

采用局部凸性和八叉树的点云分割算法

傅欢¹, 梁力², 王飞¹, 赵季中¹

(1. 西安交通大学电子与信息工程学院, 710049, 西安; 2. 西安交通大学软件学院, 710049, 西安)

摘要: 针对粗糙点云分割效果差的问题,提出了一种采用八叉树和局部凸性的点云分割算法.该算法首先通过仪器扫描得到仅包含坐标信息的点云数据,然后对点云进行法向量估算,并根据点云的法向量信息进行八叉树初始分割得到面片,最后根据面片之间的局部凸性特征进行融合,得到最终的分割结果.与其他同类算法相比,采用八叉树和局部凸性的点云分割算法不仅能有效地减少曲面数量,而且在曲面质量上也优于同类算法.采用塔身震落石块的点云数据进行的实验表明,该算法在处理分布较均匀的闭合点云数据时,能够有效减少最终的曲面个数,且面片的质量与手工分割拟合度达到 90% 以上.

关键词: 点云分割;法向量估算;八叉树;局部凸性

中图分类号: TP391. 4 **文献标志码:** A **文章编号:** 0253-987X(2012)10-0060-06

A Point Cloud Segmentation Algorithm Using Local Convexity and Octree

FU Huan¹, LIANG Li², WANG Fei¹, ZHAO Jizhong¹

(1. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China;

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

Abstract: An algorithm of point cloud segmentation using octree and local convexity is proposed to deal with the fact that normal segmentation methods have poor effect on coarse point clouds. The proposed algorithm is composed of three steps. Firstly, point clouds with 3D coordinates are obtained using special instruments. Then, normal estimation approaches are used to get the normal vector and to segment the point clouds into patches by octree. Finally, the neighboring patches are merged if local convexity is held. Compared with other works, the proposed algorithm can reduce the number of surface effectively and produce surfaces with better quality. Experimental results show that the proposed algorithm can better segment the nearly and equally distributed data into several meaningful surfaces and 90% of the surfaces is the same as that produced by human operation.

Keywords: point cloud segmentation; normal estimation; octree; local convexity

随着现代三维扫描技术与建模技术的提高,基于采样点的三维数据模型,即点云模型,被广泛地应用于计算机可视化、文物保护、环保监控和航空监控等领域^[1].点云模型由于具有表达细节能力强、存储简单、信息量大等特点,已成为常用的三维物体表示模型之一.近年来,对三维数据的处理方法成为了研究热点,而点云的曲面分割,作为点云数据处理的第

一步,也是研究的热点之一.

现有的点云曲面分割技术主要分为 2 类:一是采用几何特征进行分割^[2-5],这类方法对点云模型的要求极为严格,当其应用于粗糙实物点云时,容易产生过分割的问题;二是利用其他领域的模型来解释点云模型,然后利用这种模型的特性来进行分割,这类方法时间复杂度较高,适用范围十分有限.在现实

应用中,点云模型通常是粗糙的,且应用在时间代价上有一定的限制,因此主要应用都采用第一类方法,主要有:文献[2]提出首先建立点云的 4 邻域无向图,然后根据点的邻域信息拟合一个局部平面,利用平面的法向量的局部凸性特征进行分割;文献[3]提出利用八叉树进行分割后,对相邻平面进行合并.但是,文献[2]方法的 4 邻域无向图无法有效计算粗糙曲面的法向量特征,而文献[3]方法仅对点云中平面提取效果较好.为了更好地解决无组织散乱粗糙点云的分割问题,本文提出基于局部凸性和八叉树的分割算法,在利用更多邻域点估算法向量特征的情况下,利用改进的八叉树算法进行初步分割,随后利用相邻面片之间的局部凸性特性^[6-7]来进行曲面融合,最终得到一个满足应用要求的分割结果.本文算法避免了过分割现象的发生,也在时间代价上满足了应用的要求.

1 曲面分割的定义

点云数据通常被看成是一个点集,而点云曲面分割就是根据一定的特征量或者特征量的组合对点云进行聚类计算.简单地说,就是将原始点集分割成若干个互不相交的子集且每个子集的分割依据互不相同.因此,根据集合的定义可以将点云曲面分割表述为:设点云数据集合为 S ,分割所采用的特征量集合 $F = \{F_1, F_2, \dots, F_n\}$,最终分割结果集合 $R = \{R_1, R_2, \dots, R_n\}$,其中 n 为要求的最终分割后的曲面数. R 中的每一个元素即为对应的 F 中的元素的聚类结果,代表着分割后得到的一个曲面,集合 R 必须满足以下条件:

- (1) $\bigcup_{i=1}^n R_i = S$,即点云数据集合 S 中的每一个点都被分配到曲面中;
- (2) $R_i \cap R_j = \emptyset$,表示每 2 个曲面之间没有交集,即不存在一个点同时属于 2 个曲面;
- (3) 每个 R_i 对应一个 F_i ,且相邻的 2 个曲面 R_i, R_j 对应使用的特征量 F_i, F_j 也不相同;
- (4) 每一个 R_i 必须是连通的.

在不同的分割算法中,特征量的选择不尽相同.本文算法采用的特征量为法向量.

2 点云的法向量估算

点云由一系列的散乱点组成,在几何上表现为三维空间中的一组离散点,无法直接计算其法向量.通常情况下,需要使用参考点的临近点信息来估算参考点的法向量.因此,在估算一个点的法向量时,

首先应选定该点位参考点,然后使用与之相邻的点按照一定的规则组成三角面片,所有包含参考点三角面片都会影响参考点的法向量,最后使用一个加权函数来估算参考点的法向量.在计算一个点的法向量之前,首先要把该顶点与其相邻的点组成一系列三角面片.本文采用最常见的德劳内三角化方法得到三角面片.由于一个点可能出现在多个三角面片中,因此一个点的法向量会受到多个面的法向量的影响,通常采用平均加权的方法来估算法向量.点 p 的法向量 $\hat{n}_p$ 由式(1)、式(2)计算

$$n_p = \frac{\sum_{i=1}^n n_i}{m} \quad (1)$$

$$\hat{n}_p = \frac{|n_p|}{n_p} \quad (2)$$

式中; m 为包含点 p 的三角面片的个数; n_i 代表包含该点的第 i 个三角面片的法向量.点 p 的法向量由包含点 p 的 6 个三角面片的法向量决定,如图 1 所示.

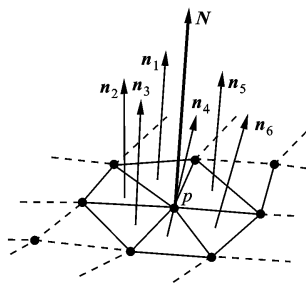


图 1 点的法向量的估算

3 初步分割

根据点的法向量进行点云曲面分割的方法有自顶向下^[8]和自底向上^[9]的 2 种方法.自顶向下方法首先将待分割的点云看作一个整体,然后进行逐步细分,直到满足某个条件才停止分割,是一个聚类细化的过程.这种分割方法的优点是可以将所有的点都考虑到分割的过程中,但是如果最终的分割结果依赖于分割停止的条件,很容易出现过分割的情况.自底向上的方法则是首先将所有的点分别看作一个聚类,然后进行融合,在融合的过程中,最初采用的特征量会一步一步弱化.这种方法可以确保最终得到的曲面的个数,但是可能得到的曲面没有意义.本文采用自顶向下的方法,使用八叉树数据结构对点云数据进行存储和分割.

八叉树的建立过程如图 2 所示.左侧是点云分

割的过程,先找到一个长方体将点云包含在内,然后根据长方体的中心点将其逐步细分.右侧即为左侧的分割对应建立的八叉树.树的每个顶点保存着对应的点云数据的最小外截长方体的8个顶点,其中包含的点云数据以及它的8个子节点.图中节点0被分割为8个子节点,其中子节点3和7被再次分割为8个子节点.

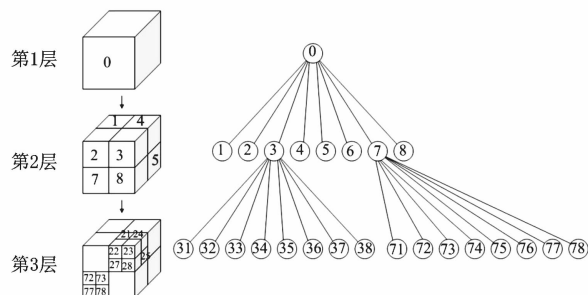


图2 八叉树的建立过程

分割步骤如下:首先分别找到点云数据中在X、Y、Z轴上的最大值 $x_{\max}$ 、 $y_{\max}$ 、 $z_{\max}$ 和最小值 $x_{\min}$ 、 $y_{\min}$ 、 $z_{\min}$,根据这6个值可以确定一个最小长方体将所有的点云数据包含在内,而这个长方体的8个顶点数据也作为八叉树的根节点属性信息保存,同时保存根节点内的点云数据;然后用式(3)计算该节点内的点的平均法向量

$$\mathbf{n}_{\text{ave}} = \sum_{i=0}^m \mathbf{n}_i / m \quad (3)$$

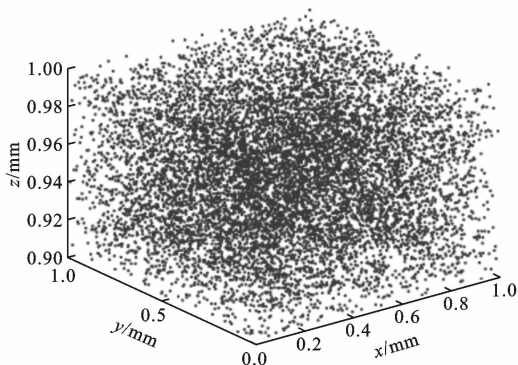
式中: m 为根节点所包含的点的个数; $\mathbf{n}_i$ 为每个点的单位法向量.在完成上一步后,利用式(4)计算根节点的法向量的标准差 σ ,如果 σ 小于或等于某个给定的阈值 T_{thr} ,则不再进行细分,否则进行进一步的细化.

$$\sigma = \left| \frac{\left(\sum_{i=1}^m (\mathbf{n}_i - \mathbf{n}_{\text{ave}}) \right)^{1/2}}{m} \right| \quad (4)$$

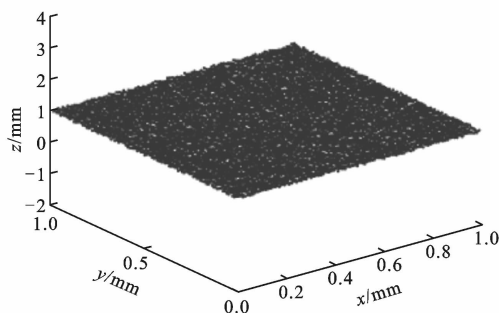
在处理过程中,可能会出现某个节点包含的点云数据十分有限,小于点云数据数量阈值 N_{num} ,此时默认为这个节点不再可分.如果出现某个节点内没有点云数据,则将该节点设置为空.

在这个过程中存在2个参数——法向量标准差允许误差阈值 T_{thr} 和点云数量阈值 N_{num} ,这2个参数决定了八叉树初始分割的结果. T_{thr} 决定了一个面中允许存在的法向量差值,即决定了初始分割的面片的大小,从而决定了初始分割的尺度.当 T_{thr} 值较小时,点云数据中的局部细微凸起将会被分割成2个部分,当 T_{thr} 值取得较大时,一些原本应该被识

别出的凸起信息则没有被识别到,从而导致分割的结果不能满足应用的要求.图3为不同尺度下的点云模型,同一点云模型在不同的尺度下可以表现出不同的形态,图3a中不能看出该点云的形态,而图3b则表现出一个近似平面.同理,在特征量的尺度空间中,选取的尺度大小同样会影响到点云的特性.



(a)小尺度下的点云



(b)大尺度下的点云

图3 不同尺度下的点云模型

T_{thr} 通常根据点云模型确定,但是在实际应用中,这个参数的合适值是难以确定的,所以引进了点云数据数量阈值 N_{num} ,用来控制八叉树的层数,根据这个最小点云数量阈值,可以保证在 T_{thr} 较小的情况下仍能点云分割成一定数量范围内的面片,且能较好地找出点云模型中的平面部分的点云.

面片的数量是决定下一步计算量的主要因素,而面片的特征量的一致性则确定了分割的准确度.这两者之间相互影响,很难确定一组较好的值确保两者都能达到一个好的结果.因此,本文的实验中,在确定一个可接受的面片数量的范围后,法向量标准差阈值则要求尽量小,从而可以尽可能多地保留点云局部特征.

4 分割融合

由上述的步骤可知,整个分割效果除了与点云

的法向量特征^[10-11]估算的准确程度有关,还与其中给定的 2 个阈值 T_{thr} 和 N_{num} 相关. 一旦这 2 个阈值不合适就会造成过分割的现象,而利用法向量的局部凸性进行分割后的区域融合则能在一定程度上解决这个问题.

局部凸性定义如图 4 所示,表明了相邻 2 个曲面的法向量之间的角度关系. 图中 p_1 和 p_2 代表着 2 个曲面上的一个点, n_1 和 n_2 代表着 2 个曲面在该点的法向量,当 p_1 所在平面上的点都在 p_2 所在的平面的下方时,则 p_1 所在的曲面和 p_2 所在的曲面满足局部凸性.

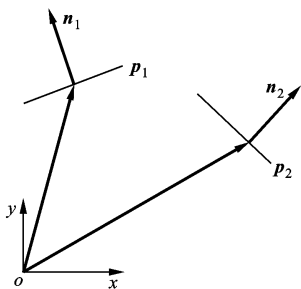


图 4 局部凸性的定义

判定 2 个曲面是否满足局部凸性的准则如下式

$$l(s_i, s_j) = \left[\left(n_i d_{ij} \leq \|d_{ij}\| \cos\left(\frac{\pi}{2} - \epsilon_1\right) \right) \wedge \left(n_j d_{ji} \leq \|d_{ji}\| \cos\left(\frac{\pi}{2} - \epsilon_1\right) \right) \right] \vee \left[n_i n_j \geq 1 - \|n_{ij}\| \cos\left(\frac{\pi}{2} - \epsilon_2\right) \right] \quad (5)$$

式中: $d_{ij} = p_j - p_i$ 为曲面中心点与坐标原点组成的向量的差; ϵ_1 是控制局部凸性的程度的参数,在相对比较平滑的曲面上可以取值为 0; ϵ_2 是控制点云数据中的噪声容忍程度参数. 当 2 个相邻曲面使式(5)的值为真时,表明 2 个曲面可以融合. 融合后将相关的 p_j 和 n_j 进行更新,然后进行下一步的融合. 在实验中,随机选取一个曲面作为起始曲面开始融合. 如果没有曲面可以与其融合,则看是否有曲面未被融合过. 如果有,则重新选择一个曲面重复这一过程,如果没有,则融合过程结束.

这一步骤中涉及到 2 个参数 ϵ_1 和 ϵ_2 . 在实验中, ϵ_1 简单地取值为相邻面的法向量的夹角,通常取值范围在 0 至 π . 当应用于场景中的实物分割时, ϵ_1 取值为 $\pi/2$ 将能很好地识别出垂直于场景平面的物体. 在本文的实验中,需要分割出较明显的曲面边缘,所以 ϵ_1 取值为 $\pi/4$. 在非人工数据的实验过程中,点云数据存在的噪声情况很难确定,所以 ϵ_2 通常是一个经验值. 在本实验中暂不考虑噪声的影响,

所以 ϵ_2 取值为 0.

5 实验结果分析

本文算法主要经过了初分割和曲面增长的合并算法 2 个阶段. 其中初分割阶段产生的是一系列拥有邻接关系的细小面片,曲面合并阶段产生的则是若干个大的曲面,曲面的数量则是由点云数据的法向量的分布和采取的合并策略相关的.

为了证明本文提出的方法的有效性,本文采用塔身震落的石块点云模型进行分割. 由于在现实应用中,可能有一些先验知识为本文的分割过程提供帮助,所以本文采取了 2 种曲面合并策略:迭代合并和指定最终曲面数. 迭代合并策略是一种无监督的自主合并策略,只需要在初分割的结果中寻找一个种子节点即可,然后依据局部凸性的特性进行曲面融合. 指定最终曲面数的策略则需要指定合并过程中需要的所有种子节点,然后依据局部凸性进行聚类操作.

图 5 是本文实验中采用的点云数据的原始目标图像. 根据图像可以大致判断出物体的基本形状以及分割后的应产生的曲面数及曲面的理想形状.

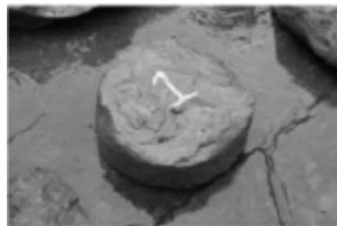
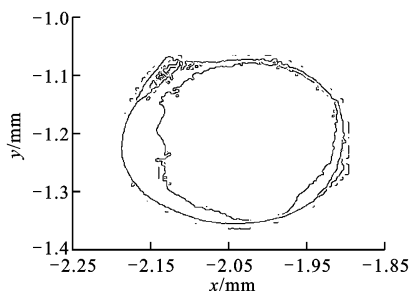


图 5 实验数据的原始目标

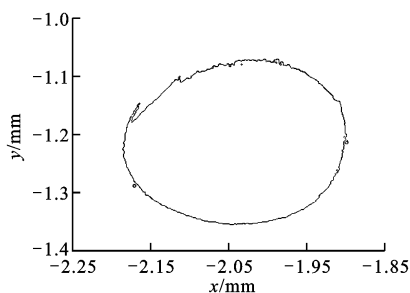
采用文献[2]方法与本文算法进行比较,为了更好地表现分割出的曲面信息,本文采用最终结果的边缘图像证明本文算法的有效性,如图 6 所示.

从实验的分割效果来看,本文的算法可以较好地将点云上端的曲面完整地分割开,而文献[2]方法主要对平面检测有较好的效果,对于粗糙点云,则在局部点云上没能将点云的曲面进行很好的分割,所以在分割结果的边缘检测图中可以明显地看出在一些局部地区有一些空隙,这些空隙则对应着点云中并没有把曲面较好地分割的部分. 从最终分割出的曲面数量上来看,本文采用的 2 种融合策略分别分割出了 10 个和 3 个曲面,文献[2]方法则分割出了 8 个曲面. 在这 2 种方法中,都存在点云模型中外延部分的分割效果不够理想的情况.

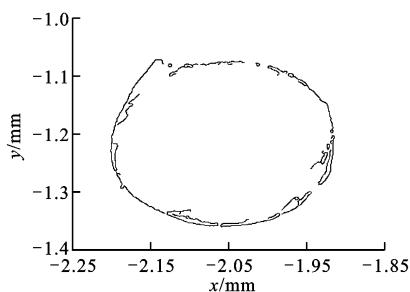
经多组实验分析表明,造成点云模型的外延部



(a) 本文算法无指定种子节点分割结果



(b) 本文算法指定种子节点分割结果



(c) 文献[2]方法的分割结果

图6 本文的2种融合策略与文献[2]方法的对比图

分分割效果不理想的主要原因是点云的分布密度在外延部分和顶部曲面部分差距明显,因此,在估算特征进行初始分割和后续的面片合并过程中,应该根据不同的区域采用不同的参数,才能得到较好的效果,在后续的工作中,将继续研究点云的分布密度对分割的影响,进一步完善本文的方法.从实验的时间消耗上来看,本文算法与文献[2]方法的时间复杂度均为 $O(kn^2)$,其中 n 为点云数据中点的数量,在 matlab 下仿真均需要大于 1 min 的时间,所以通过有效地简化点云模型达到减少算法的时间消耗也是下一步的研究工作之一.

综上所述,本文的算法可以有效地减少曲面的最终分割数量,且分割出的曲面能够较好地表现出点云的原有特性.相对于原有的方法,本文提出的方法能有效地保留原有方法中体现出的局部特性,同

时也能有效地合并相邻的近似曲面,最终达到一个满足应用要求的分割结果.同时,本文的方法有一些需要改进的地方,例如在保持局部特性的情况下对点云进行简化^[12],然后进行曲面分割,以此提高分割效率.

6 结 语

本文在采用法向量作为特征、利用八叉树的数据结构进行初步分割的基础上,利用局部凸性特性来进行曲面融合,最终得到了理想的分割结果.在以后的工作中,将从以下两方面进行改进:①寻找一种简单的方法来表示点与点之间的相邻关系,目前的表达方式过于复杂,影响查询效率;②目前的算法对于近似等密度的封闭点云模型有好的分割效果,下一步则需要消除点云密度和点云模型的封闭性对算法的影响.

参考文献:

- [1] 费文龙,魏志辉. Mumford-Shah 模型在卫星图象分割中的应用[J]. 中国图象图形学报, 2009, 14(4): 598-603.
FEI Wenlong, WEI Zhihui, An application of Mumford-Shah model in segmentation of satellite cloud image [J]. Journal of Image and Graphics, 2009, 14(4): 598-603.
- [2] MOOSMANN F, PINK O, STILLER C. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion[C]// Proceedings of IEEE Intelligent Vehicles Symposium. Piscataway, NJ, USA: IEEE, 2009: 215-220.
- [3] TSENG Yi-Hsing, WANG Miao. Automatic plane extraction from LIDAR data based on octree splitting and merging segmentation [C]// Proceedings of IEEE International Geoscience and Remote Sensing Symposium. Piscataway, NJ, USA: IEEE, 2005: 3281-3284.
- [4] WANG Miao, TSENG Yi-Hsing. Automatic segmentation of Lidar data into coplanar point clusters using an octree-based split-and-merge algorithm [J]. Photogramme Thrice Engineering and Remote Sensing, 2010, 76(4): 407-420.
- [5] DOUILARD B, UNDERWOOD J, KUNTZ N, et al. On the segmentation of 3D LIDAR point clouds [C]// Proceedings of IEEE International Conference on Robotics and Automation. Piscataway, NJ, USA: IEEE, 2011: 2798-2805.
- [6] DAVID S, JIRI Z. Graph cut based point-cloud seg-

- mentation for polygonal reconstruction [C]// Proceedings of the 5th International Symposium on Advances in Visual Computing. Heidelberg, Germany: Springer Verlag, 2009: 218-227.
- [7] MARIO R, MARKUS V. Point cloud segmentation based on radial reflection [C]// Proceedings of 13th International Conference on Computer Analysis of Images and Patterns. Heidelberg, Germany: Springer Verlag, 2009: 955-962.
- [8] JAGANNATHAN A, MILLIER E L. Three-dimensional surface mesh segmentation using curvedness-based region growing approach [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(12): 2195-2204.
- [9] STEINHAUSER D, RUEPP O, BURSCHKA D. Motion segmentation and scene classification from 3D LIDAR data [C]// Proceedings of IEEE Intelligent Vehicles Symposium. Piscataway, NJ, USA: IEEE, 2008: 398-403.
- [10] ZHU Xiaolong, ZHAO Huijing, LIU Yiming, et al. Segmentation and classification of range image from an intelligent vehicle in urban environment [C]// Proceedings of 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems. Piscataway, NJ, USA: IEEE Computer Society, 2010: 1457-1462.
- [11] PANG Xufang, PANG Mingyong. An algorithm for extracting geometric features from point cloud [C]// Proceedings of 2009 International Conference on Information Management, Innovation Management and Industrial Engineering. Piscataway, NJ, USA: IEEE Computer Society, 2009: 78-83.
- [12] 张连伟, 李炎, 刘肖林, 等. 基于局部曲面拟合的散乱点云简化方法[J]. 计算机工程与科学, 2010, 32(12): 65-68.
- ZHANG Lianwei, LI Yan, LIU Xiaolin, et al. A simplification method for cloud points based on local surface fitting [J]. Computer Engineering & Science, 2010, 32(12): 65-68.

(编辑 刘杨)