

采用张量投票理论的三角网格特征边提取算法

张慧娟^{1,2)}, 耿博^{1,2)}, 汪国平^{1,2)}

¹⁾(北京大学计算机系人机交互与多媒体实验室 北京 100871)

²⁾(北京大学机器感知与智能教育部重点实验室 北京 100871)

(zhanghj@graphics.pku.edu.cn)

摘 要: 识别与提取模型几何特征在几何模型的编辑处理中起着重要作用, 然而大多数已有算法在处理质量较差的三角网格模型时往往会失效, 为此提出一种基于张量投票理论的特征边提取算法. 首先根据张量投票矩阵特征值分布与顶点几何特征之间的对应关系对顶点进行分类; 采用断点连接方法来保证顶点分类过程中能够正确地区分平滑特征上的边点及角点; 根据顶点的类别结果进行区域增长, 并提取区域增长后的边界, 从而得到网格特征边. 实验结果表明, 文中算法对大多数模型可靠有效, 能够处理网格分布不均匀, 以及含有狭长三角形或含有孔、缝的模型, 处理有噪声的模型也能达到较好的效果.

关键词: 张量投票理论; 区域增长; 聚类; 特征边提取
中图法分类号: TP391

Feature Edge Extraction Method of Triangle Meshes Based on Tensor Voting Theory

Zhang Huijuan^{1,2)}, Gen Bo^{1,2)}, and Wang Guoping^{1,2)}

¹⁾(HCI & Multimedia Laboratory of Department of Computer Science, Peking University, Beijing 100871)

²⁾(The Key Laboratory of Machine Perception and Intelligent, MOE, Beijing 100871)

Abstract: Feature detection and extraction plays important role in mesh editing. However, most existing algorithms often fail in dealing with irregular meshes. To overcome those problems, an algorithm for extracting feature edges of triangle meshes based on tensor voting is presented. First, all vertices of an input mesh are classified according to the observation that there is a close correspondence between the eigenvalue distribution of the tensor voting matrix and geometrical features. The classified vertices are then optimized by connecting breakpoints. Region growing is performed for each seed triangle and the boundaries of the regions are extracted as the edges. The experimental results show that the proposed algorithm is effective in nearly all cases, including models with non-uniformly distributed triangles, long and narrow triangles or even holes. It is also robust on noisy data.

Key words: tensor voting theory; region growing; clustering; edge extraction

随着坐标测绘及激光扫描等技术的迅速发展和应用, 复杂三维模型数据的采集技术走向成熟, 并在游戏动画、产品设计与制造等领域中得到了广泛应用. 把扫描得到的点云数据重建为网格模型有利于模型的形状分析和编辑形变, 其中识别和提取网格模型的几何特征成为必要的步骤. 但现有的多种特

征提取算法对模型特征形状有一定的限制条件, 处理网格分布不规则或含有噪声的复杂模型的效果不好, 易忽略平缓的特征, 不能够满足多个领域的模型编辑和进一步的设计要求. 本文提出一种基于张量投票理论的特征边提取算法, 能够计算给定的三角网格模型的特征边, 并提取模型的特征边框; 在特征

提取的基础上, 可以方便地对模型进行光顺、重网格化、模型表示转换和纹理映射等多种操作. 大量的实例验证结果表明, 本文算法对网格模型的特征提取具有良好的鲁棒性, 能够处理网格分布不均匀, 以及含有狭长三角形或者是孔、缝的网格模型, 处理有噪声的模型也能达到良好的效果, 表现出比现有方法更好的适应性.

1 相关工作

近年来, 为了适应不同应用的需要, 人们提出了很多模型的特征识别方法. 根据对象的不同, 这些方法可以分为 2 类: 第一类是提取将网格模型分割成对应的子几何体的特征^[1-3]. 该方法试图将“自然物体”分割成为有意义的部分, 这种分割希望与人的主观想法一致, 因此其主观性较强, 对精度的要求不高, 常常应用于动画、游戏等领域. 另一类方法致力于 CAD 模型的逆向工程应用^[4]. 这类方法提取特征边并将网格分割成具有几何性质的表面片, 对精度的要求较高. 本文算法根据网格表面上的张量投票性质将网格分割成若干表面片, 因此其属于第二类方法.

根据输入模型的不同, 可以将特征边提取算法分为 2 类: 输入点云数据和输入三角网格模型. 第一类算法首先要估计点处的曲率, 根据估计曲率计算得到特征边^[5-7], 但这类算法采用不同的曲率估计方法得到的结果精确度相差较大. 目前, 点云重构网格的技术已比较成熟, 因此本文采用第二类方法, 基于三角网格模型进行特征边提取.

根据实现方法的不同, 可以将特征边提取算法分为 2 类^[8]. 1) 基于边的方法^[9-10]. 该类算法一般分两步进行: 找出曲率突变点作为边点(也称特征点), 再将这些离散的边点连接成线(也称特征边). 由于这类方法只利用了与边点相邻的数据点, 因此对数据点本身的误差以及计算误差很敏感; 而在算法中如何避免误差影响, 提高特征的提取精度是很重要的, 文献^[9-10]都没有对如何避免误差影响进行研究. 2) 基于面的方法. 该类方法首先从一组“种子面”开始进行“区域增长”, 找出具有相似特征的一组点构成曲面, 直至周围邻域没有特征一致性的点才停止增长; 然后根据相邻曲面相交或其他算法求得曲面边界^[11-12]. 该类算法主要是采用二面角以及顶点处的主曲率作为顶点的特征, 其中二面角阈值的确定一般是凭经验给出的, 不够精细, 从而缺乏普适性.

给实际应用带来了不便; 同时, 由于仅使用了二面角和主曲率的信息, 使得该类算法对于某些复杂几何体表面三角网格的特征, 如含有狭长三角形的特征边, 难以精确地提取出来. 本文的特征边提取算法是在三角网格模型基础上采用基于面的方法实现的, 根据张量投票理论, 采用张量投票矩阵特征值作为顶点的特征, 并根据该特征值分布与顶点几何特征之间的对应关系对顶点进行分类, 能有效地避免误差及噪声影响, 处理复杂几何体表面的三角网格模型.

本文参考并优化了 Kim 等^[13]的方法, 在利用张量投票理论时, 由于 Kim 等^[13]的方法计算张量投票矩阵只考虑了三角形 1-邻域的三角形对其的影响, 因此对于网格分布极其不均匀的模型或者是含有狭长三角形或狭长缝隙的模型得不到有效的结果. 本文依据三角形的邻域三角形面积的分布, 动态地选择 n -邻域三角形进行计算, 有效地避免了网格分布不均匀带来的误差. 另外, 对于一些特殊情况, 采用传统的区域增长的方法往往会出现过度增长的现象; 因此本文提出了一种新的基于权重的区域增长方法, 有效地避免了传统方法的不足, 其对大多数模型都能得到较好的结果.

2 本文算法概述

特征通常是指曲面上具有至少一个较大的主曲率的区域; 特征边是指(极大或极小)主曲率沿相应主曲率方向的极值点的连线(或主曲率沿着相应主曲率方向的方向导数的过零点)^[14]. 本文算法的主要目的是在给定的三角网格模型中识别出特征边, 从而提取该模型的特征边框. 本文算法主要分为 4 步:

Step1. 针对三角网格模型每个顶点求其张量投票矩阵, 解得其对应的特征值.

Step2. 根据特征值将顶点分类(分为面点、边点、角点 3 类).

Step3. 进行区域增长, 得到曲率尽可能平缓的模型区域表面.

Step4. 提取面与面之间的交界线, 得到特征边.

3 基于法向的张量投票

Medioni 等^[15]定义了三角网格顶点 v 上基于法向的张量投票 T_v , Kim 等^[13]利用该张量投票理论进行多维网格的特征识别. 但是, 由于该方法只考虑了每个点 1-邻域三角形对该点位置的影响, 当三角

网格分布极其不均匀,尤其是含有较多狭长或者面积近乎为零的三角形时,通过特征值分析得到的点的位置信息往往是错误的。Page 等^[16-17]采用一种法向投票的方法,它是由张量投票改进得到的。取每个三角形法向在顶点 v 处的对称向量作为投票法向,用测地线距离取代了欧氏距离,把 1-邻域三角形集合扩大到了某个测地距离范围内的所有三角形集合。该方法避免了网格分布不均匀带来的影响,但是反复的求解测地距离导致算法效率较低。本文参考了文献[13, 15]中的张量投票的方法,对获取邻域三角形集合做了改进,根据该点处网格平均面积的大小动态选择邻域集合。

3.1 张量投票矩阵

Medioni 等^[15]定义 T_v 为三角网格顶点 v 处基于法向的张量投票协方差矩阵。 V_{t_i} 表示三角形 t_i 的投票分量,由 \triangle_{t_i} 的法向定义; n_{t_i} 表示三角形 t_i 的法向,设 $n_{t_i} = (a, b, c)$, 则

$$T_v = \sum_{t_i \in S_v^n} \mu_{t_i} V_{t_i},$$

$$V_{t_i} = n_{t_i} \circ n_{t_i}^T = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \circ (a \ b \ c) = \begin{pmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{pmatrix}.$$

其中, S_v^n 表示顶点 v 的 n -邻域内所有三角形面片组成的集合; μ_{t_i} 为 \triangle_{t_i} 的投票分量在张量投票中所占权重。本文在 Medioni 等^[15]定义的权重函数的基础上增加了邻域 \triangle_{t_i} 所处第 m 邻域对该点张量投票矩阵的影响,定义

$$\mu_{t_i} = \left[\frac{\text{area}(t_i)}{\max \text{area}(S_v^n)} \right] \exp \left[-\frac{m \cdot \|c_{t_i} - p_v\|}{\sigma \beta} \right] \quad (1)$$

其中, $\text{area}(t_i)$ 表示 \triangle_{t_i} 的面积, $\max \text{area}(S_v^n)$ 表示在 S_v^n 中最大三角形的面积; c_{t_i} 表示 \triangle_{t_i} 的中心的坐标, p_v 表示点 v 的坐标; σ 表示该邻域包围盒的边长, m 表示 \triangle_{t_i} 位于顶点 v 的第 m 邻域($m \leq n$)。由式(1)可知,三角形的面积越小,三角形距离顶点 v 越远,其在张量投票中占的权重越小。在面积一定的情况下, $\|c_{t_i} - p_v\|$ 越大,权重越小。在求解张量投票

矩阵的过程中,首先是确定 n 的大小,找到顶点 v 的 S_v^n ; 本文通过

$$\sum_{j=1}^n \text{avg}(\text{sub}S_v^j) \leq u \cdot \text{avg}(\text{all}), \quad n \geq 1 \quad (2)$$

来确定 n 。其中, $\text{avg}(\text{all})$ 表示整个网格所有三角形的平均面积, $\text{avg}(\text{sub}S_v^j)$ 表示属于顶点 v 的第 j 邻域的三角形平均面积, $\text{sub}S_v^j$ 指顶点 v 的第 j 邻域中的三角形集合,是 S_v^n 的子集。求得满足式(2)的 n 的最大值,即为要求的 n ; u 为调整参数,当 u 取值较大时,则求得的 n 较大。若式(2)无解,则认为 $n=1$ 。为了处理在不均匀网格的同时不增加计算复杂度,本文取 $u=2.0$ 。

当三角网格模型分布较均匀且取 $u=1.0$ 时,将其代入式(2)计算得到 $n=1$,在式(1)中 $m=1$ 。此时上述计算方法简化为文献[13, 15]中的张量投票矩阵计算方法。

3.2 张量投票矩阵特征值分析

由于张量投票矩阵是一个对称半正定的矩阵,因此可以将其对角化为

$$T_v = \lambda_1 e_1 e_1^T + \lambda_2 e_2 e_2^T + \lambda_3 e_3 e_3^T \quad (3)$$

其中, $\lambda_1, \lambda_2, \lambda_3$ 是 T_v 的特征值; e_1, e_2, e_3 是 T_v 的特征向量。式(3)可以改写成

$$T_v = (\lambda_1 - \lambda_2) e_1 e_1^T + (\lambda_2 - \lambda_3) (e_1 e_1^T + e_2 e_2^T) + \lambda_3 (e_1 e_1^T + e_2 e_2^T + e_3 e_3^T).$$

Medioni 等^[18]定义

$$\begin{cases} S_s = \lambda_1 - \lambda_2 \\ S_c = \lambda_2 - \lambda_3 \\ S_n = \lambda_3 \end{cases} \quad (4)$$

其中, S_s 表示顶点位于面上的特征大小, S_c 表示顶点位于边上的特征大小, S_n 表示顶点位于拐角处的特征大小。因此,式(4)中特征值之间的关系式能够明显表示顶点的位置信息。如图1所示,图中的颜色条由蓝色到红色,表示特征越来越尖锐。可以看出,特征值对于曲率的变化较为敏感,能够很好地反映曲率变化。

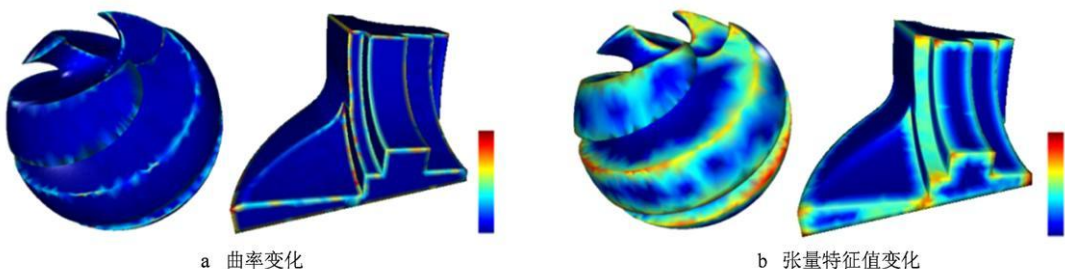


图1 曲率与张量投票特征值所隐含的位置信息

4 顶点分类

Shimizu 等^[9]提出, 根据张量投票矩阵求得的特征值可以将三角网格模型中的点分为面点、边点和角点 3 类. $\lambda_1, \lambda_2, \lambda_3$ 是 T_v 的特征值, 设 $\lambda_1 \geq \lambda_2 \geq \lambda_3$, 则如果 λ_1 相对于 λ_2, λ_3 较大, 同时 λ_2, λ_3 都接近于 0 时, 相应的顶点为面点; 如果 λ_1, λ_2 相对较大, λ_3 接近于 0, 相应的顶点为边点; 如果 $\lambda_1, \lambda_2, \lambda_3$ 值比较接近, 都不为 0 时, 该顶点为角点. 如图 2 所示.

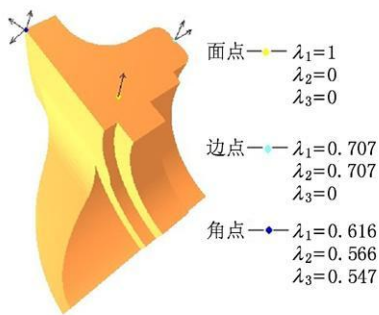


图 2 张量投票特征值分布与顶点几何特征的对应关系

文献 [13] 中采用 K -Means 聚类算法对顶点进行分类, 这样做的优点是能够将网格顶点分成 n 类, 可以考虑模型的颜色、材质等信息. 但是, 采用 K -Means 聚类算法时初始点的选取对于最终的结果影响较大, 很难找到合适的初始点选择算法来保证对于大多数模型都能得到较好的顶点聚类效果. 如图 3 所示, 在图 3 a 中, 平缓的边点聚为一类, 尖锐的边点跟角点聚为一类; 在图 3 b 中, 同一条边上的点可能聚为不同的类; 在图 3 c, 3 d 中, 对“自然物体”模型的处理效果也不理想.

本文并没有采用 K -Means 聚类算法, 而是根据特征值与顶点几何特征的对应关系, 采用简单的阈值控制方法来进行顶点分类, 即根据特征值中含有 0 的个数将所有的点分成面点、边点与角点. 设置阈值 ξ 来控制精度, 当 $\lambda_i \leq \xi$ 时, 则认为 $\lambda_i = 0$. 采用这种方法, ξ 越小, 对曲率的敏感性越强, 在类似圆柱表面的弧面上的点, 即判断为边点; ξ 越大, 对曲率的敏感性越低, 同时具有一定的抗噪性. 该方法简单高效, 并且通过后续断点连接可保证其对大多数模型是有效的.

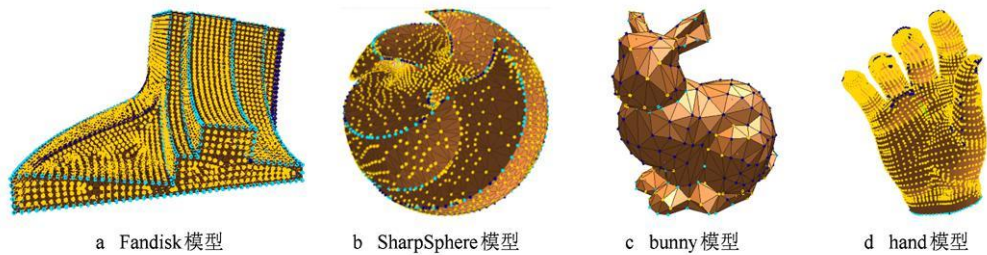


图 3 采用 K -Means 聚类算法得到的结果

由于在某些边上的点的曲率变化较小, 会导致有些边上的点无法识别, 从而出现一些断点. 不论采用 K -Means 聚类算法还是本文的阈值控制方法, 都会产生断点. 本文定义的断点是指其 1-邻域中所有邻接点只有一个为边点或者角点. 当遇到断点时, 则采用下文描述的断点连接策略, 识别出断点附近应该位于边上的点, 将断边继续生长, 直到遇到其他边并将其连接起来.

进行断点连接时, 首先要找到断点 1-邻域中最可能位于边上的点. 由式 (4) 可知, $S_c = \lambda_2 - \lambda_3$ 表示点位于边上的特征值, 其中, λ_2 与 λ_3 为待判断的顶点的特征值; 同时边越平滑, 则认为该边质量越好. 因此, 采用权重函数

$$W_i = \cos(\theta_i) \cdot S_c, \quad i \in V_v^1$$

选择最可能位于边上的点作为边连接的新增长点.

其中 V_v^1 , 指在顶点 v 的 1-邻域处的所有顶点的集合, W_i 是指顶点 v_i 的权重, θ_i 是在 vv_i 与边原始方向 $v_p v$ 之间的夹角 (如图 4 所示), v_p 是指断点 v 的前一个边点. 最终取权重最高的顶点作为增长出去的点, 若该点已经是边点或角点, 则停止增长; 否则, 把新增的边点设为断点, 继续增长, 直至其与边连接. 图 5 a 所示为未进行连接的断点, 图 5 b 所示为断点连接处理后的结果. 图 5 中黄色的点表示面点, 浅蓝色的点表示边点, 深蓝色的点表示角点.

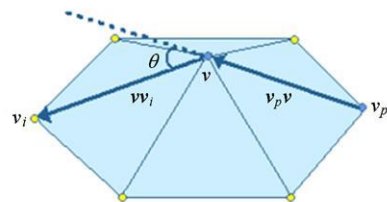


图 4 断点增长时边方向的改变角度

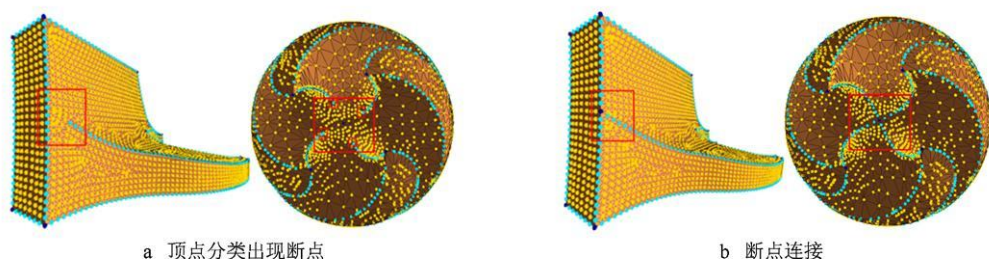


图 5 顶点分类的结果

5 区域增长

Kim 等^[13]借鉴了 Lavoué 等^[20]的方法进行区域增长. 首先选取 3 个顶点类型一致的三角形作为“种子面”, 然后向边相邻的 3 个三角形做扩展测试. 若通过测试, 则将其合并为一个区域; 否则, 停止增长. 该方法不能处理如图 6 所示某些比较特殊的情况. 对图 6a 中红色矩形框中的三角形, 选取 3 个顶点都为边点的三角形作为种子面, 因此会沿着边一直增长产生错误的结果, 出现如图 6b 中的情况. 本文中如未做特殊说明, 则黄色的点表示面点, 浅蓝色的点表示边点, 深蓝色的点表示角点.

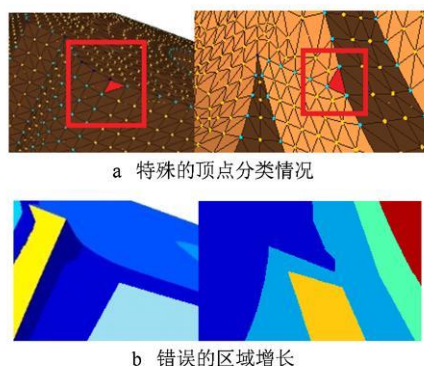


图 6 传统区域增长算法不能处理的特殊情况

本文并未采用上述区域增长算法, 而是采用一种新的基于权重的区域增长算法, 在任意的三角网格模型中, 只存在如图 7 中所示的 10 类三角形和 6 类边. 设置面点、边点、角点的权值分别是 0, 1 和 2, 由此计算得到相应边及相应三角形的权值. 其中, 6 类边及 10 类三角形所对应的权值如图 7 所示, 图中红色数字表示编号, 黑色数字表示权值.

本文进行扩展测试的主要思想如下: 在一般的三角网格模型中, 若每一个三角形不孤立成为一个区域, 则其必定与一个邻居三角形同属一个区域.

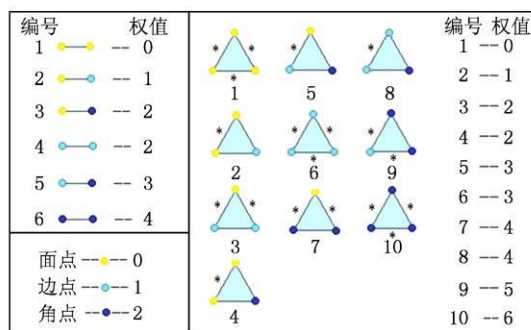


图 7 在三角网格模型中所有点、线、面的分类

图 7 中标记 * 的边指的是在该三角形中权值最小的边. 每个三角形必定与以权值最小的边为公共边的邻居三角形属于同一区域. 当三角形含有 2 条或者 2 条以上权值最小的边时, 则认为该三角形与这 2 个或者 2 个以上的邻居三角形都属于同一区域. 另外, 当三角形的 3 个顶点与初始种子三角形的顶点类型均不一致时, 则此三角形一定不属于该区域.

本文算法首先寻找一个种子三角形, 其选取原则是三角形的 3 个顶点权值相同, 并且未进行过区域扩展. 按照权值从低到高的顺序选取. 如图 7 中的 1, 6 及 10 号三角形都可以作为种子三角形, 按其权值顺序分别选择 1 号、6 号和 10 号三角形. 然后, 再进行区域的扩展测试.

选择好种子三角形后, 对其进行标记; 然后分别向周围的 3 个邻居三角形做扩展测试. 如果通过测试, 则将其并入同一区域. 与种子三角形做同样的标记; 否则, 停止测试. 直到第一个区域扩展结束之后寻找下一个种子三角形; 当不存在候选种子三角形时, 区域扩展测试阶段结束. 注: 若该种子三角形是一个孤立的区域, 则其 3 个点往往是角点, 其不能通过邻域扩展测试从而标记为单独的区域.

根据此区域增长策略有 2 个特例, 即在图 6a 中遇到的 6 号三角形及 8 号三角形. 由于其位于面的边缘处, 因此应该也归入该面三角形增长的区域. 遇

到此类三角形时, 判断的方法是选择与其法向最相近的邻居三角形归入同一区域; 若其邻居三角形中有多于一个三角形与其法向方向一致, 则认为该三

角形与此多个邻居三角形同属于一个区域。
依据此区域增长算法, 能够对大多数模型得到较好的效果, 如图 8 所示。

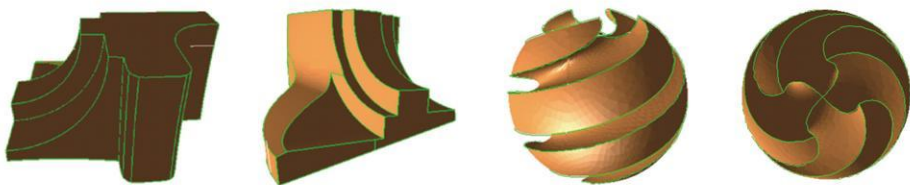


图 8 区域增长后的结果

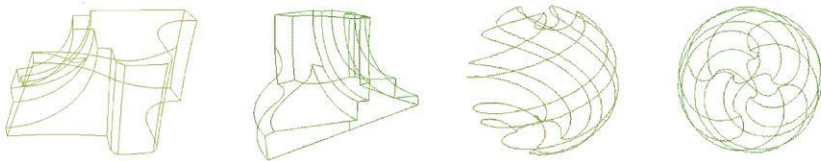
6 提取特征边

在区域增长结束之后, 各个面被明显地标记为

不同的区域, 只需将区域之间的边界提取出来, 即能得到需要的特征边. 图 9 a 所示为由图 8 区域增长结果得到的特征边识别结果, 图 9 b 所示为由此提取的特征边。



a 特征边识别结果



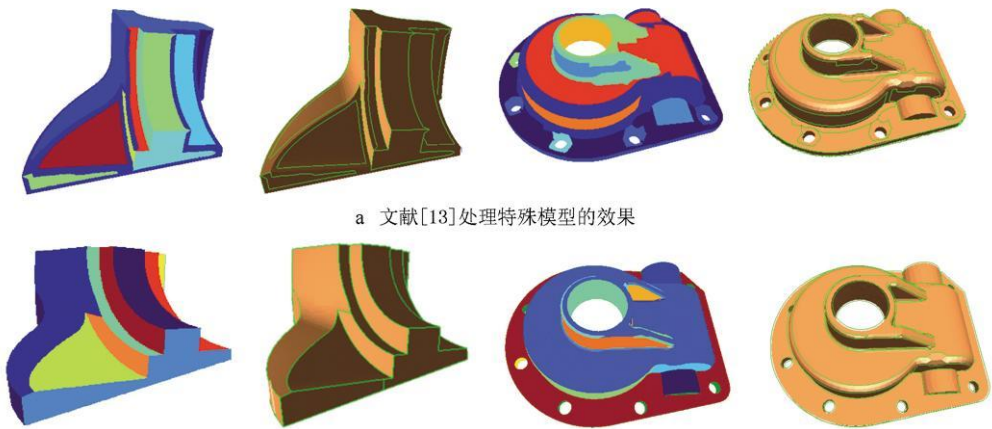
b 提取特征边结果

图 9 提取特征边后的结果

7 实验结果

本文选用的 Fandisk 模型及 Casting 模型经过特征值计算之后, 均出现图 6 所示的特殊三角形, 因

此采用文献[13] 的区域增长方法会出现过度增长, 提取结果不理想, 如图 10 a 所示. 相比而言, 本文算法能够在时间效率更快的情况下得到较好的特征边提取结果, 如图 10 b 所示, 其与文献[13] 时间效率比较如表 1 所示. 由表 1 数据可知, 本文算法在顶点



a 文献[13]处理特殊模型的效果

b 本文算法处理特殊模型的效果

图 10 本文算法与文献[13] 算法的特征提取效果比较

分类这一步的效率远远高于文献[13] 的 K -Means 聚类的效率, 因此总时间效率较高. 运行时间是在一台具有 Intel 酷睿 2.9GHz CPU 和 2GB 内存的 PC 机上进行测量的, 本文对 Fandisk 模型与 Casting 模型进行顶点分类时的阈值 ξ 分别为 0.01 和 0.06; 采用文献[13] 方法对三维模型进行处理, 聚类时将顶点聚为 3 类.

表 1 本文算法与文献[13] 算法的运行时间比较

算法	步骤	Fandisk 模型/s	Casting 模型/s
文献[13] 算法	特征值计算	0. 312	0. 292
	顶点 K -Means 聚类	3. 241	1. 697
	区域增长	0. 075	0. 062
	总时间	3. 628	2. 051
本文算法	特征值计算	0. 481	0. 326
	顶点分类	0. 003	0. 002
	区域增长	0. 094	0. 062
	总时间	0. 578	0. 39

在文献[13] 算法中, 区域合并及清理两步时间占用的权重较小, 而本文方法无需进行这两步, 因此未做这两步算法的时间比较.

本文算法具有更广的适用性, 能够对大多数模型进行特征边的提取, 并且具有一定的抗噪性, 图 11 所示为其对具有噪声的模型的处理结果.

在对于噪声模型的处理中, 可以通过调整阈值 ξ 来得到比较理想的效果, 但是在噪声影响较大的情况下, 往往识别不出比较平缓的特征. 图 11 中阈值 ξ 分别为 0.02, 0.08, 0.2 和 0.4, 红色的矩形框表示未被识别的特征. 由图 11 可以看出, 在噪声较小时的识别效果还是不错的; 但是当噪声增大到 20% 时, 识别效果不是很理想.

采用本文算法对一系列 CAD 模型进行了实验, 表 2 与图 12 分别列举了针对例子模型用本文算法得到的时间结果与特征提取效果. 实验结果表明, 本文算法对网格模型的特征提取具有良好的鲁棒性, 执行效率比较高.

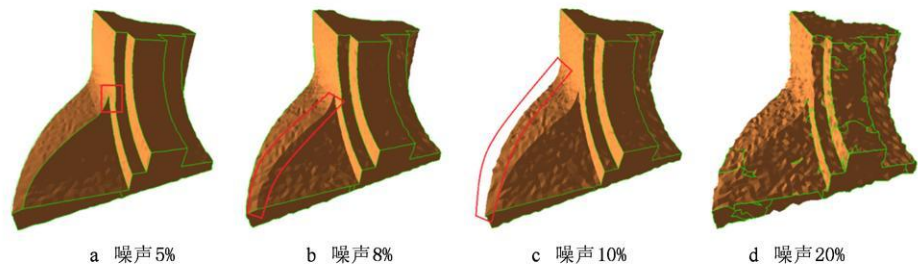


图 11 噪声模型的处理结果

表 2 针对例子模型用本文算法得到的结果

图序号	模型	面片数	ξ 值	特征值计算时间/ms	区域增长时间/ms	特征边提取时间/ms	总运行时间/ms
图 9c, 9d	SharpSphere	18 864	0. 08	749	109	62	920
图 12a	Gear	5 110	0. 01	218	47	78	343
图 12b	Alternator	4 448	0. 01	265	62	212	539
图 12c	Coolingup	2 541	0. 01	172	234	188	594
图 12d	Coolingdown	1 741	0. 01	78	16	47	141
图 12e	Body	768	0. 01	47	15	31	93

本文算法处理 CAD 模型的效果较好, 但是处理“自然物体”模型的效果还有待提高. 如果模型中不含有较细长的部分, 其效果相对好一些. 如图 13 处理时阈值 ξ 分别为 0.08, 0.02 和 0.09. 图 13a 所示的 bunny 模型面片数较多(共有 7461 个面片), 在特征处面片的曲率变化不大, 因此本文算法误将某些局部的噪声识别为特征, 如图中耳朵及前腿处. 图 13b 所示的 bunny 模型为简化后的模型(共有

704 个面片), 并且面片的分布均匀, 因此识别效果还可以. 对于另外一类“自然物体”模型, 如四肢动物以及其他一些局部含有细长部分的模型, 本文算法的处理效果不理想. 如图 13c 所示, 由于在手指处横截面积相对整个模型来说较小, 其上面的点即使未处于特征处, 曲率变化可能比模型其他部位特征上点的曲率变化大. 因此, 本文算法会忽略掉模型较平缓部位的特征, 若过于关注细长部位的变化, 则特征

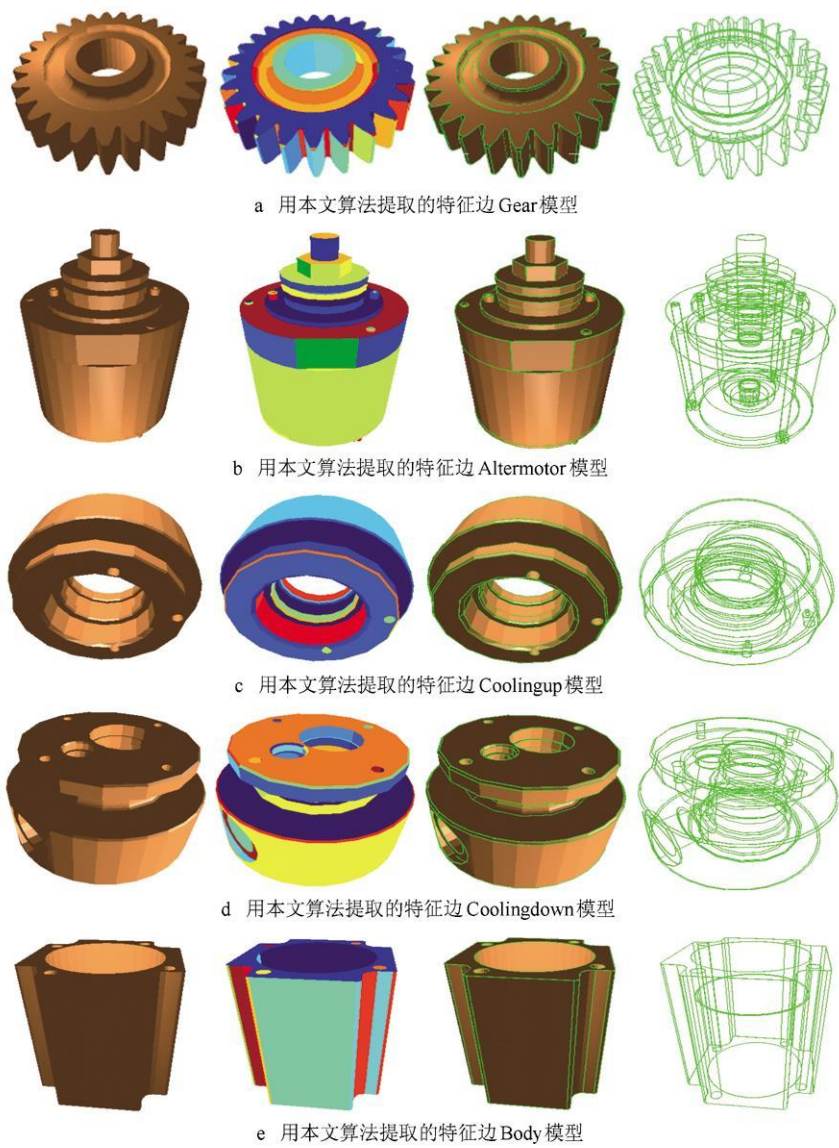


图 12 部分模型的实验结果



图 13 部分“自然物体”模型的实验结果

提取的结果不理想. Kim 等^[13]的方法能够处理多维的网格模型, 采用 K -Means 聚类算法分类可以将其分为 n 类, 因此能够识别平缓部位的特征, 处理“自然物体”模型的效果比本文算法好. 但是, 若在 K -Means 聚类时初始点选取不合适, 或者是区域增长时出现过度增长现象, 则特征提取效果不如本文算法.

8 结 论

本文提出了一种基于张量投票的特征边提取算法, 通过引入张量投票的度量快速地对顶点进行分类, 并通过区域增长的边界线找到特征边, 该算法

能够使用于网格分布不均匀、有噪声、有空洞或者边界不连接的网格模型.但是,本文算法仅仅对处理 CAD 模型效果较好,对于“自然物体”模型(如兔子、马等)提取特征边的效果不够理想.我们今后的研究重点是针对更加广泛的模型进行特征边提取.

致谢 感谢本实验室朱利丰、李声初及高雷雷同学对本文提出的建设性意见.文中所有的模型由 AIM @SHAPE 模型库提供,在此表示感谢!

参考文献(References):

- [1] Reniers D, Telea A. Hierarchical part-type segmentation using voxel-based curve skeletons [J]. *The Visual Computer*, 2008, 24(6): 383—395
- [2] Lai Y K, Hu S M, Martin R R, *et al.* Fast mesh segmentation using random walks [C] // *Proceedings of ACM Symposium on Solid and Physical Modeling*. New York: ACM Press, 2008: 183—191
- [3] Mortara M, Patané G, Spagnuolo M. From geometric to semantic human body models [J]. *Computers & Graphics*, 2006, 30(2): 185—196
- [4] Attene M, Falcidieno B, Spagnuolo M. Hierarchical mesh segmentation based on fitting primitives [J]. *The Visual Computer*, 2006, 22(3): 181—193
- [5] Tong W S, Tang C K. Robust estimation of adaptive tensors of curvature by tensor voting [J]. *IEEE Transactions on Pattern Analysis, Machine, Intelligence*, 2005, 27(3): 434—449
- [6] Yang P H, Qian X P. Direct computing of surface curvatures for point-set surfaces [C] // *Proceedings of IEEE/Eurographics Symposium on Point-Based Graphics*. Aire-la-Ville: Eurographics Association Press, 2007: 29—36
- [7] Mérigot Q, Ovsjanikov M, Guibas L. Robust Voronoi-based curvature and feature estimation [C] // *Proceedings of ACM Symposium on Solid and Physical Modeling*. New York: ACM Press, 2009: 1—12
- [8] Liu Shenglan, Zhou Rurong, Zhang Liyan. Feature line extraction from triangular mesh model [J]. *Journal of Computer-Aided Design & Computer Graphics*, 2003, 15(4): 444—453 (in Chinese)
(刘胜兰, 周儒荣, 张丽艳. 三角网格模型的特征线提取 [J]. *计算机辅助设计与图形学学报*, 2003, 15(4): 444—453)
- [9] Milroy M J, Bradley C, Vickers G W. Segmentation of a wrap-around model using an active contour [J]. *Computer-Aided Design*, 1997, 29(4): 299—320
- [10] Yang M, Lee E. Segmentation of measured point data using a parametric quadric surface approximation [J]. *Computer-Aided Design*, 1999, 31(7): 449—457
- [11] Sapidis N S, Besl P J. Direct construction of polynomial surfaces from dense range images through region growing [J]. *ACM Transactions on Graphics*, 1995, 14(2): 171—200
- [12] Chen Y H, Liu C Y. Quadric surface extraction using genetic algorithms [J]. *Computer-Aided Design*, 1999, 31(2): 101—110
- [13] Kim H S, Choi H K, Lee K H. Feature detection of triangular meshes based on tensor voting theory [J]. *Computer-Aided Design*, 2009, 41(1): 47—58
- [14] Hu Shimin, Yang Yongliang, Lai Yukun. Research progress of digital geometry processing [J]. *Chinese Journal of Computers*, 2009, 32(8): 1—18 (in Chinese)
(胡事民, 杨永亮, 来煜坤. 数字几何处理研究进展 [J]. *计算机学报*, 2009, 32(8): 1—18)
- [15] Medioni G, Tang C K, Lee Mi-Suen. Tensor voting: theory and applications [OL]. [2010-07-08]. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.9125>
- [16] Page D L, Koschan A, Sun Y, *et al.* Robust crease detection and curvature estimation of piece wise smooth surfaces from triangle mesh approximations using normal voting [C] // *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington D C: IEEE Computer Society Press, 2001, (1): 162—167
- [17] 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*. Piscataway: IEEE Press, 2002: 825—828
- [18] Medioni G, Lee M S, Tang C K. A computational framework for segmentation and grouping [M]. Amsterdam: Elsevier, 2000
- [19] Shimizu T, Date H, Kanai S, *et al.* A new bilateral mesh smoothing method by recognizing features [C] // *Proceedings of the 9th International Conference on Computer Aided Design and Computer Graphics*. Washington D C: IEEE Computer Society Press, 2005: 281—286
- [20] Lavoué G, Dupont F, Baskurt A. A new CAD mesh segmentation method, based on curvature tensor analysis [J]. *Computer-Aided Design*, 2007, 37(1): 975—987