

基于谱分解的三角网格面排序及条纹纹理贴图

李 岩¹ 周世哲² 邓建松¹

(1. 中国科学技术大学数学科学学院 安徽合肥 230026; 2. 湖南大学信息科学与工程学院 湖南长沙 410082)

摘要: 现有网格文件格式中三角面布局混乱, 对大型网格后续处理带来不便。利用输入网格的对偶网格的谱分解, 可以得到一个良好的网格三角面排序, 但是排序的结果是偏序的。针对该算法的不足, 通过一种独特的遍历算法, 得到了一个三角面布局是全序的三角网格。并且进一步的, 得到了一系列有序的三角带, 其中输入的原始网格的几何和拓扑性质不发生改变。最后对三角带参数化, 实现条纹纹理贴图。实验表明: 经过排序后, 网格的三角面布局相关性相比于流网格得到进一步的加强, 而且网格有一个很好的三角带表示。

关键词: 谱分解; 全序; 三角带; 条纹纹理贴图

中图分类号: TP391.41 文献标识码: A doi: 10.3969/j.issn.0253-2778.2017.09.002

2010 Mathematics Subject Classification: 65Y04

引用格式: 李岩, 周世哲, 邓建松. 基于谱分解的三角网格面排序及条纹纹理贴图 [J]. 中国科学技术大学学报, 2017 A7(9): 721-728.

LI Yan, ZHOU Shizhe, DENG Jiansong. Face sorting and stripe texture mapping of triangle mesh based on spectral decomposition [J]. Journal of University of Science and Technology of China, 2017 A7(9): 721-728.

Face sorting and stripe texture mapping of triangle mesh based on spectral decomposition

LI Yan¹, ZHOU Shizhe², DENG Jiansong¹

(1. School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, China;

2. College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China)

Abstract: The layout of the triangle face in the existing mesh file format is often out of order, which brings great difficulties in the subsequent processing of large meshes. A useful approach later proposed based on spectral decomposition of dual meshes, did lead to a good face order. However, the order was only partial. In view of the drawback of the algorithm, a unique traversal algorithm was presented here which can get a triangular mesh with a total ordered layout of the faces. Furthermore, a series of ordered triangle strips were obtained, and the geometric and topological properties of the original input mesh didn't change in the process. Finally, the triangle strips were parameterized, and the stripe texture mapping was applied. Experimental results show that the method is superior to the spectral decomposition method and can generate a satisfactory triangular strip.

Key words: spectral decomposition; total order; triangle strip; stripe texture mapping

收稿日期: 2016-10-11; 修回日期: 2017-05-22

基金项目: 国家自然科学基金(61303147), 中央高校基本科研费(湖南大学青年教师成长计划 2015 和 2016 批次)资助。

作者简介: 李岩, 女, 1990 年生, 硕士。研究方向: 计算机图形学。E-mail: ly92915@mail.ustc.edu.cn

通讯作者: 周世哲, 博士/副教授。E-mail: shizhe@hnu.edu.cn

0 引言

在计算机图形学中，三角网格是表示三维物体的一种非常流行的形状表现方式。三角网格是一系列顶点、边、面的集合。随着科学技术的发展，网格模型日益复杂。但是由于计算机内存的限制，大型网格数据已经不能完全加载在计算机主内存中。*out-of-core* 思想是先将网格剖分，然后逐段的将部分网格加载在计算机内存中。*out-of-core* 思想已经被广泛地应用到大型网格模型的可视化、网格简化、网格压缩、网格重建等领域中，该算法的前提是网格三角面片之间有一个良好的相关性。

但是如今大部分网格还是沿用传统的文件格式（PLY、OBJ、IV、OFF、VRML 等）存储网格。例如 PLY 文件格式，仅由顶点坐标列表和面的顶点引用列表组成，没有考虑网格顶点、面之间的连贯性，这样往往导致网格的布局相关性很差。

此外，三角带作为网格的一种表示形式。一方面，在没有改变原始网格的条件下，它可以作为网格三角面片的一种排序，使得网格三角面片的布局相关性增强。另一方面，它在许多领域内有着重要的应用。例如能够加快 CPU 与 GPU 之间的数据传输、加速渲染、进行网格压缩、实现条纹纹理贴图等。

为了解决原始网格模型三角面布局混乱问题，Matrin 等^[1] 通过谱分解算法得到了一个流网格，但是得到的网格模型中三角面片排序是一种偏序的形式。本文中，我们通过改进其算法得到一个三角面片排序是全序的网格（与原始网格模型相比只是三角面的顺序发生改变）。同时为了得到一个有效的网格三角带表示，我们在全序网格上通过深度优先算法得到一系列三角带。因为该三角带的生长方向与得到的网格三角面片排序方向密切相关，所以此三角带是一种有序的三角带。这里我们不改变输入原始网格顶点、面信息，只是网格顶点、面的输出顺序发生了改变。最后作为一个应用，我们对得到的一系列有序三角带进行参数化，得到一个条纹纹理贴图。

1 相关工作

1.1 网格排序工作的简述

Wu 等^[2] 采用 *out-of-core* 思想，将大型网格的三角面片分批逐段载入到一个固定容量的缓冲器

中，在缓冲器上执行边折叠操作（edge collapse）来渐进地简化网格。但是该算法的前提是网格三角面片已经有了一个很好的排序（三角面片之间有良好的相关性）。针对 Wu 方法中必须有的前提，Isenburg 等^[3] 提出了一种网格的压缩流表示，该表示提供了一个顶点何时结束引用的详细信息，解决了网格预排序的问题。当输入一个三角面片时，只需查找引用该三角面片上的顶点的其他三角面片，将它们一起输入到缓冲器上即可。

大部分的图形显卡都是使用顶点缓存器缓存少数顶点。为了充分利用这些缓存器，对于后续待处理的三角面，能够重新引用缓冲器上已有的顶点是非常必要的。Dering^[4] 提出了一种几何数据压缩表示，它在存储三角面片时，提供了在缓冲器上哪个顶点将要被代替的详细信息。Hoppe 等^[5] 对于一个特定的顶点缓存器通过贪婪的三角带增长和局部最优的排序方法给出了一个最优的网格三角面片排序算法。Bogomjakov 等^[6] 则创建了一个对任意缓冲器都适用的网格三角面布局相关性良好的排序方法。文献[7]讨论了在图像和几何处理过程中，针对不同缓冲器的几种网格三角面片排序算法。

我们的网格排序主要是基于 Matrin 等^[1] 的工作，Matrin 等是通过谱方法创建了一个流网格，但是得到的流网格中三角面片的排序是偏序的，我们基于 Matrin 的工作得到了一个全序的网格，并且进一步得到一系列有序的三角带。

对网格模型进行三角带表示的研究一直在两个方面进行。其一，不改变输入的原始网格，找到一条或尽可能少的三角带条数。其二，忽略输入的原始网格，通过增加新的网格顶点或减少原来的网格顶点进行重新三角化，来得到一条完整的三角带。在不改变输入网格的拓扑性质条件下，网格的三角带表示也是对网格三角面片进行排序，但是本文中，三角带的相关工作作为单独的一节进行阐释。

1.2 网格三角带工作的简述

寻找网格的三角带也即是寻找其对偶网格的哈密顿回路，在超过 100 个面的凸多边形网格上寻找一个哈密顿圈，所需要的时间用现有的已知指数时间算法是无法计算的^[8]。因此，许多寻找三角带的算法工作都尝试从输入的原始三角网格中去寻找一些可能的三角带。SGI^[9] 公司开发了一个程序，在网格内使用启发式算法产生三角带。该算法是通过在创建三角带时，选择一邻域面数和最小的三角面作为三

角带开始和结束的三角面。并且在选择三角带要相连的下一个三角面时，也选择一邻域面数和最小的三角面，该算法的目标是减少孤立三角面片的数量。

经典的 STRIPE 算法^[10]对于输入的原始多边形网格在局部上采用 SGI 的启发式算法，得到一个链式数据结构，该数据结构存储网格每个三角面的一邻域面的个数。然后进行全局性分析，在一个很大的“块状”区域上寻找一条最长的三角带。Chow^[11]则通过尽可能重复利用先前三角带中的顶点来增加三角带的长度。

Snoeyink 等^[12]对于不规则的三角网格，使用一种特殊的遍历方法来生成它的对偶网格生成树。该生成树并不是必要的，只是在寻找三角带的工作中遍历的贪婪生成，最后将很多条短的三角带连接生成的三角带。Xiang 等^[13]提出了一种有效且鲁棒的算法把一个网格模型分解成很多条三角带。该算法使用深度优先或者宽度优先的搜索方法生成网格对偶图的生成树，然后将生成树分解成多条三角带，最后使用一种聚合方法，将短的三角带聚集为长的三角带。

进一步，Shafae 等^[14]提出了一种基于视觉的网格简化和渲染工作中的动态、实时的三角带管理方法和数据结构：DStrips。该算法通过边折叠和顶点分裂实时的改变网格，由于网格改变导致相应的三角带发生改变，每一帧三角带的变化被保存在 Dstrip 结构中。Demaine 等^[15]放宽了三角带的定义，允许相邻的三角面共享一个顶点而不是一条边。Porcu 等^[16]提出了 ETA 三角带算法，在其对偶网格上执行一个简单的拓扑操作，该算法依赖于管道长度和生成三角带的预算时间。

秦爱红等^[17]提出了一种基于混合模式缓存优化的三角形带生成算法，该算法基于贪心算法生成三角带，每步依据传输代价函数来选择下一个将被加入到三角带的候选三角面。当函数判断出如果继续沿原方向增长三角带将会导致缓存命中率下降时，将转变三角带增长方向来选择候选三角面。张洁等^[18]通过定义一个 Conners 数据结构，在编码三角面时生成三角带，但是会有大量游离点存在。

魏潇然等^[19]提出一种无损拓扑压缩算法。该算法首先将三角网格模型剖切成图，然后将剖切图表示成三角节点树，二叉树中每个节点记录三条边的拓扑信息。最后根据其索引在二叉树中搜索其两端相邻节点，通过把两端相邻节点加入到三角带中来

获取一条长三角带。

在上述提到的寻找三角带算法中，输入的原始网格的几何和拓扑性质是没有被修改的。

其他一些工作都是通过重新三角化网格来寻找一条单一的三角带表示。例如，在文献[20]中算法对于一个平面点集不做 Delaunay 三角剖分而是进行一个哈密顿路径剖分。此外，文献[20]把输入的平面点集仅仅作为一个简单的平面多边形，检查在多边形内部是否存在单条三角剖分。如果没有，使用 Steiner 顶点三角化。该文献同时证明了对于一个有洞的平面多边形，计算它的哈密顿路径是一个 NP 问题。

QuadTIN^[21]方法通过在四叉树上增加 Steiner 点三角化一个不规则的数据集，产生一个动态的与视觉相关的三角化，并且进一步得到了单一的三角带。给定一个四边形流行网格，Taubin^[22]将每个四边形分裂成三角形，并且重新排序使其存在单一的三角带。

Gopi 等^[23]对一个任意流形网格运用对偶网格匹配算法将原始网格分成一个个闭合的圈，然后通过引进新的顶点来得到单一的三角带。基于文献[23]的算法基础，Diaz 等^[24]通过增加一个完美的加权匹配方法得到了一个良好的三角带，实现了一个交互式的高性能渲染功能。该算法可以根据用户定义的与功能相关的约束条件来产生单一的三角带。

Gurung 等^[25]提出了一种表示三角网格拓扑信息的网格拓扑结 LR，其中是通过一种贪婪的算法来获得网格的近似哈密顿回路。

上述方法都对原始网格进行了重新三角化，改变了原始网格顶点、边和面的信息。

我们对于一个任意流形三角网格，在没有改变原始输入网格的情况下，得到了一系列有序的三角带，它在可视化、纹理映射、压缩等方面都有很大的应用。在这里我们主要讨论在纹理映射方面的应用。

纹理映射一直是图形学方向研究的经典问题，有很多的经典方法。例如，Floater^[26]提出的保形映射。我们这里参照 Floater 的方法对每一条三角带进行参数化。

2 本文算法

2.1 流网格

文献[1]提出了一种谱分解算法对网格三角面片进行排序，排序后的网格被称为流网格。给定一

个封闭的三角网格 $M = \{f_1, f_2, \dots, f_N\}$, 其中 f_i 是网格的三角面片, N 是网格三角面片的个数. 网格 M 的对偶网格 $\bar{M} = (\{v_1, v_2, \dots, v_N\}, E)$ 的邻接矩阵 $L(l_{ij})$ 定义如下. 对 L 做谱分解, 求出它的第二小特征值对应的特征向量 v . 根据 v 中元素值从小到大的顺序, 对应对原始网格三角面进行重新排序, 得到一个新的流网格 $SM = \{f'_1, f'_2, \dots, f'_N\}$ (假设三角面片 f_j 在原始网格 M 的索引值为 t , 则有 $v[t]$ 是 v 中第 j 小值). SM 与 M 相比, 只是三角面片的顺序发生了改变.

$$l_{ij} = \begin{cases} \deg(v_i) = 3, i = j; \\ -1 & (v_i, v_j) \in E; \\ 0 & (v_i, v_j) \notin E \end{cases} \quad (1)$$

文献 [1] 创建的流网格的三角面片是沿着网格主方向(垂直于网格的最大边界)螺旋向上排序的,但是在主方向的每个截面上,三角面片的排序是混乱的,所以得到的网格三角面片排序是偏序的,如图 1 所示,黄色的盘绕曲线表示网格三角面片的一种螺旋排序方式. 我们在文献 [1] 工作的基础上,在主方向的每个截面上对网格三角面片进行重新排序,使其沿着一个定方向(逆时针或者是顺时针)排序. 最后在此基础上,得到网格带有顺序的一系列三角带.

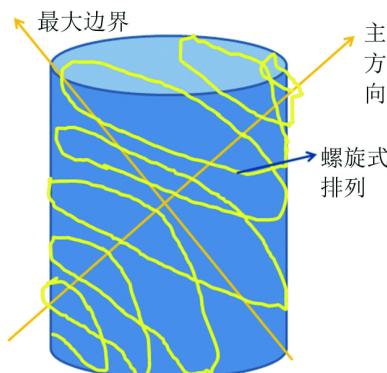


图 1 网格模型的几种方向示意图

Fig.1 Several direction patterns of grid models

2.2 在粗排序基础上全排序

我们把流网格 SM 的三角面片按照面索引值分为 K 组, 第 i 组三角面片的个数为 n_i , 其中

$$n_1 = n_2 = \dots = n_{K-1} = \left[\frac{N}{K} \right], n_K = \left[\frac{N}{K} \right] + N \% K.$$

对于每一组三角面片, 我们分别进行再排序.

我们以第 i 组三角面片进行说明, 我们记第 i 组三角面片构成的集合 $F_i = \{f'_1, f'_2, \dots, f'_{n_i}\}$, 这里

$f'_j, j \in \{1, 2, \dots, n_i\}$, 表示第 i 组的第 j 个三角面片. 假定第 i 组三角面的中点构成点集 $S1 = \{c_1^1, c_2^1, \dots, c_{n_i}^1\}$, 其中 c_j^1 是 F_i 中第 j 个面的中点. 我们构建平面 P , 该平面 P 是由第一个面的中点 c_1^1 以及从 c_1^1 出发的第一个面的法向量 p_1 和第一个面中点指向第二个面中点的向量 $p_2 = c_1^1 c_2^1$ (若 p_1, p_2 平行, 则 p_2 换成 $p_3 = c_1^1 c_3^1$, 一直重复, 直至找到 $p_t = c_1^1 c_t^1$, p_t 不平行于 p_1) 确定. $S1$ 投影到平面 P 上得到一个对应的新点集 $S2 = \{c_1^2, c_2^2, \dots, c_{n_i}^2\}$, c_j^2 是 c_j^1 在 P 上的投影点.

o_2 是由 $S2$ 构成的凸多边形的中点, 我们计算向量 $o_2 c_j^2$ 与每个向量 $o_2 c_j^2$ 的夹角(θ_j), 在平面 P 上按照逆时针顺序根据角度的大小对 F_i 中的三角面片重新排序得到新的面集合 $F_i^{\text{first}} = \{f_1^F, f_2^F, \dots, f_{n_i}^F\}$ (若 $\theta_i \leq \theta_j$, 则第 i 个三角面 f_i 要排在第 j 个三角面 f_j 前面).

对每组 F_i 执行上述操作, 即保证了对网格的所有三角面片都进行了排序(三角面片数目的有穷性和投影不变性决定的), 同时也保证了经过我们的方法排序后得到新网格($M^{\text{first}} = \{f_1^F, f_2^F, \dots, f_N^F\}$)其三角面片的相关性相比于流网格得到了进一步的加强(三角网格的所有面片有了一个更加良好的排序). M^{first} 与 SM 相比, 只是三角面片的顺序发生了变化.

但是 M^{first} 并不是一条条三角带表示的. 我们已经知道三角带在图形学方面有很多的应用. 下一步需要对 M^{first} 进行三角带表示.

2.3 输出一系列有序的三角带

在得到 M^{first} 的基础上, 我们按照开始分组的步骤初步地将网格 M^{first} 中三角面片分为 K 组, 即

$$\{f_1^F, f_2^F, \dots, f_{n_i}^F\}, \{f_{n_i+1}^F, f_{n_i+2}^F, \dots, f_{n_i+n_2}^F\}, \dots, \{f_{n_1+n_2+\dots+n_{K-1}+1}^F, f_{n_1+n_2+\dots+n_{K-1}+2}^F, \dots, f_{n_1+n_2+\dots+n_{K-1}+n_K}^F\},$$

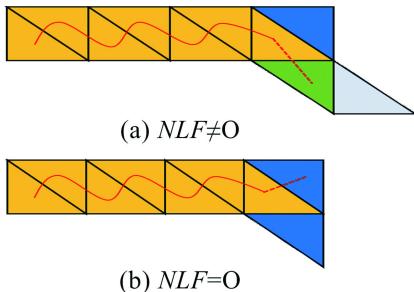
其中, n_1, n_2, \dots, n_K 如节 2.2 中的定义. 我们从第一组开始遍历每个三角面, 采用定义的深度优先搜索算法寻找一条最长的三角带, 然后对剩余的三角面, 我们把它划分到下一组, 在下一组寻找一条最长的三角带, 循环进行, 直到网格中剩下的三角面都是孤立的.

深度优先搜索: 假设当前执行到第 i 组, 第 i 组三角面片构成点集 $F_i^I = \{f_1^I, f_2^I, \dots, f_{n_i}^I\}$, 在 F_i^I 中正在执行的三角带 $TS = \{f_1^{IS}, f_2^{IS}, \dots, f_j^{IS}\}$, 集合 $F_i^R = \{f_m^R \mid f_m^R \in F_i^I, f_m^R \notin TS\} = F_i^I - TS$. E' 表示网格 M^{first} 中所有边的集合, 集合 $LF = \{f_m^{IL} \mid f_m^{IL} \in F_i^R, f_m^{IL} \cap f_j^{IS} \neq \emptyset\}$.

$\in E\}$. 集合 $NLF = \{f_m^N \mid f_m^N \in LF, \exists f_n^R \in F_i^R, f_n^R \cap f_m^N \in E\}$. 定义集合 RF 如下, RF 的定义形式是一种深度优先搜索的, 可以保证三角带尽可能长.

$$RF = \begin{cases} LF, NLF = O; \\ NLF, NLF \neq O \end{cases} \quad (2)$$

为了尽可能减少三角带扭曲, 我们对每一个 $f_m^L \in RF$, 计算 f_m^L 的 3 个顶点的 1-邻域面在现有三角带 TS 中的个数之和, 记为 sum , 选择 sum 值最小的三角面 f_m^L 作为下一个 f_{j+1}^S 加入到集合 TS 中, 如图 2(a) 所示. 在图 2(a) 中, 由于 $NLF \neq O$, 所以三角带的生长方向沿着绿色三角形(NLF)的方向延长. 在图 2(b) 中, 在 $NLF = O$ 的情况下, 两个蓝色三角形, 由于靠近上方的三角形的 $sum = 3$, 靠近下方的三角形的 $sum = 4$, 所以三角带的生长方向如图 2(b) 所示.



黄色区域表示 TS , 其他颜色区域为 F_i^R . 绿色和蓝色区域表示 LF , 绿色区域表示 NLF . 红色实线连接已经存在的三角带, 红色虚线指向下一个要加入到 TS 中的三角面

图 2 三角带的选取

Fig.2 Selection of triangle strips

本文定义的深度优先算法既继承了前人深度优先算法生成三角带的思想, 同时又有着自己独特之处, 在我们的定义下, 由于网格的三角面片是有限个数的, 显然是可以保证对每组的每一个的三角面片都是可以遍历到的, 而且找到的三角带是最优的(在三角带尽可能少扭曲下, 三角带是最长的).

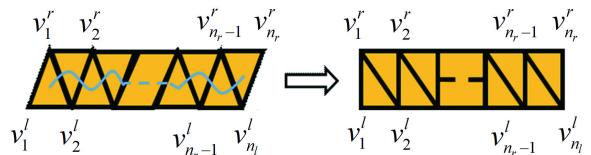
假设我们得到了 M^{first} 的 m_1 条三角带, 为了得到更加有序的输出结果, 需要进一步对三角带进行排序. 首先按照得到三角带的顺序将每条三角带的首个三角面存入到集合 FF 中, $FF = \{f'_1, f'_2, \dots, f'_{m_1}\}$. 然后找到这些三角面对应到网格 M^{first} 中的三角面的索引值, 得到一个新的数集 D , $D = \{d_1, d_2, \dots, d_{m_1}\}$, 其中 d_i 表示三角面 $f'_i \in FF$ 在网格 M^{first} 中的面索引值. 按照 D 中值的大小顺序对得到的所有三角带再次排序, 若 $d_i < d_j$, 则第 i

组三角带排在第 j 组三角带的前面. 最后得到一个新的网格 $M^{\text{second}} = \{f_1^S, f_2^S, \dots, f_N^S\}$, M^{second} 与 M^{first} 相比, 只是三角面片的顺序发生了变化.

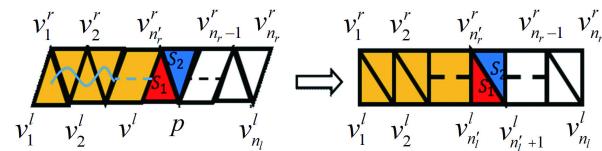
最后, 我们分析得到的一系列三角带, 通过聚合将短的三角带聚集为长的三角带. 由于三角带的生长方向与网格的排序密切相关, 所以得到的三角带是有序的, 这是前人的工作中没有的, 除了应用于三角带的一些常用工作(不要求三角带的生长方向), 同时还能将我们的工作应用到大型网格的 out-of-core 算法、网格切片等.

2.4 条带纹理映射

我们把网格的每一条三角带参数化到一个条形矩形区域上, 这里我们取一条三角带 ST 说明, 其中集合 $VST = \{v_1^r, v_2^r, \dots, v_{n_r}^r, v_1^l, v_2^l, \dots, v_{n_l}^l\}$ 表示三角带 ST 中顶点的集合. 我们用两个集合 $R_t = \{v_1^r, v_2^r, \dots, v_{n_r}^r\}, L_t = \{v_1^l, v_2^l, \dots, v_{n_l}^l\}$ 分别表示三角带 ST 参数化到矩形区域上边和下边顶点的集合. 其中 R_t 中顶点在集合中的顺序与该顶点被参数化到矩形上边区域从左到右的位置一一对应, L_t 同之, 如图 3(a) 所示. 下面对 R_t, L_t 中的顶点选取做详细介绍.



(a) 一条完整的三角带参数化映射到一个矩形区域



(b) 三角带参数化细节

图 3 三角带的参数化

Fig.3 Parameterization of triangle strips

对于三角带 ST , 假设现在已经存在 $R'_t = \{v_1^r, v_2^r, \dots, v_{n_r'}^r\}, L'_t = \{v_1^l, v_2^l, \dots, v_{n_l'}^l\}$ 分别为已经参数化到矩形区域上边和下边顶点的集合. 现在判定顶点 p ($p \in VST$, 且 $p \notin R'_t, p \notin L'_t, p \neq v_{n_r'}^r, v_{n_l'}^l$ 构成三角面 S_1) 是参数化到矩形上边还是下边, 也即判断 p 是加入到集合 R_t 还是加入到 L_t 中. 在三角带 ST 中找到与三角面 S_1 的边相连的三角面 S_2 , 且三角面 S_2 还没有被参数化. S_1 与 S_2 共同边的两个顶点分别是 p 和 v_t , 若 $v_t = v_{n_r'}^r$, 则将 p 加入到 L'_t 中. 否则

$v_t = v_{n_t}^l$, 将 p 加入到 R_t' 中. 如图 3(b) 所示 , 其中由于 $v_{n_t}^r \in R_t'$, 则 p 加入到 L_t' 中.

由于得到的每条三角带长度是不同的 , 若仅仅只取长度固定的条带纹理 , 则会出现过于缩小或放大现象 , 这里我们选取结构性纹理 , 根据三角带长度的大小 , 选取不同数量的纹理单元 . 条纹纹理映射的结果如图 4 所示 .

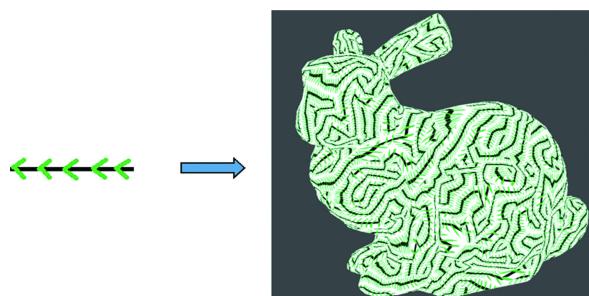


图 4 纹理映射. 纹理单元(左) Bunny 模型纹理映射的结果(右)

Fig.4 Texture mapping. The texture unit (left) , result of the texture mapping of the Bunny model (right)

3 实现结果

我们采用 APPACK 库来计算对偶网格邻接矩阵的第二小特征值. 在实验中不确定的参数是在对网格进行第一次排序时每组三角面的个数 , 这里我们记参数为 Num. 按照网格模型的三角面布局格式遍历网格所有的三角面片 , 得到的轨迹距离之和 , 记为 trace , 这里的距离指每两个三角面中点的欧氏距离. 通过遍历选取使网格 M^{first} 的 trace 值最小的每组三角面的个数作为每个网格模型的 Num 值.

首先 对于 Bunny 模型 , 分别比较其原始网格 M 、文献 [1] 提出的流网格 SM 、第一次排序后的网格 M^{first} 、第二次排序后的网格 M^{second} 的可视化结果 , 如图 5 所示 , 其中 Bunny 模型中参数 Num = 98. 由图 5 可以明显看出经过重新排序的网格相比于原始网格 M , 其三角面布局相关性得到了加强 , 而且从图 5(d) 可知 M^{second} 是三角带表示的 , 而且三角带的生长方向与网格三角面片的排序密切相关.

其次 , 我们按照网格三角面的输出顺序 , 将网格三角面的中点用折线依次连接 , 我们输出原始网格 M 、流网格 SM 、第一次排序后的网格 M^{first} 、第二次排序后的网格 M^{second} 的结果 , 如图 6 所示. 由图 6(b) 可知 , 虽然 SM 的三角面布局相关性相比于 M 得到加强 , 但是在整体上其三角面布局是混乱的.

经过对比可知 , 经过我们的算法得到的 M^{first} 和 M^{second} 的三角面布局要比 SM 更加有序 , 并且 M^{second} 是三角带表示的 , 但是通过图 6(d) 可以看到有少量的孤立黑点 , 也即是孤立三角面的存在 .

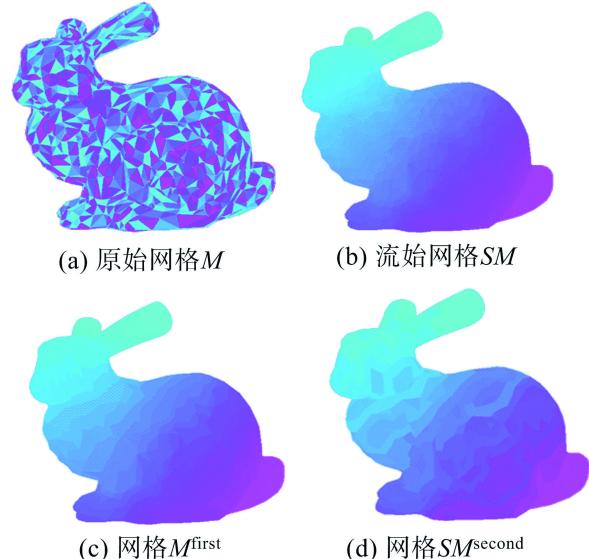


图 5 网格模型的可视化. 颜色编码是随着三角面在网格中的位置渲染的

Fig.5 Visualization of the mesh model. Color coding is rendered with the location of the triangle in the mesh

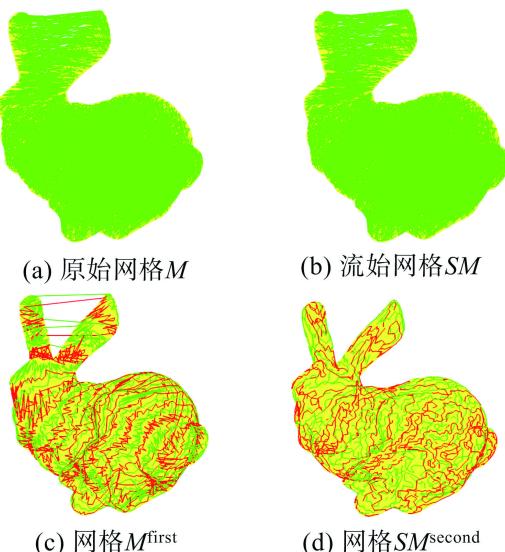


图 6 网格模型按三角面布局的面中点连线

Fig.6 Middle point connection of mesh faces according to the facet layout

显然对于同一网格模型 , 三角面布局的相关性越强 , 网格的 trace 值越小. 我们比较不同网格模型在 4 种不同排序下的 trace 值 , 分别记为 $\text{trace}_{\text{original}}$, $\text{trace}_{\text{stream}}$, $\text{trace}_{\text{first}}$, $\text{trace}_{\text{second}}$, 比较结果如表 1 所示 .

表 1 不同模型在不同排序下的轨迹距离(cm)
Tab.1 Trajectory distance of different models in different sort (cm)

mesh model	Num	trace _{original}	trace _{stream}	trace _{first}	trace _{second}
Bunny	98	3 862.021 973	1 429.676147	47.641 478	84.351 822
Eight	85	79.020 1	267.889	41.959	18.946 7
Ant	77	428.412 09	260.314 972	43.801 105	29.513 6
Cylinder	18	3 208.352 295	80.881 905	9.601 136	3.180 664
Cactus	18	279.875	67.286 8	27.165 7	10.097 5
King	34	77.498 5	63.177 3	9.625 59	9.696 52
David	233	5 590.375 000	3 170.777 832	53.728 31	68.825 8
Arm	195	988.523	397.691	36.579 6	16.985 4

一方面，网格三角面顺序越有序，数据载入到内存的速度越快。另一方面，在绘制网格时，由于网格三角面布局相关性增强，导致数据在 CPU 与 GPU 之间传输效率更高，而且三角带在传输过程中具有更低的传输代价（一条具有 n 个三角面的顶点仅仅需要 $n+2$ 的顶点被传输）且能很好地被图形引擎支持。所以对于网格在计算机的显示过程来说，当对输入的原始网格模型的三角面片进行排序后，三角面片越有序，网格在计算机的显示时间越短，而且具有一个良好三角带表示的网格的显示时间更短。不同原始网格模型在 4 种不同排序下的网格显示时间的对比，如表 2 所示。

表 2 不同模型在不同排序下的载入-显示时间(ms)

Tab.2 The loading-to-display time on different mesh models with four different facet layout (ms)

mesh model	size/kB	M	SM	M_{first}	M_{second}
Bunny	8 475	15.784	14.567	12.032	10.041
Eight	7 838	20.791	20.047	15.243	10.076
Ant	1 024	25.412	20.314	15.801	12.513
Cylinder	9 753	20.352	18.881	14.601	13.180
Cactus	14 579	20.791	20.047	15.432	10.097
King	4 054	15.242	10.982	9.063	7.203
David	6 958	14.153	14.612	12.601	10.033
Arm	5 286	13.007	11.842	11.036	9.034

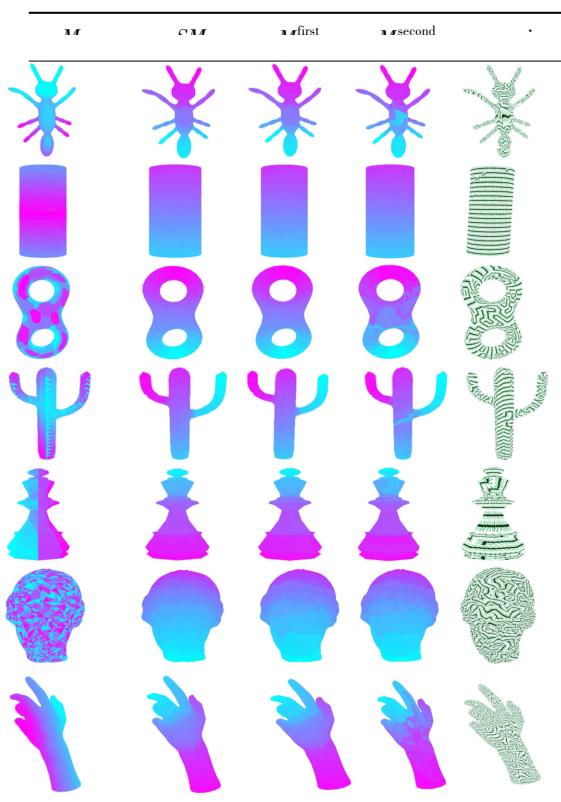
表 1、表 2 的实验数据进一步证明了经过我们的算法对网格进行排序后，网格的相关性得到了进一步的加强，而且网格有一个很好的三角带表示。

对排序前和排序后的不同网格的可视化如表 3

所示，通过表 3 可以直观看到，经过我们的方法排序后，网格的三角面有良好的相关性，而且得到的三角带是有序的。

表 3 不同模型在不同排序下可视化及纹理映射结果

Tab.3 visualization and texture mapping results of different models in different order



4 结论

本文提出一个基于谱方法进行网格三角面排序的方法。此方法通过改进流网格的算法，进而得到一个网格三角面输出顺序是全序的网格三角面布局。

格式，并且在此基础上得到一系列有序的三角带，最后我们进行了一个条纹纹理贴图的应用。但是我们在得到一系列三角带的同时，会存在少量的孤立三角面。在接下来的工作中，我们研究怎样减少孤立三角面的数量以及使三角带尽可能长。另外，在应用方面，我们仅仅考察了条纹纹理贴图，将来还会研究我们的工作在网格无损压缩、变形、渲染等方面的应用。

参考文献(References)

- [1] ISENBURG M , LINDSTROM P. Streaming meshes [C]// VIS 05. IEEE Visualization , 2005. IEEE , 2005: 231-238.
- [2] WU J , KOBBELT L. A stream algorithm for the decimation of massive meshes [C]// Proceedings of Graphics Interface 2003. Waterloo , Canada: Canadian Human-Computer Communications Society , 2003: 185-192.
- [3] ISENBURG M , GUMHOLD S. Out-of-core compression for gigantic polygon meshes [C]// ACM Transactions on Graphics (TOG) . New York: ACM , 2003 , 22 (3) : 935-942.
- [4] DEERING M. Geometry compression [C]// Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM , 1995: 13-20.
- [5] HOPPE H , MILLER J W. Optimization of mesh locality for transparent vertex caching: US , US 6426747 B1 [P]. 2002: 269-276.
- [6] BOGOMJAKOV A , GOTSMAN C. Universal rendering sequences for transparent vertex caching of progressive meshes [J]. Computer Graphics Forum , 2002 , 21 (2) : 137-149.
- [7] TCHIBOUKDJIAN M , DANJEAN V , RAFFIN B. Binary mesh partitioning for cache-efficient visualization [J]. IEEE Transactions on Visualization & Computer Graphics , 2010 , 16 (5) : 815-828.
- [8] GAREY M R , JOHNSON D S , TARJAN R E. The planar Hamiltonian circuit problem is NP-complete [J]. Siam Journal on Computing , 1976 , 5 (4) : 704-714.
- [9] AKELEY K , HAEBERLI P , BURNS D. Tomesh. c: C program on SGI developer's toolbox CD [R]. Milpitas , CA: Silicon Graphics , 1990: T990.
- [10] EVANS F , SKIENA S , VARSHNEY A. Optimizing triangle strips for fast rendering [C]// Proceedings of the 7th Conference on Visualization '96. IEEE , 1996: 319-326.
- [11] CHOW M M. Optimized geometry compression for real-time rendering [C]// Proceedings of the 8th Conference on Visualization '97. IEEE , 1997: 347-ff.
- [12] SPECKMANN B , SNOEYINK J. Easy triangle strips for tin terrain models [J]. International Journal of Geographical Information Science , 2001 , 15 (4) : 379-386.
- [13] XIANG X , HELD M , MITCHELL J S B. Fast and effective stripification of polygonal surface models [C]// Proceedings of the 1999 Symposium on Interactive 3D Graphics. New York: ACM , 1999: 71-78.
- [14] SHAFAE M , PAJAROLA R. Dstrips: Dynamic triangle strips for real-time mesh simplification and rendering [C]// Proceedings Pacific Conference on Computer Graphics and Applications , 2003. IEEE , 2003: 271-280.
- [15] DEMAIN E D , EPPSTEIN D , ERICKSON J , et al. Vertex-unfoldings of simplicial manifolds [C]// Proceedings of the Eighteenth Annual Symposium on Computational Geometry. New York: ACM , 2002: 237-243.
- [16] PORCU M B , SCATENI R. Partitioning meshes into strips using the enhanced tunnelling algorithm (ETA) [C]// VRIPHYS 2006. Geneve , Switzerland: Eurographics Association , 2006: 61-70.
- [17] 秦爱红 , 石教英. 基于混合模式缓存优化的三角形条带化 [J]. 计算机辅助设计与图形学学报 , 2011 , 23 (6) : 1006-1012.
- [18] QIN Aihong , SHI Jiaoying. Cache-friendly triangle strip generation based on hybrid model [J]. Journal of Computer Aided Design & Computer Graphic , 2011 , 23 (6) : 1006-1012.
- [19] 张洁 , 吴佳泽 , 郑昌文 等. 应用哈密顿回路的三角网格拓扑压缩 [J]. 计算机辅助设计与图形学学报 , 2013 , 25 (5) : 697-707.
- [20] ZHANG Jie , WU Jiaze , ZHEN Changwen , et al. Connectivity compression of triangle meshes based on Hamiltonian cycle [J]. Journal of Computer Aided Design & Computer Graphic , 2013 , 25 (5) : 697-707.
- [21] 魏潇然 , 耿国华 , 张雨禾. 几何信息预测的三角网格模型拓扑压缩 [J]. 西安电子科技大学学报(自然科学版) , 2015 , 42 (5) : 194-199.
- [22] WEI Xiaoran , GEN Guohua , ZHANG Yuhe. Connectivity compression of triangle meshes based on geometric parameter predict [J]. Journal of Xi'an Electronic and Science University (Natural Science Edition) , 2015 , 42 (5) : 194-199.

(下转第 748 页)

- sums over Galois rings with application to some binary sequences derived from Z_2 [J]. IEEE Trans Inform Theory, 2006, 52(5): 2260-2265.
- [10] LAHTONEN J, LING S, SOLÉ P, ZINOVIEV D V. Z_8 -Kerdock codes and pseudo-random binary sequences [J]. J Complexity, 2004, 20: 318-330.
- [11] WAN Z X. Finite Fields and Galois Rings [M]. Singapore: World Scientific, 2003.
- [12] LIDL R, NIEDERREITER H. Finite Fields [M]. Cambridge, UK: Cambridge University Press, 1997.
- [13] GOLOMB S W, GONG G. Signal Design for Good Correlation for Wireless Communication, Cryptography and Radar [M]. New York: Cambridge University Press, 2005.
- [14] LING S, BLACKFORD J T. $Z_{p^{k+1}}$ -linear codes [J]. IEEE Trans Inform Theory, 2002, 48(9): 2592-2605.
- [15] HAMMONS A R Jr, KUMAR P V, CALDERBANK A R, et al. The Z_4 -linearity of Kerdock, Preparata, Goethals, and related codes [J]. IEEE Trans Inform Theory, 1994, 40(2): 301-309.

(上接第 728 页)

- [20] ARKIN E M, HELD M, MITCHELL J S B, et al. Hamiltonian triangulations for fast rendering [J]. Visual Computer, 1996, 12(9): 429-444.
- [21] PAJAROLA R, ANTONIJUAN M, LARIO R. QuadTIN: Quadtree based triangulated irregular networks [C]// Proceedings of the Conference on Visualization 2002. IEEE, 2002: 395-402.
- [22] TAUBIN G. Constructing Hamiltonian triangle strips on quadrilateral meshes [M]// Visualization and Mathematics III. Berlin: Springer, 2003: 69-91.
- [23] GOPI M, EPPSTIEN D. Single-strip triangulation of manifolds with arbitrary topology [J]. Computer Graphics Forum, 2004, 23(3): 371-379.
- [24] DIAZ-GUTIERREZ P, BHUSHAN A, GOPI M, et al. Single-strips for fast interactive rendering [J]. Visual Computer, 2006, 22(6): 372-386.
- [25] GURUNG T, LUFFEL M, LINDSTROM P, et al. LR: compact connectivity representation for triangle meshes [J]. ACM Transactions on Graphics, 2011, 30(4): 76-79.
- [26] FLOATER M S. Parametrization and smooth approximation of surface triangulations [J]. Computer Aided Geometric Design, 1997, 14(3): 231-250.