

支持 STL 数据源的网格曲面动态空间索引

郭洪帅¹, 孙殿柱^{1*}, 李延瑞², 李 聪¹

(1. 山东理工大学 机械工程学院, 山东 淄博 255049; 2. 西安交通大学 机械工程学院, 西安 710049)

(* 通信作者电子邮箱 dianzhus@sdut.edu.cn)

摘 要:针对 STL 文件格式存在网格顶点数据冗余以及缺乏面片邻接信息等缺陷,提出一种基于多维动态空间索引的显式曲面拓扑重建算法,在消除网格顶点数据冗余的过程中逐步构建网格曲面顶点的 KD 树,通过该索引提高顶点数据冗余消除效率,并基于 KD 树叶节点层数据结构的开放性融入半边数据结构,实现曲面拓扑结构的快速重建。最后,对 6 个不同规模的数据模型进行实验:与采用 R*-Tree、数组、散列表作为索引等方法相比,所提出的 KD 树与半边结构融合的动态空间索引在处理近百万面片的数据文件时,去除冗余顶点用时 11.93 s,拓扑重建仅仅需要 2.87 s,大大减少了冗余顶点的去除时间和拓扑重建时间,并且有效支持网格曲面拓扑邻域信息的快速查询,查询时间在 1 ms 之内,远小于对比算法所用时间。实验结果表明:所提算法能够提高网格曲面冗余顶点去除效率和拓扑重建效率,实现网格曲面拓扑邻域信息的快速查询。

关键词: STL 文件格式; K 维树; 半边结构; 曲面拓扑重建; k -近邻查询

中图分类号: TP391.72 **文献标志码:** A

Dynamic spatial index of mesh surface for supporting STL data source

GUO Hongshuai¹, SUN Dianzhu^{1*}, LI Yanrui², LI Cong¹

(1. School of Mechanical Engineering, Shandong University of Technology, Zibo Shandong 255049, China;

2. School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an Shaanxi 710049, China)

Abstract: Focusing on the issue that there exist defects of vertex data redundancy and lack of adjacency information among the faces in the STereo Lithography (STL) file format, an explicit algorithm of surface topology reconstruction was presented based on the multi-dimensional dynamic spatial index. During the process of eliminating the copies of the mesh vertex data, the K-Dimensional Tree (KD Tree) of the vertices on the mesh surface was gradually built. The efficiency of eliminating the vertex copies was improved by the index and the surface topology was rapidly built based on the storage openness of the data in the leaf node layer of KD-tree, in which the half-edge data structure could be integrated. Finally, compared with methods using R*-Tree, array and hash table as index, the proposed dynamic spatial index integrated KD-Tree with half-edge data structure used 11.93 s to remove redundant vertices and 2.87 s to reconstruct surface topology when dealing with the data file of nearly one million faces, which significantly reduced the time of eliminating the redundant vertices and surface topology reconstruction. And the index effectively supported quick query of the topology information of mesh surface with the query time in 1 ms, which was far less than the comparison algorithms. The experimental results show that the proposed algorithm can improve the efficiency of eliminating the vertex data redundancy and the topological reconstruction as well as achieve quick query of the topology information of mesh surface.

Key words: STereo Lithography (STL) file format; K-Dimensional Tree (KD Tree); half-edge data structure; surface topology reconstruction; k -neighborhood query

0 引言

STL(STereo Lithography)文件格式是 3D Systems 公司为立体光固化成型计算机辅助设计(Computer Aided Design, CAD)系统开发的一种以三角网格结构^[1]记录三维空间曲面信息的文件格式,因其数据格式简单和良好的跨平台性,现已被广泛应用于快速成型、CAD/计算机辅助制造(Computer Aided Manufacturing, CAM)、虚拟现实等领域^[2]。

但是在实际应用中,STL 文件格式存在着网格顶点数据

冗余以及缺乏面片邻接信息等固有缺陷^[3-4]。随着 3D 打印技术的逐渐普及,这些缺陷所带来的负面影响日益突出。现有的解决方法主要是采用顶点数据信息的引用来代替网格顶点的数据冗余,然后基于 STL 文件中的三角面片信息显式地重建曲面拓扑^[5-9]。文献[5]采用二叉树结构进行顶点聚合,该算法可快速查询面片信息,实现拓扑重建;但二叉树的高度会因数据量增大而急剧增加,导致调整树的平衡付出时间代价过大。文献[6]采用红黑树作为索引结构,可有效去除 STL 文件中的冗余信息;但红黑树建树复杂,为了检测树

收稿日期:2015-04-23;修回日期:2015-06-02。 基金项目:国家自然科学基金资助项目(51075247)。

作者简介:郭洪帅(1989-),男,山东潍坊人,硕士研究生,主要研究方向:逆向工程、点云处理; 孙殿柱(1956-),男,山东烟台人,教授,博士,主要研究方向:CAD/CAM、逆向工程; 李延瑞(1979-),男,山东郯城人,博士研究生,主要研究方向:三维数据处理、曲面重建; 李聪(1985-),男,山东菏泽人,硕士,主要研究方向:逆向工程、曲面重建。

的平衡性需要进行节点颜色的改变和树的旋转。文献[7]为 STL 文件记录的三角面片集合构建动态空间索引,利用索引的空间查询功能获取曲面局部拓扑信息,该方法的定位查询效率显著优于显式的曲面拓扑重建方法;但是在曲面局部拓扑信息获取效率与存储利用率等方面低于后者。文献[8]通过数组存储顶点信息;但是数组无法动态存储,插入和删除操作效率低。文献[9]以哈希表作为数据结构,哈希表在查询特定键值的数据效率非常高;但其最大难点是散列函数的设定,设置不当会使查询效率下降,并且哈希表对于不同键值的查找容易误判,导致查询的准确度降低。

本文提出一种基于多维动态空间索引的显式曲面拓扑重建算法,通过建立顶点数据的索引结构消除 STL 文件中冗余的顶点数据,并基于索引结构的开放性融入半边结构,实现网格曲面的快速拓扑重建。在保证曲面网格冗余顶点去除效率与拓扑结构重建效率的前提下,实现网格曲面拓扑邻域信息的快速查询。

1 网格曲面顶点数据复本的消除

1.1 STL 文件描述

STL 文件格式分为美国信息标准交换码 (American Standard Code for Information Interchange, ASCII) 与二进制格式两种形式,其中前者具有自解释性,后者存储效率较高且易于解析,但无论是 ASCII 还是二进制形式的 STL 文件,本质上均表达为图 1 所示的面片列表结构。

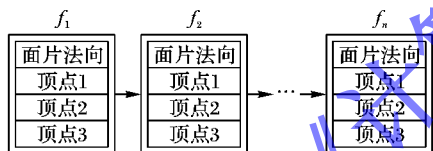


图1 STL三角面片列表

在 STL 数据中存在大量具有邻接关系的面片,若三角面片 f_a 与 f_b 邻接,则 f_a 数据区段中必定有两个顶点(如图2中 p, q 两点)在 f_b 的数据区段中存在复本。

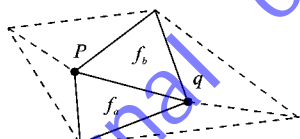


图2 相邻面片的冗余顶点

网格中三角面片的理想状态是等边三角形,即对于每一个顶点,理想状态下会被6个三角形共用,因此在文件中会产生5/6的冗余顶点。由此可见,顶点数据的复本是造成 STL 文件信息冗余的主要因素。因此在对 STL 数据进行曲面拓扑重建之前,通过构建网格顶点集合,以顶点数据的引用代替数据复本,消除 STL 文件中的数据冗余,进而提高拓扑重建效率。

1.2 KD 树建立与冗余顶点消除

对于含有 n 个三角面片数据区段的 STL 文件,设网格曲面顶点集合为 V ,其一般构建过程为:解析 STL 文件中的三角面片数据区段,所得顶点数据若非 V 中某个顶点的复本,则将其添加至 V 中。由于这一过程需对 V 中网格顶点数据进行线性遍历,导致 V 构建过程的时间复杂度为 $O(n^2)$,效率较低。

为解决这一问题,现有方法主要是借助散列表、高度平衡树 (Adelson-Velsky and Landis' tree, AVL tree)、红黑树等一维动态索引将顶点复本搜索范围缩小为 V 的子集,并以该子集进行复本判断。

顶点数据的复本判断本质上是三维空间中的点查询运算,虽然一维索引可辅助改善查询效率,但其数据结构容易在三维空间中的某一方向发生恶化而使查询效率降低。与一维索引相比,KD 树、R 树及变体等多维动态空间索引可为空间点查询运算提供更为稳健的支持^[10-11]。

本文采用基于 k -均值聚类的 KD 树^[12] 作为网格顶点索引结构,并基于文献[12]提出的目标点的 k -近邻查询算法实现顶点数据复本的判断,从而将网格顶点集合 V 的构建问题转化为构建网格顶点索引 U ,具体过程如下(设 STL 文件中三角面片数为 n):

- 1) $i \leftarrow 1, U \leftarrow \emptyset$ (其中 $i = 1, 2, \dots, n, \emptyset$ 表示空集,即将 U 置为空集)。
- 2) 解析 STL 文件中的第 i 个面片的数据区段,获得面片 f_i 三个顶点 v_1, v_2, v_3 。
- 3) $j \leftarrow 1$ (其中 $j = 1, 2, 3$)。
- 4) 应用 KD 树的 k -近邻查询算法,在 U 中查询 v_j 的最近邻点 v'_j ,若 $v_j \neq v'_j$,则用 KD 树的构建算法^[12] 将 v_j 插入至 U 中;否则,直接转 5)。
- 5) $j \leftarrow j + 1$ 。
- 6) 重复 4) 与 5),直至 $j > 3$ 。
- 7) $i \leftarrow i + 1$ 。
- 8) 重复 2) 至 7),直至 $i > n$ 。

当上述过程结束后, U 的叶节点层所存储的数据即为去除顶点数据复本后的网格曲面顶点集合。由于后续曲面拓扑重建需要将网格曲面顶点与其他几何元素建立联系,需使 KD 树叶节点的数据结构保持一定的开放性,即允许向其中添加其他类型的数据对象。

2 曲面拓扑的半边结构表示

基于 STL 文件记录的三角面片信息重建曲面拓扑,本质上是建立网格曲面的各类基本几何元素之间的关系。常用的数据结构有邻接表^[13]、翼边结构^[14]等,本文采用比翼边结构更加完善的半边结构^[15]表示曲面拓扑,并建立顶点、半边以及面片等几何元素之间的关系。

半边结构以有向线段的形式记录网格曲面中各个面片及其邻接面片的信息,对于网格曲面中的任一面片 f (其顶点为 a, b 与 c),设面片 f' 为 f 的一个邻接面片,则 f' 与 f 的邻接边界是端点为 a 与 b 的线段,该线段在两个面片中体现为方向相反但长度相同的半边 ab 与 ba ,将 ab (或 ba) 称为 ba (或 ab) 的对偶半边,如图3所示。若在半边数据结构中完备的记录上述信息,则至少需包含4个数据域:1) 半边终点(或起点) e ;2) 下一条半边 h_n ;3) 所属面片 f ;4) 对偶半边 h_c ;因此任意一条半边 h 均可表示为四元组 (e, h_n, f, h_c) 。

虽然一个面片由3条半边构成,但是由于半边结构中已经记录了这些半边的首尾顺次连接关系,因此面片的数据结构只需记录其中任意一条半边 h 即可。鉴于 STL 文件中记录了每个面片的法向信息 n ,因此半边结构的面片 f 可表示为

(h, n) 。

为基于任一顶点快速获取所属的半边集合,需在网格顶点的数据结构中至少包含一条以该顶点为起点的半边——出射半边^[16],但是这一数据结构仅适用于具有完备曲面拓扑信息的半边结构。由于 STL 文件中缺乏面片的邻接关系,为便于重建这一关系,可在网格顶点的数据结构中包含该顶点所有的出射半边信息,即:对于半边 h 的终点所对应的网格顶点 e ,若 I 为 e 的出射半边集合,利用网格曲面顶点 KD 树的叶节点数据结构的存储开放性,将 I 的信息存储于 e 在 KD 树中对应的叶节点数据结构中。

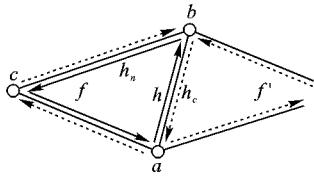


图3 半边结构

3 基于 STL 文件的曲面拓扑重建

3.1 基于索引建立半边结构表示的三角面片

基于曲面网格顶点 KD 树,上文所述的半边、顶点与面片之间的关系可在 STL 文件面片数据区段解析过程中逐步建立,步骤如下(其中 H 表示半边集合, F 表示三角面片集合):

- 1) $i \leftarrow 1, H \leftarrow \emptyset, F \leftarrow \emptyset$ (将 H, F 置为空集);
- 2) 解析 STL 文件中的第 i 个面片的数据区段,获得面片法向 n_i 与顶点数据 v_{i1}, v_{i2}, v_{i3} ;
- 3) 构造半边数据结构 h_{i1}, h_{i2}, h_{i3} 以及面片数据结构 f_i ,并将其数据域均初始化为空;
- 4) $j \leftarrow 1$;
- 5) $k \leftarrow (j + 1) \bmod 3$;
- 6) 应用 KD 树的 k -近邻查询算法,在网格曲面顶点 KD 树 U 中获取 v_{ik} 的最近邻点所在的叶节点 $U(v_{ij})$;
- 7) 将 $U(v_{ij})$ 作为 h_{ij} 的终点,并将 h_{ik} 作为 h_{ij} 的下一条半边;
- 8) 将 h_{ik} 作为 v_{ij} 的一条出射半边添加至 $U(v_{ij})$ 的半边集合 H 中;
- 9) $j \leftarrow j + 1$;
- 10) 重复 5) ~ 9), 直至 $j > 3$;
- 11) 将 h_{i1} 作为 f_i 的起始半边,将 n_i 存入 f_i 对应的法向信息的数据区域,然后将 f_i 添加至 F 中;
- 12) 将 f_i 作为 h_{i1}, h_{i2} 与 h_{i3} 的所属面片,然后将 h_{i1}, h_{i2} 与 h_{i3} 逐次添加至 H 中;
- 13) $i \leftarrow i + 1$;
- 14) 重复 2) ~ 13), 直至 $i > n$ 。

上述过程完成了 STL 文件中各个三角面片的拓扑重建,即建立了面片集合 F 与半边集合 H ,且在 KD 树叶节点层中为每个顶点的数据结构添加了该顶点的出射半边信息。此时,半边集合 H 中所有的半边尚且缺乏对偶半边信息。

3.2 匹配对偶半边,重建三角网格拓扑结构

为每条半边寻找对偶半边的过程本质上是面片邻接关系的重建过程。对偶半边的判断规则为:对于任一半边 $h(e, h_e, f, h_e)$,若其对偶半边 h_e 存在,则必定存在于 e 的出射半边集合

I 中,且 h_e 的终点对应的出射半边集合中也必包含 h 。设半边集合 H 中总共包含 m 条半边,则为每条半边搜索对偶半边的过程如下:

- 1) $i \leftarrow 1$;
- 2) 对于 H 中第 i 条半边 h_i ,在其终点的出射半边集合 I 中按上述判断规则搜索 h_i 的对偶半边 h' ;
- 3) 若 h' 存在,将其信息记录于 h_i 的对偶半边数据域,并从 h_i 终点的半边集合中删除 h' ;
- 4) $i \leftarrow i + 1$;
- 5) 重复 2) ~ 4), 直至 $i > m$ 。

至此完成半边结构形式的曲面拓扑重建的整个过程, KD 树的叶节点记录了全部的网格曲面顶点数据,同时也记录了该顶点一条出射半边的信息。在半边结构中,每条半边的终点均为 KD 树叶节点层存储的网格曲面顶点的引用,最终所得网格曲面顶点的 KD 树与半边结构之间的关系如图 4 示。

KD 树

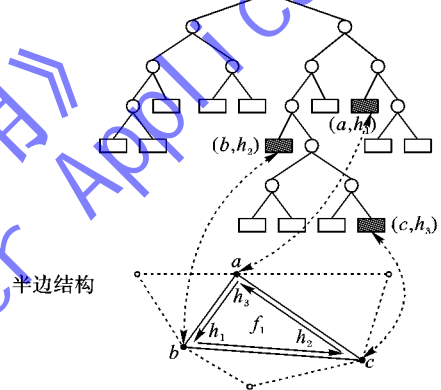


图4 网格曲面的动态空间索引结构

4 算法分析与应用实例

为验证本文算法的有效性,对图 5 中 6 个不同数据规模的三角网格曲面模型对应的 STL 文件分别应用文献[7]、文献[8]、文献[9]与本文算法进行顶点数据冗余消除、网格曲面拓扑重建和网格曲面拓扑邻域查询实验并比较所用时间,在硬件配置为 HP xw8600 Workstation (Inter Xeon CPU, 2.50 GHz, 4.0 GB 内存),操作系统为 GNU/Linux 的测试环境中进行实验。

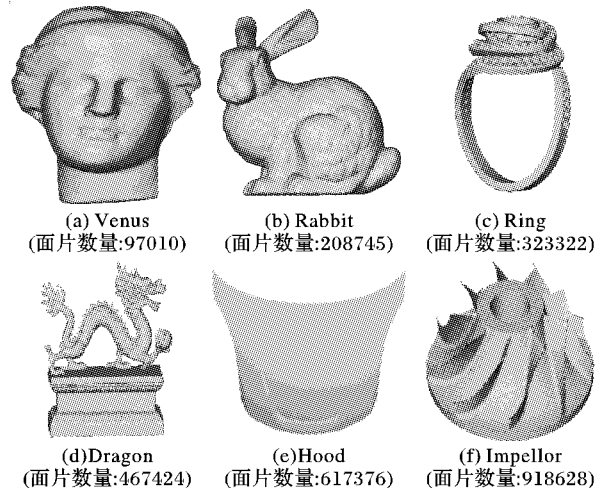


图5 网格曲面模型

4.1 数据复本消除时间分析与比较

在此过程中构建网格顶点的KD树,建树时间复杂度为 $O(n \log n)$,空间复杂度为 $O(n)$,最近邻查询的时间复杂度为 $O(\log n)$,其中 n 为网格顶点的个数。从表1中可以看出,相比消除顶点数据复本而对所有网格顶点进行最近邻查询所用的时间,建树时间要小2个数量级,即便是数据量近100万面片的数据文件,建树时间大约是0.2 s,几乎可以忽略。

表1 KD树建树时间与最近邻查询时间对比

网格曲面模型	建树时间/s	查询时间/s
Venus	0.016	1.263
Rabbit	0.035	2.879
Ring	0.054	3.983
Dragon	0.087	6.331
Hood	0.117	8.682
Impellor	0.196	11.730

不同算法在网格曲面顶点数据复本的消除过程中所用时间如图6所示。本文算法在顶点数据复本判断过程中采用三维动态空间索引,利用KD树高效的查询性能对STL文件中所有顶点进行最近邻查询运算,冗余顶点去除的总时间为建树时间与查询时间的总和。从图6中可以看出,本文采用KD树去除冗余顶点,相比其他几种算法,效率得到显著提高。

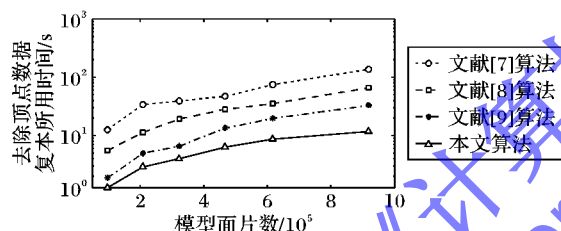


图6 顶点数据复本消除时间比较

4.2 拓扑重建时间比较

在曲面拓扑重建过程中,时间复杂度为 $O(m)$,其中 m 为半边集合 H 中的半边条数。通过数据复本消除过程中所构建的网格索引,将半边结构中的出射半边信息加入到索引结构的叶节点中,利用KD树的 k -近邻算法进行对偶半边的信息快速查询,提高邻接面片匹配的效率和实现网格曲面的快速拓扑重建。从图7中可看出,本文算法在处理90多万面片的模型时,拓扑重建仅仅需要2 s多的时间。

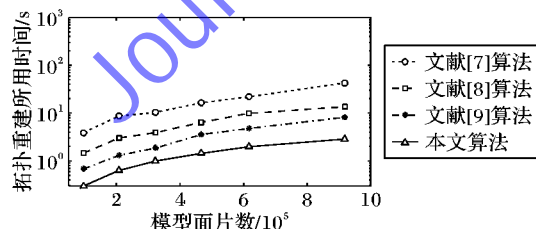


图7 曲面拓扑重建时间比较

以上两个对比实验表明,本文算法在冗余顶点数据消除与网格曲面拓扑重建的方面效率非常高。但对于一个网格模型的拓扑重建,多数情况下只需重建一次,后期常常需要对网格进行去噪、平滑、精简等工作,因此网格曲面拓扑邻域查询会对后续工作起到非常关键的作用,查询效率的高低在一定程度上将决定算法的优劣。

4.3 拓扑邻域查询时间比较

为比较不同算法的拓扑邻域查询性能,基于不同索引结构对图5所示网格曲面上任意一点进行1-环域查询,即查询围绕该顶点的所有三角面片信息,如图8所示。在不同网格曲面上随机采集样点进行1-环域查询操作,记录查询时间。为避免查询时间的偶然性,将这一过程重复1000次,计算平均查询时间,并比较不同算法在拓扑邻域查询所用的时间,用时越少,查询性能越好。

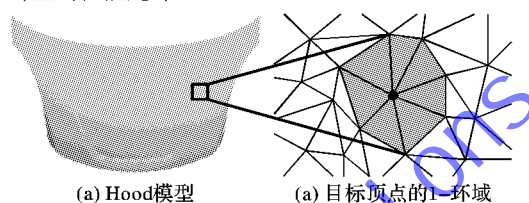


图8 网格曲面上任意一点的1-环域查询

在1-环域查询过程中,空间索引叶节点存储网格曲面顶点和半边信息,通过KD树快速查询顶点信息,利用半边结构的半边环序及对偶半边信息,直接定位对偶半边所指向的面片信息,并依据半边环序,顺次查找下一条半边所指向的面片信息,直至半边环序遍历结束,即可得到该点的1-环域。本文算法采用半边结构与KD树相融合的索引结构,充分利用KD树查询速度快与半边结构存储邻域面片信息的优势,使拓扑邻域查询效率大大提高。不同算法1-环域查询时间比较结果如表2所示。

表2 网格曲面上随机点的1-环域查询时间比较 ms

STL文件	文献[7]算法	文献[8]算法	文献[9]算法	本文算法
Venus	10.23	15.62	5.62	0.559
Rabbit	19.53	26.55	11.86	0.645
Ring	30.18	36.81	15.64	0.781
Dragon	45.21	59.47	23.72	0.842
Hood	59.68	89.38	33.87	0.872
Impellor	102.35	96.13	57.19	0.883

5 结语

1) 在基于STL文件的网格曲面顶点数据复本消除环节,本文算法采用支持多维数据查询的动态空间索引进行网格曲面顶点数据复本判断,冗余顶点去除效率显著提升。

2) 在曲面拓扑重建过程中,保持网格曲面顶点KD树叶节点层数据存储的开放性,在网格曲面顶点的数据结构中加入半边结构并记录所有以该顶点为起点的半边信息,提高了半边面片邻接关系的重建效率。

3) 半边结构与KD树相融合所得网格曲面动态空间索引,在保证曲面拓扑重建效率的前提下,有效支持网格曲面拓扑邻域信息的快速查询。

4) 所构建的网格动态索引仅支持流形面的拓扑重建,具有一定的局限性,后续研究可以将本文算法扩展为支持非流形面的拓扑重建,提高算法的普适性。

参考文献:

- [1] ZHANG Y, GE L. Improved moving least squares algorithm for directed projecting onto point clouds [J]. Measurement, 2011, 44 (10): 2008-2019.

- [2] CHUA C K, LEON K F, LIM C S. Rapid prototyping: principles and applications [M]. River Edge: World Scientific, 2010: 25 – 34.
- [3] DAI X, CUI H-G, ZHANG L-P, *et al.* Hexahedral mesh generation for geometry with multi-featured constraints [M]// HUANG D-S, BEVILACQUA V, FIGUEROA J C. Intelligent Computing Theories. LNCS 7995. Berlin: Springer, 2013: 612 – 620.
- [4] HU C, LI Y, ZHANG Y. Research on repair algorithms for hole and cracks errors of STL models [M]// Computing and Intelligent Systems. Berlin: Springer, 2011: 42 – 47.
- [5] YANG S, DU Y, SHU S. Reconstruction algorithm of surface meshes based on STL files [J]. Computer Engineering, 2011, 37(4): 10 – 11. (杨晟院, 杜亚娟, 舒适. 基于 STL 文件的曲面网格重建算法[J]. 计算机工程, 2011, 37(4): 10 – 11.)
- [6] AN T, DAI N, LIAO W, *et al.* An efficient algorithm for topological reconstruction of STL data [J]. Mechanical Science and Technology for Aerospace Engineering, 2008, 27(8): 1031 – 1034. (安涛, 戴宁, 廖文和, 等. 基于红黑树的 STL 数据快速拓扑重建算法[J]. 机械科学与技术, 2008, 27(8): 1031 – 1034.)
- [7] SUN D, LI X, LI Y, *et al.* Research on high precision and fast generation algorithm of NC tool path for triangular mesh surface [J]. Transactions of the Chinese Society for Agricultural Machinery, 2010, 41(7): 223 – 226. (孙殿柱, 李心成, 李延瑞, 等. 三角网格曲面高精度刀轨快速生成算法[J]. 农业机械学报, 2010, 41(7): 223 – 226.)
- [8] HOU B, LIU X. The rapid topological reconstruction of 3D-solid and simplification of weighted QEM [C]// Proceedings of the 2013 IEEE International Conference on Vehicular Electronics and Safety. Piscataway: IEEE, 2013: 273 – 277.
- [9] WANG Z. Fast topological reconstruction algorithm for a STL file [J]. Journal of Computer Applications, 2014, 34(9): 2720 – 2724. (王增波. STL 格式文件的快速拓扑重建算法[J]. 计算机应用, 2014, 34(9): 2720 – 2724.)
- [10] GAEDE V, GUNTHER O. Multidimensional access methods [J]. ACM Computing Surveys, 1998, 30(2): 170 – 231.
- [11] OOI B C, McDONELL K J, SACKS-DAVIS R. Spatial kd-tree: an indexing mechanism for spatial databases [C]// Proceedings of the 1987 11th Annual International Computer Software and Applications Conference. Piscataway: IEEE, 1987: 85.
- [12] MUJA M, LOWE D. Scalable nearest neighbour algorithms for high dimensional data [EB/OL]. [2015-01-07]. http://www.cs.ubc.ca/research/flann/uploads/FLANN/flann_pami2014.pdf.
- [13] YAN W, WU W. Data structure [M]. Beijing: Tsinghua University Press, 2002: 163 – 164. (严蔚敏, 吴伟民. 数据结构[M]. 北京: 清华大学出版社, 2002: 163 – 164.)
- [14] MAHDIRAJI A R, BAUMANN P. Database support for unstructured meshes [J]. Proceedings of the VLDB Endowment, 2013, 6(12): 1404 – 1409.
- [15] PARIS R. Modified half-edge data structure and its applications to 3D mesh generation for complex tube networks [D]. Louisville: University of Louisville, 2013.
- [16] BRONNIMANN H. Designing and implementing a general purpose half-edge data structure [M]// BRODAL G S, FRIGIONI D, MARCHETTI-SPACCAMELA A. Algorithm Engineering, LNCS 2141. Berlin: Springer, 2001: 51 – 66.

(上接第 2564 页)

参考文献:

- [1] ZHANG Q, QU Y. Topic-opposite sentiment mining model for online review analysis [J]. Journal of Frontiers of Computer Science and Technology, 2013, 7(7): 620 – 629. (张倩, 瞿有利. 用于网络评论分析的主题-对立情感挖掘模型[J]. 计算机科学与探索, 2013, 7(7): 620 – 629.)
- [2] MOGHADDAM S, ESTER M. Aspect-based opinion mining from online reviews [EB/OL]. [2015-02-02]. <http://www.docin.com/p-543993078.html>.
- [3] MOGHADDAM S, ESTER M. The FLDA model for aspect-based opinion mining: addressing the cold start problem [C]// Proceedings of the 22nd International Conference on World Wide Web. Geneva: Canton of Geneva, 2013: 909 – 918.
- [4] NEWMAN D, ASUNXION A, SMYTH P, *et al.* Distributed inference for latent Dirichlet allocation [EB/OL]. [2015-02-02]. http://ciir.cs.umass.edu/wiki/pub/Main/791mtSpring09/distributed_topic_modeling.pdf.
- [5] BLEI D M, NG A Y, JORDAN M I. Latent Dirichlet allocation [J]. Journal of Machine Learning Research, 2003, 12(6): 993 – 1022.
- [6] SUN Y, ZHOU X, FU W. Unsupervised topic and sentiment unification model for sentiment analysis [J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2013, 49(1): 102 – 108. (孙艳, 周学广, 付伟. 基于主题情感混合模型的无监督文本情感分析[J]. 北京大学学报: 自然科学版, 2013, 49(1): 102 – 108.)
- [7] LU Y, ZHAI C. Opinion integration through semi-supervised topic modeling [EB/OL]. [2015-01-02]. <http://sifaka.cs.uiuc.edu/~yuelu2/pub/www08-opinion.pdf>.
- [8] JO Y, OH A H. Aspect and sentiment unification model for online review analysis [C]// Proceedings of the 4th ACM International Conference on Web Search and Data Mining. New York: ACM, 2011: 815 – 824.
- [9] SHAN B, LI F. A survey of topic evolution based on LDA [J]. Journal of Chinese Information Processing, 2010, 24(6): 43 – 49. (单斌, 李芳. 基于 LDA 话题演化研究方法综述[J]. 中文信息学报, 2010, 24(6): 43 – 49.)
- [10] LIN C, HE Y. Joint sentiment/topic model for sentiment analysis [C]// Proceedings of the 18th ACM Conference on Information and Knowledge Management. New York: ACM, 2009: 375 – 384.
- [11] HEINRICH G. Parameter estimation for text analysis [EB/OL]. [2010-10-10]. <http://www.arbylon.net/publications/text-est.pdf>.
- [12] PANF B, LEE L. Thumbs up: sentiment classification using machine learning techniques [C]// Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2002: 79 – 86.
- [13] MASE F, COLLIGONO A, VANDERMEULEN D, *et al.* Multimodality image registration by maximization of mutual information [J]. IEEE Transactions on Medical Imaging, 1997, 16(2): 187 – 198.
- [14] WILSON T, WIEBE J, HOFFMANN P. Recognizing contextual polarity in phrase-level sentiment analysis [EB/OL]. [2014-12-10]. <http://aclweb.org/anthology/H/H05/H05-1044.pdf>.