

基于张量投票的快速网格分割算法

舒振宇^{1,2}, 汪国昭¹

(1. 浙江大学 数学系图像图形研究所, 浙江 杭州 310027; 2. 浙江大学宁波理工学院
信息处理与优化设计研究所, 浙江 宁波 315100)

摘 要: 为了根据网格模型上的尖锐几何特征对三角网格曲面进行合理分片, 提出一种新的基于张量投票(tensor voting)理论的三角网格分割算法. 该算法将输入网格模型上所有的三角面片聚类成由用户指定数目的若干个区域, 使得区域内部三角面片上点的尖锐几何特征尽可能接近. 根据网格模型顶点上基于法向的张量投票矩阵的特征值分布与顶点尖锐几何特征的对应关系, 算法将网格分割转化为能量最小化问题, 并适当简化能量函数的形式, 用快速聚类算法求解. 通过引入启发式约束, 算法较好地防止了分割区域的分离. 实验表明: 与已有算法相比, 该算法具有较快的速度, 同时能够较好地分割网格曲面上的尖锐几何特征区域.

关键词: 网格分割; 张量投票理论; 聚类; 法向; 几何特征

中图分类号: TP 391.4

文献标志码: A

文章编号: 1008-973X(2011)06-0999-07

Fast mesh segmentation algorithm based on tensor voting

SHU Zhen-yu^{1,2}, WANG Guo-zhao¹

(1. Institute of Computer Graphics and Image Processing, Department of Mathematics, Zhejiang University, Hangzhou 310027, China; 2. Laboratory of Information and Optimization Technologies, Ningbo Institute of Technology, Zhejiang University, Ningbo 315100, China)

Abstract: A novel algorithm for triangular mesh segmentation based on tensor voting theory was proposed to correctly segment the input triangular mesh according to the sharp geometrical features on the mesh. All triangles of the input mesh clustered to a user-specified number of regions such that the sharp geometrical features of vertices belonging to the same region were as similar as possible. With the correspondence between the sharp geometrical features and the distribution of normal tensor voting matrix's eigen values, the mesh segmentation was converted to an energy minimization problem. Then a fast clustering method was applied to solve the problem with simplified energy terms. By introducing a heuristic constraint, no segment was separated into disconnected parts with the algorithm. Experimental results show that the algorithm is faster and the regions with sharp geometrical features are segmented better compared with some existing algorithms.

Key words: mesh segmentation; tensor voting theory; clustering; normal; geometrical feature

网格分割技术在几何建模和数字几何处理中起着非常关键的作用. 在网格参数化、纹理映射、形状匹配、几何变形、多分辨率建模、网格编辑、网格简

化、压缩和动画等多个领域, 网格分割都有着重要的应用. 例如, Shu 等^[1]先把三角网格分割成若干个近似的重心 Voronoi 区域(centroidal Voronoi region), 然

收稿日期: 2009-12-24.

浙江大学学报(工学版)网址: www.journals.zju.edu.cn/eng

基金项目: 国家自然科学基金资助项目(60773179, 60970079); 国家自然科学基金资助项目(60933008); 宁波市自然科学基金资助项目(2009A610071).

作者简介: 舒振宇(1979—), 男, 博士生, 从事计算机图形学、数字几何处理的研究. E-mail: shuzhenyu@163.com

通信联系人: 汪国昭, 男, 教授, 博导. E-mail: wanggz@zju.edu.cn

后将每个区域用一个中心点来代替,从而达到简化网格的目的. 网格分割是指将一个网格分割成若干个互不相交的区域,这些区域的并集构成了网格本身. 网格分割算法的结果往往要满足某些约束条件,例如, Cohen-steiner 等^[2]将给定的网格分割成若干个接近于平面的区域. Gelfand 等^[3]将网格分割成一些简单的几何曲面的近似,包括平面、球面和圆柱面等. 本文提出了一种新的基于张量投票理论的网格分割算法,该算法根据网格模型每个顶点上基于法向的张量投票特征值分布情况与顶点的尖锐几何特征(点位于模型的面上,位于边上或位于角上)之间的对应关系,通过最大化分割后子网格片上的能量函数,使得同一子网格片内部点的尖锐几何特征尽量相似(即同属于模型面上的点,边上的点或角上的点)来实现对网格曲面的分割. 实验表明该算法速度较快,效果较好.

1 相关工作

近些年来,为了适应各种不同的应用场合,很多网格分割算法被提了出来. 根据分割对象的不同,这些算法可以分为 2 大类:(1)将网格模型对应的体分割成若干个子几何体^[4-6]; (2)根据网格表面上的某些几何性质将网格模型分割成若干个表面片,例如,在文献^[7-9]中,网格模型被分解成平面、圆柱面,及球面等基本几何形状. 本文提出的算法根据网格表面上的张量投票性质将网格分割成若干个表面片,因而属于上述第 2 类网格分割算法. 一般来说,对于不同的应用场合,网格分割算法对分割后得到的子网格片的要求也各不相同. 例如,在纹理映射^[10-12]、几何图像(geometry-image)^[13]等应用领域中,分割得到的子网格片应该尽量与圆盘同胚,而且在参数化到二维平面后,不应该引入较大的形变. 另外,在网格重构和网格简化等场合^[2,14-19],子网格片经常会由一系列的平面多边形来代替,因此,要求分割得到的子网格片应该尽量接近平面. 根据对分割后子网格片特征的不同要求,有越来越多的网格分割算法被提了出来. 例如, Petitjean 等^[20-21]提出了在 CAD 反向工程和建模中以一般二次曲面为分割目标的分割算法. Wu 等^[22-23]用平面、球面、圆柱面等基本几何形状来替换子网格片. Mitani 等^[24-26]提出了将网格模型分割成条带状面片和拟可展面片的分割算法. Gelfand 等^[3]提出了基于滑移分析(slippage analysis)的网格分割算法,该算法得到的结果能够保证位于同一个表面片中的顶点和区域具

有相似的滑移运动特征. Podolak 等^[27]采用对称性分析的方法来分割网格曲面. 而 Shlafman 等^[28]采用了 K-Means 聚类方法来将网格模型上的三角形聚类成若干表面片. Katz 等^[29]提出了一种基于模型表面元素之间测地距离和角距离的模糊聚类算法,该算法能够针对原模型进行层次性的网格分解,而且分解后得到的表面片之间的边界是较光滑的. 李奇敏等^[30]提出一种基于图的网格分割算法,该算法以顶点和三角面片作为图中元素,利用森林分割图来实现网格的分割. 钱江等^[31]利用全局光滑参数化技术提出了一种网格分割算法,该算法具有简单,控制灵活的优点. Shamir^[32]给出了更多相关算法的一个综述.

本文提出的算法是基于张量投票理论的三角面片聚类算法. 张量投票(tensor voting)理论^[33]在网格特征检测和网格对象识别领域有着广泛的应用. 由于张量投票理论能够较好的处理网格上的尖锐特征,而且对于噪声数据具有较强的鲁棒性,本文算法用基于法向的张量投票来构造聚类. 与前面这些方法不同的是,本文算法根据网格上点的张量投票矩阵特征值分布与尖锐几何特征(点位于模型的面上,位于边上或位于角上)之间的对应关系,通过定义反映同一子网格片内部的张量投票矩阵特征值分布差异度的能量函数,以最小化该能量函数为目标,不断的调整子网格片之间的边界来得到最终分割结果,从而使得分割后子网格片内部点的几何特征尽量一致. 由于对能量函数的简化,本文的算法效率较高.

2 基于法向的张量投票

Medioni 等^[33-34]定义三角网格顶点 v 上基于法向的张量投票 K_v 为:

$$K_v = \sum_{t_i \in N_i(v)} \mu_{t_i} n_{t_i} n_{t_i}^T.$$

$$\text{式中: } n_{t_i} \cdot n_{t_i}^T = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \cdot (a, b, c) = \begin{pmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{pmatrix},$$

$$\mu_{t_i} = \left(\frac{A_{t_i}}{A_{\max}} \right) \exp \left(- \frac{\|c_{t_i} - p_v\|}{\sigma/3} \right),$$

$N_i(v)$ 表示顶点 v 的 1-邻域内所有三角面片组成的集合, A_{t_i} 表示三角面片 t_i 的面积, A_{\max} 表示 $N_i(v)$ 中所有三角面片的面积最大值, c_{t_i} 是三角面片 t_i 的重心, p_v 是顶点 v 的几何位置, σ 表示顶点 v 的 1-邻域内三角形的最小立方包围盒的边长.

由于上述基于法向的张量投票是一个对称半正

定的张量, 可以将其对角化为

$$\mathbf{K}_v = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T + \lambda_3 \mathbf{e}_3 \mathbf{e}_3^T.$$

式中: $\lambda_1, \lambda_2, \lambda_3$ 是 \mathbf{K}_v 的特征值, $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ 是对应的特征向量. 根据特征值 $\lambda_1, \lambda_2, \lambda_3$, Shimizu 等^[35] 提出可以将三角网格上的顶点分为模型面上的点、边上的点和角上的点 3 种类型(如图 1).

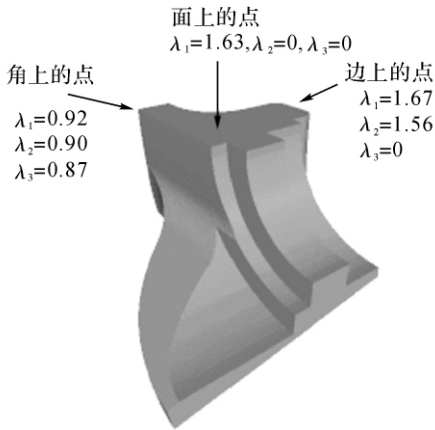


图 1 张量投票的特征值分布与顶点几何特征的对应关系
Fig. 1 Correspondence between eigen values of tensor voting and geometry feature of vertices on meshes

设 $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$, 定义顶点 v 的类型如下:

- 1) 如果 λ_1 相对 λ_2, λ_3 较大, 同时 λ_2, λ_3 都接近于 0, 则顶点 v 为模型面上的点;
- 2) 如果 λ_1 和 λ_2 相对较大, 同时 λ_3 接近于 0, 则顶点 v 为模型边上的点;
- 3) 如果 λ_1, λ_2 和 λ_3 值较接近, 则顶点 v 为模型角上的点.

以上对应关系给出了一种简捷并且鲁棒的网格上顶点几何特征的判断方法, 因此, 本文算法根据以上特征值与顶点几何特征的对应关系, 对三角网格上所有三角面片进行聚类, 使得同一集合内部的三角面片的顶点尽量属于同一类顶点.

3 目标能量函数

给定一个三角网格 M , M 由顶点集合 V 、边集合 E 和面集合 F 组成, 其中,

$$V = \{ \mathbf{p}_i \mid \mathbf{p}_i \in \mathbf{R}^3, 1 \leq i \leq m \},$$

$$E = \{ e_{ij} = (\mathbf{p}_i, \mathbf{p}_j) \mid \mathbf{p}_i, \mathbf{p}_j \in V, i \neq j \},$$

$$F = \{ f_{ijk} = (\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) \mid \mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k \in V, i \neq j, i \neq k, j \neq k \}.$$

网格分割的目标是把网格 M 分解成 n 个子网格 $M_i (i = 0, 1, \dots, n-1)$, $M_i \subset M$, $M_i = \{ V_i, E_i, F_i \}$, $V_i \in V, E_i \in E, F_i \in F, F_i \cap F_j = \emptyset, i \neq j$, 这里, n 是

用户指定的目标子网格数, 使得分割后得到的每个子网格片 M_i 内部的点尽量属于同一分类(面上的点、边上的点、角上的点). 为此, 根据张量投票 \mathbf{K}_v 的特征值与顶点特征分类的对应关系, 可以定义如下函数来度量子网格片 M_i 上任意点之间的特征分类差异:

$$E_i = \int_{M_i} \rho(x) \| \mathbf{H} - \bar{\mathbf{H}}_i \|^2 dx. \quad (1)$$

式中: \mathbf{H} 是网格片上任意点 x (不仅仅是顶点) 上张量投票的特征值归一化后组成的向量, 即 $\mathbf{H} = (1, \lambda_2/\lambda_1, \lambda_3/\lambda_1)$; $\rho(x)$ 是 M_i 上的密度函数, 一般可以看作常值函数; $\bar{\mathbf{H}}_i$ 是 M_i 上 \mathbf{H} 的平均值, 可以表示为

$$\bar{\mathbf{H}}_i = \frac{\int_{M_i} \rho(x) \mathbf{H} dx}{\int_{M_i} \rho(x) dx}.$$

为了得到满足要求的网格分割, 使得所有子网格片 M_i 上任意点之间的特征分类差异尽量小, 只需要最小化能量函数

$$E = \sum_{i=0}^{n-1} \int_{M_i} \rho(x) \| \mathbf{H} - \bar{\mathbf{H}}_i \|^2 dx. \quad (2)$$

其中 $\bar{\mathbf{H}}_i$ 可以写成

$$\bar{\mathbf{H}}_i = \frac{\sum_{T_j \in M_i} \int_{T_j} \rho(x) \mathbf{H} dx}{\sum_{T_j \in M_i} \int_{T_j} \rho(x) dx},$$

因此, 如果认为同一三角面片 T_i 内部 \mathbf{H} 保持不变, 记作 \mathbf{H}_i , 并记三角面片 T_j 的面积为 s_j , 即 $s_j = \int_{T_j} \rho(x) dx$, 则可以把能量函数 E 写成如下形式:

$$E = \sum_{i=0}^{n-1} \left(\sum_{T_j \in M_i} s_j \| \mathbf{H}_j \|^2 - \frac{\left\| \sum_{T_j \in M_i} s_j \mathbf{H}_j \right\|^2}{\sum_{T_j \in M_i} s_j} \right). \quad (3)$$

为了得到满足要求的网格分割, 只需要找到网格 M 中所有三角面片的适当聚类, 使得能量函数 E 达到最小. 由于式(3)中

$$\sum_{i=0}^{n-1} \left(\sum_{T_j \in M_i} s_j \| \mathbf{H}_j \|^2 \right)$$

的值对于任意聚类都保持不变, 因此, 为了让能量函数 E 达到最小, 只需最大化:

$$E' = \sum_{i=0}^{n-1} \left(\left\| \sum_{T_j \in M_i} s_j \mathbf{H}_j \right\|^2 / \sum_{T_j \in M_i} s_j \right). \quad (4)$$

4 算 法

本文算法的主要步骤如下:

- 1) 构造初始子网格. 在输入的三角网格模型上

随机选择 n 个三角面片作为种子三角面片(其中, n 由用户指定), 并将这些种子三角面片作为 n 个初始子网格.

2) 聚类. 反复遍历所有子网格的边界, 并以最大化能量函数(4)为目标, 对边界进行调整更新. 当所有子网格的边界不再改变时, 边界遍历结束, 并得到最终网格分割结果.

4.1 聚类过程

本文采用的聚类算法以最大化能量函数(4)为目标. 算法通过边界测试来不断的调整子网格之间的公共边界, 使得 E' 的值不断变大. 当 E' 的值停止变大时, 聚类结束. 对于任意 2 个相邻子网格之间公共边界上的边 e , 算法进行如下边界测试: 设 e 的 2 个相邻子网格为 M_i 和 M_j , e 的 2 个相邻三角面片为 T_k 和 T_l , 其中, T_k 属于 M_i , T_l 属于 M_j (如图 2). 计算 3 种情况下 E' 的值:

E' : 如果 T_k 属于 M_i , 并且 T_l 属于 M_j ;

E'_1 : 如果 T_k 和 T_l 都属于 M_i ;

E'_2 : 如果 T_k 和 T_l 都属于 M_j .

在 E' 、 E'_1 和 E'_2 中选最大值所对应的情况, 并以此更新 M_i 和 M_j 的边界. 通过反复遍历所有任意 2 个相邻子网格之间的公共边界, 并进行相应的边界测试和调整, E' 的值不断变大. 当任何子网格的边界都不再改变时, 边界遍历结束, 并得到最终的网

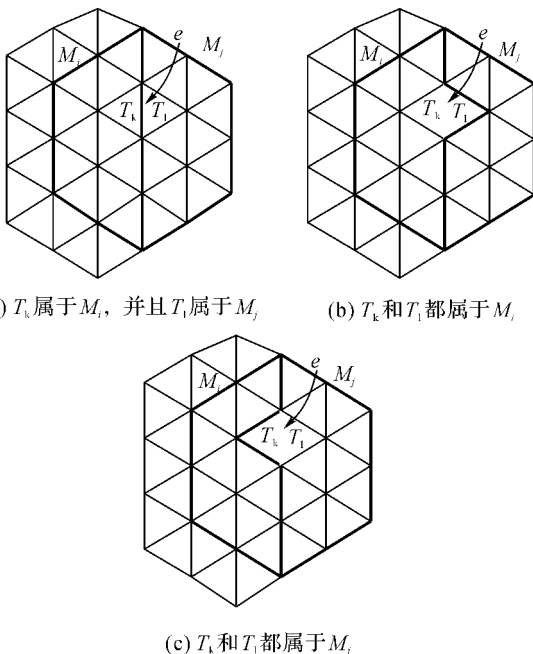


图 2 计算 3 种情况下 E' 的值, 并选择使得 E' 最大的情况

Fig. 2 Compute energy E' for three cases and case resulting maximum E' is chosen

格分割结果.

4.2 聚类的初始化和高效实现

为了构造原始网格 M 上三角面片的初始聚类, 首先在 M 上随机选择 n 个三角面片 $T_i (i=0, 1, \dots, n-1)$ 作为 n 个初始子网格, 然后遍历每个子网格的所有边界边, 对于任意子网格 M_i 边界上的任意边 e , T_k 和 T_l 是 e 的 2 个相邻三角面片, 设 T_k 属于 M_i , 如果 T_l 还未属于任何子网格, 则将 T_l 直接加入到 M_i 中; 否则, 如果 T_l 已经属于另一个子网格 $M_j (j \neq i)$, 则对边 e 执行如 4.1 节所述的边界测试, 并调整子网格 M_i 和 M_j 之间的边界, 使得 E' 变大.

为了提高执行效率, 在算法实现中需要针对每个子网格 M_i 缓存 $c_1 = \sum_{T_j \in M_i} s_j \mathbf{H}_j$ 和 $c_2 = \sum_{T_j \in M_i} s_j$ 的值. 在迭代过程中, 当任一三角面片 T_k 被加到 M_i 中, 或者从 M_i 中移除时, 只需往 c_1 和 c_2 中增加或者减去改变量 $\Delta c_1 = s_k \mathbf{H}_k$ 和 $\Delta c_2 = s_k$, 即可快速实现对能量函数 E' 的求值.

在聚类过程中, 某些子网格可能会分离成若干个互不连接的部分, 从而导致最终得到的网格分割结果中子网格的数量大于用户给定的数量 n . 为了避免出现这样的情况, 可以对聚类过程加入如下启发式约束: 当任意三角面片 T_k 从子网格 M_i 中移除, 并加入到另一子网格 M_j 中时 (如图 3 (a)), 如果这将导致子网格 M_i 的边界形状如 ‘ ∞ ’ (如图 3 (b)), 也就是说, 子网格 M_i 的边界上存在某一点 P , 使得 P 有 4 条相邻边界边 (属于 M_i), 则令该情况所对应的 E' 为 0. 由于聚类的目标为最大化 E' , 该启发式约束将防止 T_k 从 M_i 中移除, 从而有效防止任意子网格 M_i 出现分离情况. 虽然在每次边界测试时都需检查如上所述的约束, 但由于在实现中每个子网格的边界信息都已经被缓存, 由此带来的计算量并不大. 实验表明, 约束检查所需时间占总聚类时间的比例约为 10% 左右.

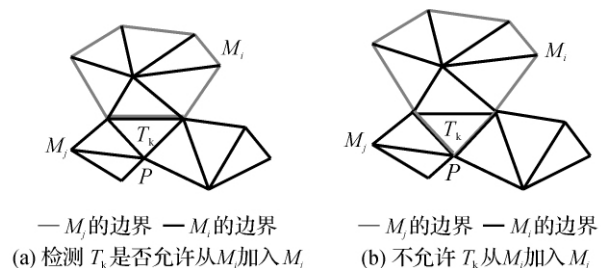


图 3 防止子网格分离成若干部分

Fig. 3 Preventing each segment from falling into several disconnected parts

4.3 算法步骤描述

输入三角网格模型 M 及用户指定的分割数量 n , 本文算法输出 M 的分割结果 $M_i (i=1, \dots, n)$. 算法执行如下:

1) 在输入的三角网格模型 M 上随机选择 n 个种子三角面片.

2) 将 n 个种子三角面片作为 n 个初始子网格 $M_i (i=1, \dots, n)$, 模型上其他三角面片不属于任何子网格.

3) 遍历所有子网格的边界边所构成的集合, 对于该集合中的任一边 e , 执行以下步骤:

(1) 如果 e 的 2 个相邻三角面片中只有一个三角面片属于某一子网格 M_i , 另一个三角面片 T_i 还未属于任何子网格, 则扩展 M_i , 将 T_i 加入 M_i .

(2) 如果 e 的 2 个相邻三角面片属于不同的子网格 M_i 和 M_j , 则对边 e 作 4.2 节所述的边界测试和调整.

(3) 如果 3) 中, 所有子网格的边界边所构成的集合不再发生改变, 算法结束, 输出所有子网格 $M_i (i=1, \dots, n)$, 即为最终分割结果. 反之, 重复 3).

4.4 收敛性

根据式 (2), 可知 $E \geq 0$ 成立, 因此, 根据式 (3), 有

$$\sum_{i=0}^{n-1} \left(\sum_{T_j \in M_i} s_j \|H_j\|^2 - \frac{\left\| \sum_{T_j \in M_i} s_j H_j \right\|^2}{\sum_{T_j \in M_i} s_j} \right) \geq 0.$$

又根据式 (4), 有

$$\sum_{i=0}^{n-1} \left(\sum_{T_j \in M_i} s_j \|H_j\|^2 \right) - E' \geq 0,$$

即

$$E' \leq \sum_{i=0}^{n-1} \left(\sum_{T_j \in M_i} s_j \|H_j\|^2 \right).$$

而对于给定的三角网格

$$\sum_{i=0}^{n-1} \left(\sum_{T_j \in M_i} s_j \|H_j\|^2 \right),$$

对于任意聚类都保持不变, 即为常数. 同时, 在 4.1 节的聚类算法的迭代过程中, E' 不断的增大, 因此, 本文的聚类算法收敛.

5 算例分析

本文算法在 Dolphin、RockerArm、Chair 等网格模型(如图 4(a)、5(a)、6(a))上进行了试验. 表 1 列出了部分试验结果, 其中运行时间是在一台具有 Intel 赛扬 CPU 1.4 GHz 和 1 GB 内存的低端 PC

机上进行测量的. 从试验结果中可以看到, 本文算法的执行效率是比较高的. 对于一些小型或中等规模的模型, 即使在一台性能较差的计算机中, 算法也能获得较快的速度, 这主要得益于本文算法所基于的快速边界测试. 图 4(a)、5(a)、6(a)显示了本文算法部分试验结果, 从中可以看到, 具有相同特征分类的邻近三角面片被尽可能的分到了同一个子网格中, 输入模型的“边”区域和“角”区域被较好的分割了出来. 表 1 还列出了针对相同网格模型, 采用 Inoue 等^[19](H 被用作区域合并的度量值)提出的算法的试验结果, 从表中可以看到, 本文算法速度远高于文献[19]中提出的算法. 图 4(b)、5(b)、6(b)显示了文献[19]算法的部分实现结果. 从图中也可以看到, 本文算法的分割结果更合理一些.

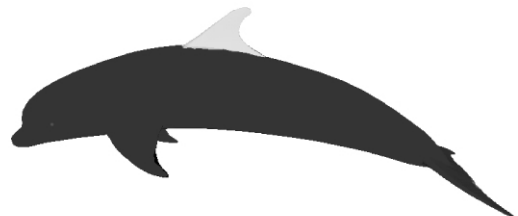
表 1 针对例子模型用本文算法和 Inoue 等人的算法^[19]得到的结果

Tab. 1 Results obtained from using our method and Inoue's method^[19] for reference models

| 图 | 模型 | 面片数 | 分块数 | 运行时间/s |
|-----|------------------|--------|-----|-----------|
| 图 4 | Dolphin(本文) | 10 000 | 10 | 1.171 |
| 图 5 | Dolphin(文 19) | 10 000 | 10 | 28.625 |
| 图 6 | Rocker Arm(本文) | 18 794 | 7 | 2.156 |
| 图 7 | Rocker Arm(文 19) | 18 794 | 7 | 116.515 |
| 图 8 | Chair(本文) | 99 944 | 50 | 20.016 |
| 图 9 | Chair(文 19) | 99 944 | 50 | 3 633.330 |



(a) 本文算法



(b) Inoue等人算法^[19]

图 4 Dolphin 模型, 面片数 10 000, 用不同算法分割成 10 块(不同的颜色仅用来区分相邻的子网格)

Fig. 4 Dolphin model with 10000 faces, which is partitioned to 10 segments from using different algorithms

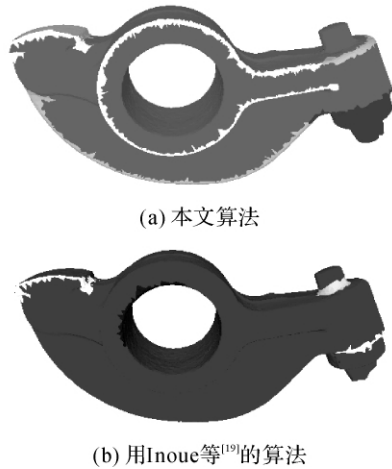


图5 Rocker Arm 模型,面片数 18 794,用本文算法分割成 7 块(不同的颜色仅用来区分相邻的子网格)

Fig. 5 Rocker Arm model with 18 794 faces, which is partitioned to 7 segments from using different algorithms

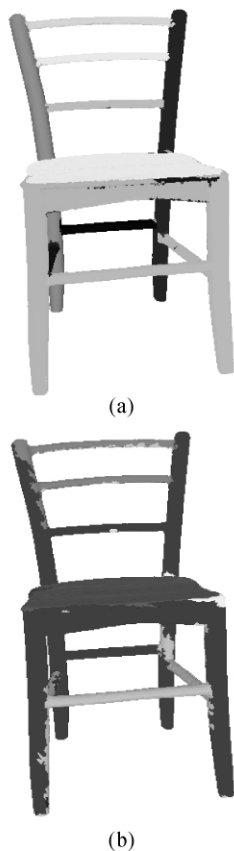


图6 Chair 模型,面片数 99 944,用不同算法分割成 50 块(不同的颜色仅用来区分相邻的子网格)

Fig. 6 Chair model with 99 944 faces, which is partitioned to 50 segments from using different algorithms

本文的算法通过优化能量函数值得到最终分割结果.在某些情况下,这将导致子网格与子网格之间的公共边界不够光滑.对于这种情况,通常可以采用

一些后置处理算法来提高边界的光顺性,这是将来要解决的问题之一.

6 结 语

本文提出了一种新的基于张量投票的快速网格分割算法.算法通过引入基于张量投票的度量,把网格分割转化成了能量最小化问题,并用快速三角面片聚类算法得到分割结果,实验表明本文算法是高效的.今后将研究如何进一步提高分割算法的效率.

参考文献 (References):

- [1] SHU Z, WANG G, DONG C. Adaptive triangular mesh coarsening with centroidal Voronoi tessellations [J]. *Journal of Zhejiang University-Science A*, 2009, 10(4): 535 - 545.
- [2] COHEN-STEINER D, ALLIEZ P, DESBRUN M. Variational shape approximation[C]// *ACM SIGGRAPH 2004 Papers*. Los Angeles: ACM, 2004: 905 - 914.
- [3] GELFAND N, GUIBAS L J. Shape segmentation using local slippage analysis[C]// *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. Nice, France: ACM, 2004: 214 - 223.
- [4] RENIERS D, TELEA A. Hierarchical part-type segmentation using voxel-based curve skeletons [J]. *The Visual Computer*, 2008, 24(6): 383 - 395.
- [5] LAI Y K, HU S M, MARTIN R R, et al. Fast mesh segmentation using random walks[C]// *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling*. New York: ACM, 2008: 183 - 191.
- [6] MORTARA M, PATAN G, SPAGNUOLO M. From geometric to semantic human body models [J]. *Computers & Graphics*, 2006, 30(2): 185 - 196.
- [7] VARADY T, MARTIN R R, COX J. Reverse engineering of geometric models—an introduction [J]. *Computer-Aided Design*, 1997, 29(4): 255 - 268.
- [8] SHAH J J, ANDERSON D, KIM Y S, et al. A discourse on geometric feature recognition from CAD models [J]. *Journal of Computing and Information Science in Engineering*, 2001, 1(1): 41 - 51.
- [9] IP C Y, REGLI W C. Manufacturing classification of CAD models using curvature and SVMs [C]// *Proceedings of the International Conference on Shape Modeling and Applications 2005*. Cambridge: IEEE Computer Society, 2005: 363 - 367.
- [10] SANDER P V, SNYDER J, GORTLER S J, et al. Texture mapping progressive meshes [C]// *Proceedings of the 28th Annual Conference on Computer Graphics*

- and Interactive Techniques. New York: ACM, 2001: 409 - 416.
- [11] SORKINE O, COHEN-OR D, GOLDENTHAL R, et al. Bounded-distortion piecewise mesh parameterization [C]// **Proceedings of the Conference on Visualization '02**. Boston: IEEE Computer Society, 2002: 355 - 362.
- [12] ZHANG E, MISCHAIKOW K, TURK G. Feature-based surface parameterization and texture mapping[J]. **ACM Transactions on Graphics**, 2005, 24(1): 1 - 27.
- [13] SANDER P V, WOOD Z J, GORTLER S J, et al. Multi-chart geometry images[C]// **Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry processing**. Aachen: Eurographics Association, 2003: 146 - 155.
- [14] KALVIN A D, TAYLOR R H. Superfaces: Polygonal Mesh Simplification with Bounded Error[J]. **IEEE Computer Graphics and Applications**, 1996, 16(3): 64 - 77.
- [15] GARLAND M, WILLMOTT A, HECKBERT P S. Hierarchical face clustering on polygonal surfaces[C]// **Proceedings of the 2001 Symposium on Interactive 3D graphics**. New York: ACM, 2001: 49 - 58.
- [16] SHEFFER A. Model simplification for meshing using face clustering[J]. **Computer-Aided Design**, 2001, 33(13): 925 - 934.
- [17] ZUCKERBERGER E, TAL A, SHLAFMAN S. Polyhedral surface decomposition with applications [J]. **Computers & Graphics**, 2002, 26(5): 733 - 743.
- [18] BOIER-MARTIN I M. Domain decomposition for multiresolution analysis[C]// **Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry processing**. Aachen: Eurographics Association, 2003: 31 - 40.
- [19] INOUE K, ITOH T, YAMADA A, et al. Face clustering of a large-scale CAD model for surface mesh generation [J]. **Computer-Aided Design**, 2001, 33(3): 251 - 261.
- [20] PETITJEAN S. A survey of methods for recovering quadrics in triangle meshes[J]. **ACM Computing Surveys**, 2002, 34(2): 211 - 262.
- [21] LAVOU G, DUPONT F, BASKURT A. A new CAD mesh segmentation method, based on curvature tensor analysis[J]. **Computer-Aided Design**, 2005, 37(10): 975 - 987.
- [22] WU Jian-hua, LEIF K. Structure recovery via hybrid variational surface approximation[J]. **Computer Graphics Forum**, 2005, 24(3): 277 - 284.
- [23] ATTENE M, FALCIDIENO B, SPAGNUOLO M. Hierarchical mesh segmentation based on fitting primitives [J]. **The Visual Computer**, 2006, 22(3): 181 - 193.
- [24] MITANI J, SUZUKI H. Making papercraft toys from meshes using strip-based approximate unfolding[J]. **ACM Transactions on Graphics**, 2004, 23(3): 259 - 263.
- [25] DAN J, KRAEVOY V, SHEFFER A. D-Charts: Quasi-Developable Mesh Segmentation[J]. **Computer Graphics Forum**, 2005, 24(3): 581 - 590.
- [26] SHATZ I, TAL A, LEIFMAN G. Paper craft models from meshes[J]. **The Visual Computer**, 2006, 22(9): 825 - 834.
- [27] PODOLAK J, SHILANE P, GOLOVINSKIY A, et al. A planar-reflective symmetry transform for 3D shapes[C]// **ACM SIGGRAPH 2006 Papers**. Boston: ACM, 2006: 549 - 559.
- [28] SHLAFMAN S, TAL A, KATZ S. Metamorphosis of Polyhedral Surfaces using Decomposition[J]. **Computer Graphics Forum**, 2002, 21(3): 219 - 228.
- [29] KATZ S, TAL A. Hierarchical mesh decomposition using fuzzy clustering and cuts[J]. **ACM Transactions on Graphics**, 2003, 22(3): 954 - 961.
- [30] 李奇敏,柯映林,李江雄. 基于图的网格分割[J]. **浙江大学学报:工学版**, 2007, 41(5): 476 - 751.
LI Qi-min, KE Ying-lin, LI Jiang-xiong. Graph based mesh segmentation[J]. **Journal of Zhejiang University: Engineering Science**, 2007, 41(5): 476 - 751.
- [31] 钱江,陈志杨,叶修梓,等. 基于参数化技术的网格分割[J]. **浙江大学学报:工学版**, 2008, 42(8): 1370 - 1375.
QIAN Jiang, CHEN Zhi-yang, YE Xiu-zi, et al. Mesh segmentation based on mesh parameterization techniques[J]. **Journal of Zhejiang University: Engineering Science**, 2008, 42(8): 1370 - 1375.
- [32] SHAMIR A. A survey on mesh segmentation techniques[J]. **Computer Graphics Forum**, 2008, 27(6): 1539 - 1556.
- [33] MEDIONI G, TANG C, LEE M. Tensor voting: Theory and applications[C]// **Proceedings of RFIA 2000**. Paris: Laas-CNRS, 2000: 38 - 48.
- [34] SUN Y, PAGE D L, PAIK J K, et al. Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing[C]// **Proceedings of IEEE International Conference on Image Processing**. New York: IEEE Computer Society, 2002: 825 - 828.
- [35] SHIMIZU T, DATE H, KANAI S, et al. A new bilateral mesh smoothing method by recognizing features [C]// **Proceedings of the Ninth International Conference on Computer Aided Design and Computer Graphics**. [S. l.]: IEEE Computer Society, 2005: 281 - 286.