

散乱点云曲面边界的二次提取算法

童立靖, 郑俊朝

(北方工业大学 计算机学院, 北京 100144)

摘 要: 针对点云曲面边界提取算法计算量大、时间耗费多的问题, 提出一种点云曲面的二次边界提取算法。采用空间包围盒法将点云曲面均匀地分为若干个小立方体, 将每个点都放入一个立方体内, 并通过每个立方体周围非空子立方体的个数以及分布情况提取边界子立方体。结合点云曲面数据点的分布特征, 在边界子立方体内将目标点的所有 K 近邻点投影到以目标点为中心的平面上, 计算投影点与中心点形成的向量与某条坐标轴的夹角, 通过判断其是否满足预先设定的条件来判定目标点是否为边界点。实验结果表明, 该方法可有效减少计算量, 提高提取精度。

关键词: 点云曲面; 二次提取; 空间包围盒; 边界提取; 子立方体; 平面投影

中文引用格式: 童立靖, 郑俊朝. 散乱点云曲面边界的二次提取算法[J]. 计算机工程, 2017, 43(2): 252-256.

英文引用格式: Tong Lijing, Zheng Junchao. Secondary Extraction Algorithm for Scattered Point Cloud Surface Boundary[J]. Computer Engineering, 2017, 43(2): 252-256.

Secondary Extraction Algorithm for Scattered Point Cloud Surface Boundary

TONG Lijing, ZHENG Junchao

(College of Computer, North China University of Technology, Beijing 100144, China)

[Abstract] To overcome the defects of large amounts of calculation and long duration time consuming and calculation amount of current point cloud surface boundary extraction algorithms, a point cloud surface secondary boundary extraction algorithm is put forward. First of all, the space bounding box method is used to divide the 3D model into several sub cubes evenly. Every cloud point is put in a small cube. The boundary sub cubes are identified by the number and distribution of the sub cubes which has any cloud point. Then, according to the distribution of the data points, every point that is in a boundary sub cube and its K neighbor points are projected to a flat surface. At last, the boundary point is identified according to the principle that the angle between one axis and the vector formed by projected point and center point meets the preset conditions. Experimental results show that the secondary extraction algorithm for scattered point cloud surface boundary can save time and improve precision.

[Key words] point cloud surface; secondary extraction; space bounding box; boundary extraction; sub cube; planar projection

DOI: 10.3969/j.issn.1000-3428.2017.02.042

0 概述

随着当前激光技术以及计算机技术的快速发展, 基于逆向工程的产品研究成为产品工程设计的重要手段, 曲面点云模型是其中一个重要组成部分, 而曲面边界则是描述点云曲面的重要特征, 因此, 找到一个高效的曲面边界提取方法是很有必要的。目前的曲面边界提取技术通常是在三维扫描技术不成熟时期所产生的, 但随着高精度三维扫描技术的发

展与普及, 以前的曲面边界提取技术越发显得计算量庞大, 效率低下。例如文献[1]方法在进行边界特征点提取阶段, 需要扫描点云所有数据点, 之后通过平面拟合以及交叉计算向量之间的角度, 消耗时间较多, 不适合现今高精度大体量的点云曲面模型。文献[2]方法同样是对所有点进行检索, 利用力学相关原理来判定当前点是否为边界点, 对于目前数据量越来越大的点云曲面模型处理效率不高。文献[3]提出一种改进型边界提取算法, 但其改进点是

基金项目: 国家自然科学基金(61371142); 北京市教委专项(PXM2013_014212_000026)。

作者简介: 童立靖(1972—), 男, 副教授, 主研方向为数字图像处理、计算机图形学; 郑俊朝(通信作者), 硕士研究生。

收稿日期: 2015-12-17 **修回日期:** 2016-01-26 **E-mail:** 784518217@qq.com

在对点云模型建立空间索引结构阶段,抛弃常用的 K 近邻搜索,选择一种类似邻域代替 K 近邻点,虽然此项改进节省了计算量,但实际应用效果不如 K 近邻结构,而且最后同样是对所有点进行边界判定,计算量较大。文献[4]方法则是对散乱点云数据集建立最小生成树,通过对较短的特征线进行剪枝处理提取点云边界特征,该方法同样需要对点集内每一点都进行判定,而且提取效果不佳。

由以上研究可知,目前的方法多是先设计一种边界点判定方法,然后对点云模型的每一个点进行判定,若符合条件即为边界点,否则为非边界点,此类方法对于数据量越大的点云模型,边界提取效率越低。本文针对以上缺陷,提出一种高精度散乱点云曲面的边界二次提取算法,以实现快速有效的点云曲面边界提取。

1 算法思路

点云模型都是散乱、无规律的,一些常用算法无法直接运用,因此,处理前通常需要先对点云模型建立一个相对有序的索引结构,将所有点云通过某种结构联系起来。本文在此采用空间包围法,具体算法流程如图 1 所示。

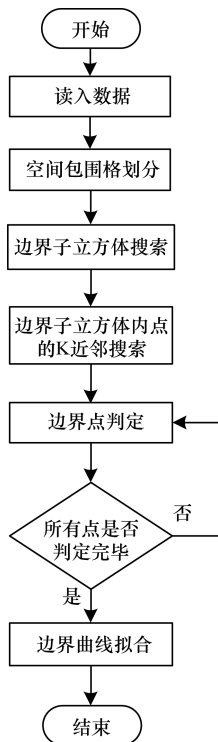


图 1 算法流程

首先将每一个数据点都放入一个空间子立方体内,接着将所有的子立方体内划分为 3 类:空子立方体,边界子立方体,非边界子立方体。空子立方体是指不包含任何数据点的子立方体;边界子立方体是指位于点云曲面边界处,包含边界点的子立

方体;非边界子立方体则是指位于曲面内部,包含非边界点的子立方体。由于边界子立方体周围的空子立方体个数明显多于非边界子立方体,由此可先将边界子立方体提取出来,然后在边界子立方体内搜索每个点的 K 近邻点,利用当前点及其所有 K 近邻点的坐标值,通过主成分分析法拟合一平面,并将点与其 K 近邻点投影到此平面上,接着以点在其 K 近邻点在平面上的投影点形成的向量与某特定向量之间的向量夹角的差异为判定条件,找出所有边界点。

2 空间包围盒建立

读入文本模型数据点集,将整个点集的坐标存入 3 个一维数组中,得到模型点集在 X, Y, Z 3 个坐标轴方向上的最大值与最小值,然后根据极值信息以及点的总数 n ,计算出子立方体边长 L ,同时将文本点云模型划分成 $m \times n \times l$ 子立方体^[5-6],记录每个子立方体所包含的数据点以及每个点所在的立方体索引号。曲面在 3 个坐标轴方向上的最大最小值,在此定义为: $Min_x, Min_y, Min_z, Max_x, Max_y, Max_z$, 则子立方体在 3 个坐标的个数可由式(1)计算。

$$\begin{cases} m = (\text{int})(Max_x - Min_x)/L \\ n = (\text{int})(Max_y - Min_y)/L \\ l = (\text{int})(Max_z - Min_z)/L \end{cases} \quad (1)$$

式中, L 为子立方体长度,由式(2)计算。

$$L = \beta \sqrt[3]{\frac{(Max_x - Min_x)(Max_y - Min_y)(Max_z - Min_z)}{n}} \quad (2)$$

其中, β 为调节立方体 L 大小的比例因子, n 为数据点的总个数^[7-8]。

若当前点 $P(x, y, z)$ 所在子立方体的索引号为 (u, v, w) , 则其索引号可由式(3)计算。

$$\begin{cases} u = (\text{int})(x - Min_x)/L \\ v = (\text{int})(y - Min_y)/L \\ w = (\text{int})(z - Min_z)/L \end{cases} \quad (3)$$

每一个子立方体建立一个链表,该表用来存储子立方体内所有点,并统计该子立方体内点的个数,数据结构如下:

```

typedef struct Cubecell
{
    LinkedList Link; //子立方体内数据点集表的头指针
    int n; //子立方体中所含数据点的个数
} Cube;
  
```

3 点云边界点提取

将所有的子立方体分为 3 类,即空子立方体、边界子立方体、非边界子立方体。以空间 3×3 为模

板,则每个子立方体周围皆存在 27 个子立方体,若子立方体为边界子立方体,则其周围非空子立方体的个数要少于非边界子立方体,以此为根据来找出所有的边界子立方体。

高精度与低精度点云模型之间的差异如图 2 所示。

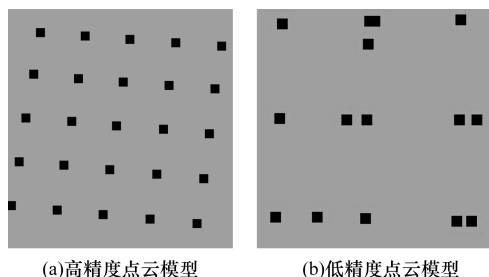


图 2 高精度与低精度点云模型的点排列比较

在图 2(a)中,高精度点云模型中点的排列比较有序,每个点在某个范围内的邻点个数以及相对位置是相对固定的,而图 2(b)中的低精度模型点的排列比较散乱,没有规律。因此,高精度点云中边界点与非边界点的 K 近邻点的分布情况是不一样的,边界点的 K 近邻点是分布在点的一侧,而非边界点则是围着点均匀分布的,可以根据此特点进行边界点的判定。

3.1 边界子立方体的初步提取

假设当前点 p_i ,若其所在子立方体不在边界上时,搜索此立方体上下、前后、左右共 27 个子立方体,在此定义这 27 个子立方体为其邻域^[9-10]。排除空立方体后,则 p_i 所在子立方体周围可能的分布情况的横截面如图 3 所示。

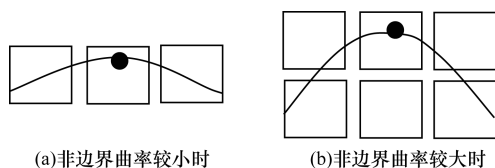


图 3 p_i 所在子立方体周围可能的分布情况横截面 1

在图 3(a)中,表明此处曲面曲率较小, p_i 所在子立方体周围共有 8 个非空子立方体;在图 3(b)中,表明此时曲面曲率较大, p_i 所在子立方体周围共有 14 个非空子立方体。

若 p_i 所在子立方体在边界上时,同样在排除空子立方体时, p_i 所在子立方体可能的分布情况的横截面如图 4 所示。

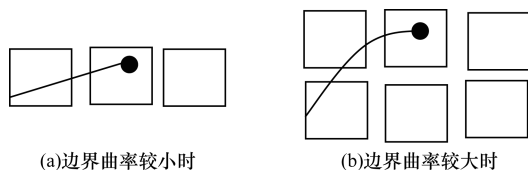


图 4 p_i 所在子立方体周围可能的分布情况横截面 2

在图 4(a)中,表明此时曲面曲率较小, p_i 所在子立方体周围共有 5 个非空子立方体;在图 4(b)中,表明此时曲面曲率较大, p_i 所在子立方体周围共有 8 个非空子立方体。

通过以上分析,首先对每个子立方体设置一标记值 $flag$, $flag$ 初始值设为 0,默认为非边界子立方体。在遍历当前点 p_i 时,先搜索其周围的非空子立方体的个数。若个数小于 8 则将其 $flag$ 值置为 1,即标记为边界子立方体。若个数为 8 则可能为图 3(a)或图 4(b)所示情形,检测非空子立方体在当前子立方体周围的分布情况,若所有非空子立方体在其周围均匀分布,视为非边界子立方体。若都在当前子立方体一侧,则将其 $flag$ 值置为 1。若数目大于 8 个,则为图 3(b)所示情况,为非边界子立方体。在遍历下一点时,先检测其所在子立方体是否已被标记,若标记,则跳过,直到找到一个点所在的子立方体未被标记时,重复以上步骤,最终得到一个由子立方体边界构成的边界。

3.2 边界子立方体内 K 近邻搜索

K 近邻的定义为 $P = P_i (i = 0, 1, \dots, n)$ 是点云曲面模型 X 上的一个采样点集,将 X 中与当前点 P_i 距离最近的 K 个点称为点 P_i 的 K-近邻,记作 $Nb(p)$ ^[11-12],在此的 K 近邻点搜索不局限于边界子立方体内。

由于已经知道每个点所在子立方体的索引号^[13-14],就可通过以下步骤求取每个点的 K 个近邻点:

- 1) 确定当前点所在子立方体的索引号,并将此子立方体视为初始搜索区域;
- 2) 计算当前点到所在立方体 6 个面的最短距离 d_s ;
- 3) 计算当前搜索区域点的总个数 N ,若 N 不小于 K ,则接着计算当前点与搜索区域内其他点的最大距离 d_{max} ,若 d_{max} 小于 d_s ,取前 K 个点为当前点的 K 近邻点,否则以当前区域为中心向周围扩展建立新的搜索区域。

每一个数据点建立一个 K 邻近表。该表用来保存每个点的所有 K 邻近点以及它们之间距离,数据结构如下:

```
typedef struct Kpoint
{
    double x,y,z;//邻近点的坐标信息
    double distance;//数据点与邻近点之间的距离
    struct Kpoint * next;//下一个邻近点
} Kpoint, * KList;
```

3.3 边界点的二次提取

非边界点与边界点周围点的分布情况如图 5 所示。从中可知,边界点与非边界点周围点的数目是不同的,假设当前点 $p(x_i, y_i, z_i)$,由于点云曲面在空间中是弯曲的,因此先将 p 与其 K 近邻点进行平面

投影^[15]。若 p 的 K 近邻点集 $X = \{x_0, x_1, \dots, x_{k-1}\}$, 则可通过主成分分析法建立一经过点 p 且拟合所有 K 近邻点的平面 L 。

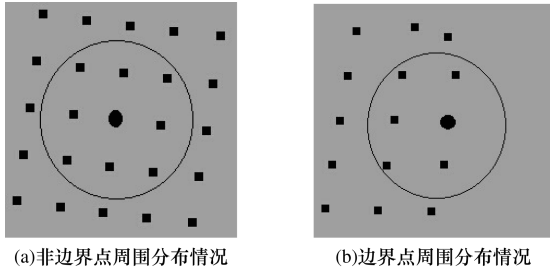


图 5 非边界点与边界点周围点的分布情况

在平面 L 上建立以 p 点为中心点的平面坐标系, 2 条坐标轴分别为散布矩阵 S 的最大特征值与次大特征值对应的特征向量 e_1 与 e_2 。其中, S 可由式(4)表示。

$$S = \sum_{i=0}^{k-1} (x_i - p)(x_i - p)^T \quad (4)$$

把 X 中的点投影到平面 L 上, 得到一投影点集 $X' = \{x'_0, x'_1, \dots, x'_{k-1}\}$, 其中投影点的坐标可由式(5)计算。

$$x'_i = e_1^T (x_i - p), y'_i = e_2^T (x_i - p), \quad i = 0, 1, \dots, k-1 \quad (5)$$

依次计算 $px'_i, i = 0, 1, \dots, k-1$ 与 e_1 向量之间的夹角, 最后得到一个夹角的集合 $\theta = \{\theta_0, \theta_1, \dots, \theta_{k-1}\}, 0^\circ \leq \theta_i < 360^\circ$, 根据夹角从小到大的顺序对所有 K 近邻投影点进行排序。若存在一个夹角满足 $|\theta_i - \theta_{i+1}| > \alpha, i = 0, 1, \dots, k-1, \theta_k = \theta_0 + 360^\circ, \alpha$ 为预先设定的一阈值, 则视 p 点为边界点。

4 边界线生成

得到边界点集后, 需要进行点与点之间连接, 生成边界线。这里采用双向最近点搜索算法来生成多边形折线^[16]。具体步骤如下:

- 1) 读取边界点集 P , 在 P 中任取一点 p_i 为边界线的起点, 以其最近点 p_j 为终点。
- 2) 对 p_i 点, 在剩余边界点中搜索 p_i 的最近点 p_r 。
- 3) 计算 p_r 到 p_i 的距离 d_i 和到 p_j 的距离 d_j , 如果 $d_i < d_j (d_i < d_j)$, 将 p_r 插入到 $p_i(p_j)$ 前并作为新的起始点, 否则往另一端 $p_j(p_i)$ 进行搜索。
- 4) 转步骤 2), 直到点集 P 为空, 算法结束。

5 实验结果及分析

本文算法在 Visual Studio 2005 环境下采用 C++ 语言实现, 测试环境为: Intel © Core™7 CPU; 内存 8 GB; 操作系统 Windows 7 的机器。在此共提供 3 个点云曲面模型, 其中模型 1 为撕裂文本曲面,

模型 2 为半版扭曲文本曲面, 模型 3 为复杂文本点云曲面。本文默认点云曲面无噪点的干扰。

原始模型和边界提取效果如图 6 所示。

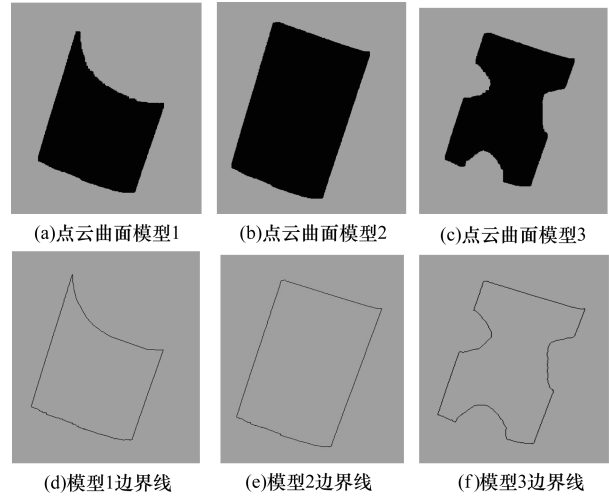


图 6 原始模型和边界提取效果

本文算法与文献[1]算法的边界提取耗时对比如表 1 所示。

表 1 边界提取耗费时间对比 ms

模型	模型点数	本文算法	文献[1]算法
模型 1	18 345	5 566	22 355
模型 2	30 543	12 325	45 972
模型 3	92 566	38 564	202 501

由表 1 可知, 本文算法消耗的时间明显比文献[1]算法要少, 而且随着 3 个模型数据点的个数的增加, 文献[1]算法消耗的时间成指数形式增加, 而本文算法则接近线性增加, 验证了本文在前面提到的以文献[1]算法为代表的传统点云曲面边界提取算法对模型所有点进行边界点判定的缺陷, 而本文算法则很好地弥补了这一缺陷。由于硬件条件的不同以及其他条件的影响, 重现的文献[1]算法所消耗的时间可能和原文有所差异, 但并不影响在同等条件下本文算法与其进行消耗时间的对比分析。

6 结束语

本文提出一种针对散乱点云曲面边界的二次提取算法。首先, 对点云模型进行基于 K 近邻与空间网格相结合的散乱点云空间拓扑排序, 增强对复杂点云曲面的处理能力; 然后, 提出点云曲面边界子立方体的概念, 即包含点云边界点的空间单元格; 最后, 通过边界子立方体与非边界子立方体周围非空子立方体的个数以及分布情况的差异, 提取所有边界子立方体, 进行边界点判定时, 可以只对边界子立方体内的点进行判定, 省去了大量对非边界点的判定。实验结果表明, 本文算法相比于文献[1]算法节省了大量时

间,而且随着点云数据量的增加,该算法的优势会更加明显。不足之处是本文只是针对文本曲面进行实验。但文本点云曲面曲率波动幅度较小,且位置相对水平,因此,下一步将在本文算法的基础上继续研究针对其他复杂曲面(例如地形等),或者曲面模型在空间中位置较复杂情况时的曲面边界提取算法。

参考文献

- [1] 潘荣江,孟祥旭.点云曲面边界提取[J].几何设计与计算的新进展,2005,27(7):52-57.
- [2] 吴禄慎,晏海平,陈华伟,等.一种基于散乱点云的边界提取算法[J].计算机应用与软件,2014,31(11):264-268.
- [3] 陈义仁,王一宾,彭张节,等.一种改进的散乱点云边界特征点提取算法[J].计算机工程与应用,2012,48(23):177-179.
- [4] Gumhold S, Macleod R, Wang Xinlong. Feature Extraction from Point Clouds [C]//Proceedings of the 10th International Meshing Roundtable. Newport Beach, USA: [s. n.], 2001:293-305.
- [5] 王晏民,郭明.大规模点云数据的二维与三维混合索引方法[J].测绘学报,2012,41(4):605-612.
- [6] Dey P, Nousath S. A Robust Skew Detection Method for Scanned Document Images[J]. Pattern Recognition, 2010,43(3):937-948.
- [7] 雷敏,仲思东,屠礼芬.一种三维点云聚类算法的研究[J].科学技术与工程,2014,14(29):20-53.
- [8] Hans-Peter K, Marisa P, Matthias S, et al. Efficient Similarity Search on 3D Bounding Box Annotations [C]//Proceedings of Medical Imaging 2012:Advanced PACS-based Imaging Informatics and Therapeutic Applications. San Diego, USA:SPIE,2012:1605-7422.
- [9] 李宝,程志全,党岗,等.三维点云法向量估计综述[J].计算机工程与应用,2010,46(23):1-7.
- [10] 张巧英,陈浩,朱爽.密度聚类算法在连续分布点云去噪中的应用[J].地理空间信息,2011,9(6):101-104.
- [11] 董洪伟.散乱点云的三角网格重构[J].计算机工程,2005,31(15):30-32.
- [12] 徐杰.一种基于立方格的散乱点云曲面网格化方法[J].北京大学学报(自然科学版),2014,50(2):228-234.
- [13] Piegl L A, Tiller W. Algorithm for Finding All k-nearest Neighbors [J]. Computer-Aided Design, 2002,4(2):167-172.
- [14] 张毅,刘旭敏,隋颖,等.基于K-近邻点云去噪算法的研究与改进[J].计算机应用,2009,29(4):1011-1014.
- [15] 叶珉吕,花向红,陈西江,等.基于正交整体最小二乘平面拟合的点云数据去噪方法研究[J].测绘通报,2013(11):37-39.
- [16] 程效军,方芳.基于形态学的散乱点云轮廓特征线提取[J].同济大学学报(自然科学版),2014,42(11):1738-1743.

编辑 金胡考

(上接第 251 页)

- [7] Kaur P, Jain A, Mittal S. Touch-less Fingerprint Analysis—A Review and Comparison [J]. International Journal of Intelligent Systems and Applications, 2012,4(6):46-52.
- [8] Labati R D, Genovese A, Piuri V, et al. Contactless Fingerprint Recognition: A Neural Approach for Perspective and Rotation Effects Reduction [C]//Proceedings of IEEE Workshop on Computational Intelligence in Biometrics and Identity Management. Washington D. C., USA:IEEE Press,2013:22-30.
- [9] Parziale G. Touchless Fingerprinting Technology, A Chapter in Advances in Biometrics [M]//Ratha N K, Govundaju V. Sensors, Systems and Algorithms. Berlin, Germany: Springer,2007.
- [10] Labati R D, Scotti F. A Neural-based Minutiae Pair Identification Method for Touch-less Fingerprint Images [C]//Proceedings of IEEE Workshop on Computational Intelligence in Biometrics and Identity Management. Washington D. C., USA:IEEE Press,2011:96-102.
- [11] 秦芬芬,廖斌.非接触指纹图像分割与增强方法的研究[J].传感器世界,2014,20(8):16-19.
- [12] Chikkerur S, Cartwright A, Govindaraju V. Fingerprint Image Enhancement Using STFT Analysis [J]. Pattern Recognition,2007,40(1):198-211.
- [13] Hiew B, Andrew B, Pang Y. Digital Camera Based Fingerprint Recognition [C]//Proceedings of IEEE International Conference on Telecommunications and Malaysia International Conference on Communications. Washington D. C., USA:IEEE Press,2007:676-681.
- [14] Hiew B, Teoh A, Ngo D. Preprocessing of Fingerprint Images Captured with a Digital Camera [C]//Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision. Washington D. C., USA:IEEE Press,2006:1-6.
- [15] Hiew B Y, Teoh A B J, Yin O S. A Secure Digital Camera Based Fingerprint Verification System [J]. Journal of Visual Communication and Image Representation,2010,21(3):219-231.
- [16] 李学明.基于Retinex理论的图像增强算法[J].计算机应用,2005,22(2):235-237.

编辑 顾逸斐