]算法可以使用更自由的网格划分算法，允许分片内部出现奇异点，分片数量会远小于本文算法，同时参数化映射的连续性也好于本文算法．最后，使用分片算法[23]会在分片时出现 T 型拼接，本文算法得到的双三次 B 样条曲面表示和细分曲面在 T 型拼接处存在近似误差．

4 结 语

4.1 结 论

本文提出了基于 B 样条的细分曲面参数化算法，和之前工作[2]相比，本文算法不仅收敛速度更快，整体扭曲也更小. 本文使用B样条的思路还可以用于其他参数化算法，如基于 Symmetric Dirichlet 能量、AMIPS 能量的局部单射或全局双射参数化[7-9]，在这些算法的优化环节进行变量代换，减少求解的线性系统的复杂度，提高相应算法的求解效率. 这是因为本文算法选择了基于 B 样条的细分曲面表达，从而在参数化时大幅减少了变量数目，并且基于样条空间的参数化映射在分片内是 C^2 连续的，因此参数化结果的扭曲更低. 在实验部分，通过大量实例，验证了本文算法的上述优势. 此外，针对一些细分曲面分片较多的情况，本文提出了合并曲面参数化算法来减少分片数量，提升参数化质量. 实验证明了本文算法有良好的缝合效果，对最终结果的质量有显著提升.

4.2 未来工作

本文使用合并曲面参数化算法，有效地减少了割缝数量，但拼接的曲面越多，扭曲越大. 因此，如何在不显著增加扭曲的前提下更多地减少割缝数量，或者更进一步达到无缝的效果，将是下一步的研究重点.

此外，参数化映射的连续性在纹理贴图中非常重要，而本文得到的合并曲面参数化映射在拼接处只有 C^0 连续. 如何获得具有更好连续性质的细分曲面参数化映射，需要进一步研究.

参考文献(References):

[1] Catmull E, Clark J. Recursively generated B-spline surfaces on arbitrary topological meshes[M] //Wolfe R. Seminal Graphics: Pioneering Efforts that Shaped the Field. New York: ACM Press, 1998: 183-188

[2] He L, Schaefer S, Hormann K. Parameterizing subdivision surfaces[M] //Proceedings of the ACM SIGGRAPH 2010 Papers. New York: ACM Press, 2010: Article No.120

[3] Liu L G, Zhang L, Xu Y, *et al*. A local/global approach to mesh parameterization[J]. Computer Graphics Forum, 2008, 27(5): 1495-1504

[4] Degener P, Meseth J, Klein R. An adaptable surface parameterization method[C] //Proceedings of the 12th International Meshing Roundtable. Santa Fe: IMR, 2003: 201-213

[5] Hormann K, Greiner G. MIPS: an efficient global parametrization method[R]. Nashville: Vanderbilt University Press, 2000

[6] Lévy B, Petitjean S, Ray N, *et al*. Least squares conformal maps for automatic texture atlas generation[J]. ACM Transactions on Graphics, 2002, 21(3): 362-371

[7] Schreiner J, Asirvatham A, Praun E, *et al*. Inter-surface mapping[J]. ACM Transactions on Graphics, 2004, 23(3): 870-877

[8] Smith J, Schaefer S. Bijective parameterization with free boundaries[J]. ACM Transactions on Graphics, 2015, 34(4): Article No.70

[9] Fu X M, Liu Y, Guo B N. Computing locally injective mappings by advanced MIPS[J]. ACM Transactions on Graphics, 2015, 34(4): Article No.71

[10] Tutte W T. How to draw a graph[J]. Proceedings of the London Mathematical Society, 1963, 3(1): 743-767

[11] Claici S, Bessmeltsev M, Schaefer S, *et al*. Isometry-aware preconditioning for mesh parameterization[J]. Computer Graphics Forum, 2017, 36(5): 37-47

[12] Liu L G, Ye C Y, Ni R Q, *et al*. Progressive parameterizations[J]. ACM Transactions on Graphics, 2018, 37(4): Article No.41

[13] Shtengel A, Poranne R, Sorkine-Hornung O, *et al*. Geometric optimization via composite majorization[J]. ACM Transactions on Graphics, 2017, 36(4): Article No.38

[14] Poranne R, Lipman Y. Provably good planar mappings[J]. ACM Transactions on Graphics, 2014, 33(4): Article No.76

[15] Kovalsky S Z, Galun M, Lipman Y. Accelerated quadratic proxy for geometric optimization[J]. ACM Transactions on Graphics, 2016, 35(4): Article No.134

[16] Rabinovich M, Poranne R, Panozzo D, *et al*. Scalable locally injective mappings[J]. ACM Transactions on Graphics, 2017, 36(2): Article No.16

[17] Jiang L J, Byrd R H, Eskow E, *et al*. A preconditioned L-BFGS algorithm with application to molecular energy minimization[R]. Colorado Univ At Boulder Dept of Computer Science, 2004

[18] Liu T T, Bouaziz S, Kavan L. Quasi-Newton methods for real-time simulation of hyperelastic materials[J]. ACM Transactions on Graphics, 2017, 36(3): Article No.23

[19] Teran J, Sifakis E, Irving G, *et al*. Robust quasistatic finite elements and flesh simulation[C] //Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. New York: ACM Press, 2005: 181-190

[20] Gong W Y, Xie X H, Ma R, *et al*. Angle-preserving quadrilateral mesh parameterization[J]. IEEE Computer Graphics and Applications, 2015, 35(6): 51-59

[21] He L, Loop C, Schaefer S. Improving the parameterization of approximate subdivision surfaces[J]. Computer Graphics Forum, 2012, 31(7): 2127-2134

[22] Peters J. Patching Catmull-Clark meshes[C] //Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 2000: 255-258

[23] Eppstein D, Goodrich M T, Kim E, *et al*. Motorcycle graphs: canonical quad mesh partitioning[J]. Computer Graphics Forum, 2008, 27(5): 1477-1486