

一种基于边缘系数的闭合点云边缘提取算法

韩玉川^{**}, 侯贺, 白云瑞, 朱险峰^{*}

天津大学精密仪器与光电子工程学院, 天津 300072

摘要 提出一种通过种子边缘点提取闭合点云边缘的算法, 通过搜索视角控制边缘点云搜索方向, 可提取尖锐边缘与相交边缘。算法通过模拟点与点之间拉力的聚集程度来判断一个点是否具有边缘点的特征。首先构建 kd-tree 以便快速查找点云数据。其次定义点云的邻域半径, 通过计算种子点邻域内点的边缘系数来选择新的种子边缘点, 并通过搜索视角来控制点云搜索的方向, 最终提取闭合的点云边缘。选择不同的种子点与搜索视角可提取不同的边缘。通过实验分析邻域半径和搜索视角对算法边缘提取效果的影响, 并采用合适的半径与搜索视角, 对尖锐边缘与相交边缘进行点云提取, 获得了准确的点云边缘, 从而证明了该算法的有效性。

关键词 图像处理; 点云边缘提取; 边缘系数; 邻域半径; 种子点

中图分类号 R318.51

文献标识码 A

doi: 10.3788/LOP55.111003

A Closed Point Cloud Edge Extraction Algorithm Using Edge Coefficient

Han Yuchuan^{**}, Hou He, Bai Yunrui, Zhu Xianfeng^{*}

School of Precision Instruments & Opto-Electronics Engineering, Tianjin University, Tianjin 300072, China

Abstract A new algorithm is proposed to extract the edge of the closed point cloud through the edge seed point. Using the algorithm can extract the sharp edge and intersecting edge by setting an appropriate search angle. The algorithm determines whether a point has the characteristic of edge point by simulating the aggregation degree of pull between points. First, build the kd-tree to find each point quickly. Second, define the neighbor radius of the point cloud, and select the new edge points by calculating the edge coefficient of the seed point located in the neighbor radius. Use the search angle of the seed point to control the search direction. Finally, extract the closed point cloud's edge. Choosing different seed points and search angles can extract different edges. The effects of the neighbor radius and the search angle were analyzed by the experiment. The appropriate neighbor radius and search angle were used to extract the sharp edge and the intersecting edge, and the accurate edge of the point cloud was obtained, which proves the effectiveness of the algorithm.

Key words image processing; point cloud edge extraction; edge coefficient; neighbor radius; seed point

OCIS codes 100.6890; 100.2000; 100.3008

1 引言

点云边缘提取是目标识别、点云下采样和可视化应用等领域的基础步骤^[1]。点云边缘不仅是描述模型曲面的一种重要的几何特征, 也对几何模型的重建质量和精度起着关键作用。点云的边缘点具有多义性, 没有明确的边缘点定义, 因而点云边缘点的判定算法也有很多种, 但其基本原则是提取的边缘点必须能准确描述数据的几何形状特征。

当前的点云边缘判定算法有微切平面法、栅格划分法、图理论法、深度图像法、协方差特征值法、八邻域深度差法、拉普拉斯算法、角度差法等。孙殿柱等^[2]通过点云采样点 P 及其临近的几个点, 用最小二乘法拟合出一个关于点 P 的微切平面。然后把以上几个点投影到微切平面上, 在微切平面上计算点 P 与其他点构成的向量间的最大夹角, 通过最大夹角结合设定的阈值来判断点 P 是否为边界点。陈义仁等^[3]也是用类似文献^[2]的方法建立微切平

收稿日期: 2018-04-16; 修回日期: 2018-05-21; 录用日期: 2018-06-04

* E-mail: zhuxianfeng@tju.edu.cn; ** E-mail: hanyuchuan2012@163.com

面,在微切平面上通过定义的点 P 与其他点的“拉力”大小结合设定的阈值来判断点 P 是否为边界点。柯映林等^[4]首先进行空间栅格划分,并建立栅格的空间拓扑关系,根据空间栅格中是否包含点云来寻找边界栅格,然后在边界栅格中的六个面中找一个合适的投影面,把栅格中的点投影到该面上,构造简单的几何判据来找到边界点。Demarsin 等^[5]用区域生长的方法根据点云的法向量对点云数据进行精简,保留处在边缘区域的点簇,再利用图的理论从这些候选点中计算出闭合的边缘轮廓。刘浩等^[6]将三维点云数据转换为深度图像,通过比较每个采样点和其邻域点间距离的变化,判断该点是否为边界点。由于深度图像只能反映单值曲面的深度状况,因此,通过深度图只能检测单值曲面的点云边界。Bazazian 等^[1]通过分析所定义的每个点 K 邻域的协方差矩阵特征值来判断该点是否具有边缘点的特征。张志佳等^[7]将特征点沿深度方向进行垂直投影并对投影点进行栅格划分,计算每个栅格与八邻域栅格的深度差,通过深度差判断栅格内是否存在边界点。Xi 等^[8]根据深度信息将三维点云转换为二维图像,然后利用改进的拉普拉斯算法判断边界点。刘增艺等^[9]计算每个数据点的 K 邻域,通过数据点 K 邻域构成最小二乘平面的投影点角度差,根据角度差的最大值判断该点是否为边界点。

本文提出一种新的边缘点判断方法,可交互指定要提取的某一条闭合边缘。要提取的点云边缘是闭合边缘,包括曲面轮廓边界及曲面相交的曲线边

缘。方法简述如下:首先给出边缘点的数学定义,然后通过已知的种子边缘点寻找种子点邻域内潜在的边缘点,最终提取种子点所在的边缘。

与前述算法相比,本文算法的特点是:1) 不需要进行点云数据三角化;2) 判断边缘点时不需要构建微平面,也不需要设定全局的阈值;3) 除了全局整体提取之外,也可提取指定的边缘。

2 点云边缘提取算法

首先构建 kd-tree 以便快速查找点。通过计算点云的间距来定义点的球形邻域半径。然后使用边缘系数判断准则来提取种子边缘点邻域内的边缘点,针对尖锐的点云边缘采取扩大搜索视角的策略来寻找边缘点。

2.1 建立点云空间拓扑

通过三维扫描设备获取的点云数据具有数据量大且散乱无规律的特点,需要建立三维数据点的空间拓扑关系来实现基于邻域关系的快速查找。建立点云之间拓扑关系的常用方法有 kd-tree、oc-tree 和空间单元格^[10]。本文采用 kd-tree 建立点云间的拓扑关系。

kd-tree^[11-12]是一种高维索引的二叉树结构,通常用来查找距离最近的两个点。处理点云数据时,只在三个维度中进行,因此是三维的 kd-tree。如图 1 所示^[12],kd-tree 的每一级节点分开所有的子节点,如果子节点在指定的维度上小于根节点,它将被分在左边的子树中,如果大于根节点,将被分在右子树中。

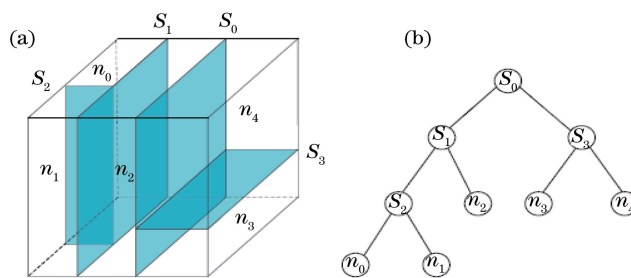


图 1 (a)三维 kd-tree(内部节点被标记在切平面,叶子节点被标记在所属空间);(b)对应的二叉树结构

Fig. 1 (a) 3D kd-tree (interior nodes are labeled as their splitting planes and leaf nodes are labeled in their boxes); (b) graph representation of the same kd-tree

在 kd-tree 中进行数据的查找,目的是检索在 kd-tree 中与查询点距离最近的数据点。基本方法如下:首先通过二叉树搜索,小于节点则进入左子树,否则进入右子树,进而快速找到目标点的近似点。然后回溯搜索路径,判断搜索路径上节点的其他子节点是否有距离目标点更近的点。如果有则进入其他节点空间继续搜索,重复该过程直到搜索路

径为空。

2.2 计算点云邻域半径

在二维数字图像中,位于坐标 (x, y) 处的像素 p 周围有八个相邻的像素,这八个像素点一起称为 p 的八邻域。利用点的八邻域信息,可以选择八个点中的一个作为下一个边缘点,进而设计边缘跟踪算法。受此启发,可以在三维点云数据中构建类似

的邻域,作为边缘搜索的基础。因为边缘点不在点云内部,而是在点云的外表面上,所以可用八邻域法寻找边缘点。

如果一个点云数据是均匀的,局部呈现网格排列,则可以定义点 p 的八邻域为距离点 p 最近的八个点,这八个点可以通过 kd-tree 快速找到。但实际点云数据分布并非呈现均匀的网格排列,按照上述方法,生成一个以点 p 为球心、包含八邻域点在内半径最小的球,在点云密集和稀疏的地方,球的半径会有显著不同,会对边缘点的跟踪造成影响。

采用如下邻域定义:在点云中随机选取 N 个点,在每个点 p_i 邻近的八个点中计算距离 p_i 最远的点与 p_i 的距离 d_i 。对所有的 d_i 求平均得到点云邻域半径的平均值 \bar{R} 。再计算 d_i 的标准差,最终得到点云的邻域半径 $R_0 = \bar{R} + \sigma$ 。

$$\bar{R} = \frac{1}{N} \sum_{i=1}^N d_i, \quad (1)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (d_i - \bar{R})^2}. \quad (2)$$

半径是一个统计平均值,考虑到点云密度并不均匀,所以额外叠加了一个标准差,以减小点云密度不均匀的影响,保证多数情况下每个邻域中有八个点。由于点云的离散程度会影响邻域点的数量,所以不能保证每个点云在邻域内都有八个临近点,但能够满足大多数的点,所定义的点云邻域半径具有一定的统计意义。

2.3 边缘系数

处在非边缘处的点,其邻域内的点分布在该点的四周;处在边缘处的点,其邻域内的点聚集在某一个方位。假设三维空间中的点与点之间存在“弹簧”连接,点与点之间存在拉力,那么每个点受到邻域内其他点的拉力的合力会有大小和方向,边缘上的点所受拉力的合力较大,非边缘上的点由于周围拉力相互抵消,整体所受拉力较小。直观而言,非边缘点四周分布着点云,边缘点附近的点只分布在某一侧,因此点所受拉力的合力大小可以作为判断该点是否为边缘点的依据(图 2)。

由于点云分布散乱,点的间距存在不同程度的差别。如果用所受周围点拉力的合力来评价该点是否具有边缘点的特征,具有相同特征的点在距离不同时计算得到的合力是不同的。因此只用合力大小作为边缘特征判断标准并不准确。此外,由于邻域半径是一个统计结果,点云邻域内邻近点的数量

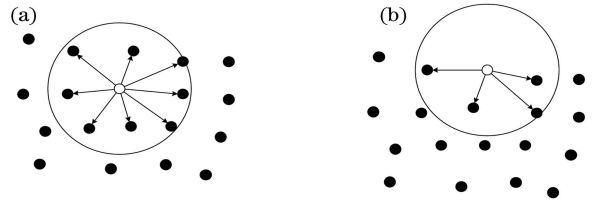


图 2 受力状况。(a)非边缘点受力状况;(b)边缘点受力状况

Fig. 2 Tension. (a) Tension at the non-edge points;

(b) tension at the edge points

也会有差别。以上两个因素会影响用合力大小作为判断准则的有效性。

为了克服上述两个因素的影响,采用边缘系数 F 来衡量点 P_0 邻域内的点分布的聚集程度。边缘系数 F 与点 P_0 邻域内点的间距没有关系,每个点对 P_0 的拉力进行归一化后再累加,消除了点云间距不均匀带来的影响。边缘系数定义为

$$F(P_0) = \frac{1}{k} \left| \sum_{i=1}^k \frac{\overrightarrow{P_0 P_i}}{|P_0 P_i|} \right|, \quad (3)$$

式中: P_i 为 P_0 的第 i 个邻域点; k 为 P_0 邻域内点的数量。边缘系数 F 越大,说明点 P_0 周围的点分布越不均匀,越脱离点云主体,具有边缘点的特征。

当通过一个已知的边缘种子点 P_0 提取边缘时,将该种子点邻域点中最有可能是边缘点的点作为下一个种子点,循环此过程直到边缘提取完毕。为了寻找 P_0 邻域 N 个点中的下一个种子点,首先计算 P_1, P_2, \dots, P_N 各个点的边缘系数 F_1, F_2, \dots, F_N , 寻找 N 个边缘系数中的最大值 F_{\max} , F_{\max} 所对应的点 P_i 将作为新的种子点。

2.4 搜寻边缘点

在计算 P_1, P_2, \dots, P_N 各个点的边缘系数 F_1, F_2, \dots, F_N 时,有可能出现两个边缘系数大小相同的情况。在寻找 F_1, F_2, \dots, F_N 的最大值时,如果存在两个相同的最大值,如 4, 4, \dots , 1, 会随机选择两个相同边缘系数中的一个。例如在点云排列有序、等距的情况下会出现这种情况,如图 3 所示。

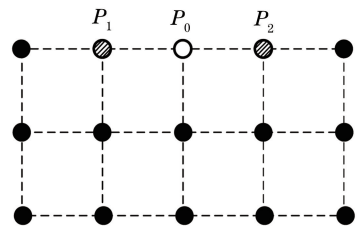


图 3 点云排列整齐时选择新种子点的情况

Fig. 3 Case of selecting the new seed point when the point clouds are evenly aligned

过程如下:计算得到种子点 P_0 邻近点 P_1 和 P_2 的边缘系数相同,随机选取 P_1 或 P_2 作为下一个种子点。如果选取 P_1 作为新的种子点,则在计算 P_1 邻近点边缘系数时会重现此问题,有可能随机选择 P_0 作为新的种子点。 P_0 和 P_1 将会交替作为种子点,程序会陷入死循环。

为了避免这种状况,引入搜索视角的概念,如图 4 所示。

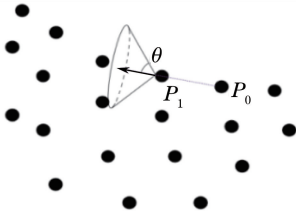


图 4 搜索视角

Fig. 4 Search angle

向量 $\overrightarrow{P_0P_1}$ 由初始种子点 P_0 指向由 P_0 找到的边缘点 P_1 。当 P_1 作为新的种子点在 P_1 的 N 个邻域点中选择下一个边缘点 P_i 时,被选择的点 P_x 需要满足: P_1 与点 P_x 形成的向量 $\overrightarrow{P_1P_x}$ 与向量 $\overrightarrow{P_0P_1}$ 之间的夹角小于设定的搜索视角 θ 。在这些满足条件的点中选择边缘系数最大的点作为 P_i 。

搜索视角 θ 的值可以根据要提取的点云边缘状况在 $(0, \pi)$ 范围内加以调整。边缘相对平滑时, θ 可以设定为较小的值,避免不必要的干扰。当边缘有较大的转折时, θ 需要设定为较大的值。

按照以上步骤自动搜索边缘点,直到再次搜索到原始种子点时停止搜索,得到的边缘点形成闭合的边缘。

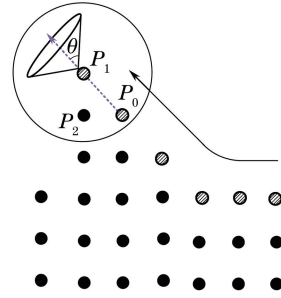
2.5 针对尖锐边缘的处理

搜索视角 θ 设置为较小的值有利于避免非边缘点的干扰以及减小计算量。因此,在进行边缘点搜索时 θ 的值设定为 $45^\circ \leq \theta \leq 90^\circ$ 。在点云边缘变化缓慢处,能够完整地提取点云边缘点,如图 5 所示,箭头方向是点云边缘的搜索方向。当 P_1 作为种子边缘点时,搜索视角内没有点存在,此时终止边缘点搜索。正确的搜索应该是将 P_2 作为下一个种子边缘点。

当种子边缘点位于尖锐边缘处时,如果邻域内有点存在但是搜索视角内没有点存在,可以对该点的搜索视角 θ 在一定范围内逐渐增大,在新的搜索视角内找到下一个种子边缘点。

3 实验结果

提出一种新的点云闭合边缘提取算法,算法提

图 5 搜索视角较小时易忽略 P_2 点Fig. 5 P_2 will be ignored if the search angle is too small

取的点云边缘包括曲面边缘以及两个面相交所形成的边缘。首先使用图 6 所示平面点云数据说明算法对尖锐边缘的处理。然后使用两个相交的圆柱体点云数据来说明算法对两个面相交产生的边缘点的提取效果。最后对一个有孔洞的兔子模型进行点云边缘提取,并与文献[7]的八邻域深度法进行对比。

尖锐边缘主要分凸和凹两种情况。图 6(a)所示为一个包含凹凸结构的平面边缘提取效果,通过一个位于 $(0, 0, 0)$ 的初始种子点提取边缘点并标记为白色。图 6(b)所示为算法对边缘的处理效果,在这种理想点云构成的平面中,凸出尖锐边缘形成的最大角度为 45° ,因此在程序实现中,搜索视角 θ 的最大值设定为 150° 。图 6(c)所示为算法在提取凹陷边缘时表现出的平滑效果,这种效果与本文定义的点云邻域半径大小有关。

图 7 所示为两个交叉圆柱模型点云的边缘提取结果。与图 6 所示点云模型不同,图 7 所示点云的边缘由两个面相交构成。图 7(a)所示为通过一个初始种子点提取的边缘,图 7(b)所示为通过六个初始种子点提取的全部边缘。初始种子点从交界面中根据边缘特征手动选取。该点云数据是散乱的点云数据,其整体的散乱程度可以通过计算邻域半径的两个参数来说明: $\bar{R} = 2.72, \sigma = 0.15$ 。图 7(c)所示为边缘点云的一处细节,放大部分中一个边缘点明显偏离了边缘的轮廓,这是由该点处点云分布不均匀造成的,类似的情况在其余边缘部分也有发生。该点云边缘提取所使用的参数如下: $R = 2.72, \theta = 120^\circ$ 。

图 8 所示为不同参数下交叉圆柱点云模型的边缘提取结果,其中 R_0 表示求得的邻域半径为 2.72。由于点云空间散乱排列,图 8(a)中当邻域半径为 R_0 ,搜索视角 $\theta = 100^\circ$ 时,一旦新的种子点 P 偏离点云的边缘轮廓,由于 θ 较小, P 的邻域点中处在搜索视角范围内的点不包括正确的边缘点。因此 θ 过小

会导致边缘提取结果中出现一些非边缘点,严重者会导致边缘提取失败。图 8(b)所示为 $R=R_0, \theta=135^\circ$ 时的边缘提取结果,与图 8(a)类似,也出现了一些非边缘点。与 $\theta=100^\circ$ 情况不同的是, $\theta=135^\circ$ 属于 θ 过大,在提取边缘点时,搜索视角内包含过多非边缘点,干扰了对边缘点的判断。因此,在处理散乱点

云时, θ 过小或过大都会使边缘提取结果中混入一些非边缘点。图 8(c)和图 8(d)所示为大邻域半径情况对边缘提取结果的影响。图 8(c)和图 8(d)分别为 $R=1.5R_0$ 时不同 θ 的边缘提取效果,可以看出点云边缘提取完整。图 8 表明在处理点云较为散乱的情况下,增大邻域半径有助于边缘点的正确提取。

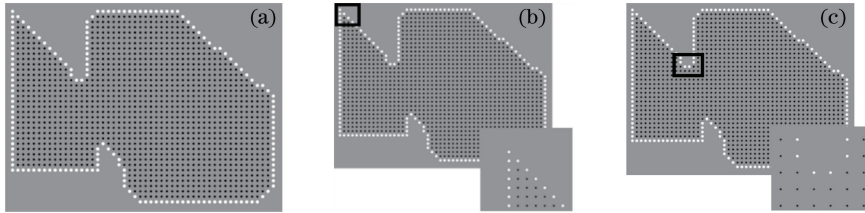


图 6 (a)具有尖锐边缘特征的理想点云数据边缘提取结果;(b)尖锐边缘处的提取效果;(c)算法提取凹陷边缘呈现出的平滑效果
Fig. 6 (a) Result of the space plane with sharp edge formed by the ideal point clouds; (b) extraction result at the sharp edge; (c) smooth effect at the depression edge obtained by the algorithm

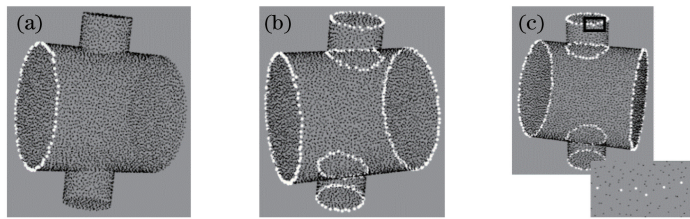


图 7 (a)通过一个种子边缘点提取的闭合边缘;(b)通过六个边缘上的种子点提取的边缘;(c)离散分布的点云对提取边缘的影响
Fig. 7 (a) Closed edge extracted by one initial seed edge point; (b) all edges extracted by six initial seed edge points; (c) effect of uneven distribution of point clouds on edge extraction

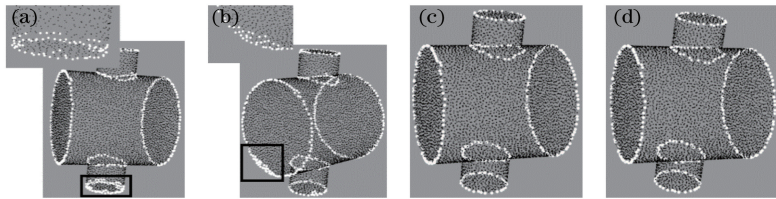


图 8 不同邻域半径与搜索视角下的边缘提取结果。(a) $R=R_0, \theta=100^\circ$; (b) $R=R_0, \theta=135^\circ$;
(c) $R=1.5R_0, \theta=100^\circ$; (d) $R=1.5R_0, \theta=135^\circ$

Fig. 8 Results for different neighbor radii and search angles of two intersecting cylinders. (a) $R=R_0, \theta=100^\circ$;
(b) $R=R_0, \theta=135^\circ$; (c) $R=1.5R_0, \theta=100^\circ$; (d) $R=1.5R_0, \theta=135^\circ$

图 9 和图 10 所示为边缘系数法与八邻域深度法对一个有孔洞的兔子点云进行边界提取的结果。

所用时间对比见表 1。



图 9 边缘系数法

Fig. 9 Edge coefficient method

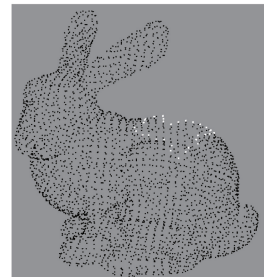


图 10 八邻域深度法

Fig. 10 Eight-neighbor depth method

表 1 孔洞边缘提取时间

Table 1 Hole edge extraction time

Algorithm	Number of points	Time /s
Edge coefficient method	3417	0.974
Eight-neighbor depth method	3417	1.625

图 9 和图 10 得到的边缘点云一致,根据表 1 可以看出边缘系数法效率比八邻域深度法提升了 1/3 左右,关键在于边缘系数算法不需要对点云进行投影、栅格数据组织等操作,节省了时间。

4 结 论

提出了一种通过种子边缘点提取点云闭合边缘的算法。算法通过模拟点与点之间的拉力聚集程度来判断一个点是否具有边缘点的特性,在提取边缘点时不需要进行局部三角化以及构建微平面。实验表明该算法能够提取曲面轮廓边缘及曲面相交的曲线边缘。针对不同散乱程度的点云数据,可以通过调整邻域半径的大小和改变搜索视角达到良好的效果。

参 考 文 献

- [1] Bazazian D, Casas J R, Ruiz-Hidalgo J. Fast and robust edge extraction in unorganized point clouds[C] // International Conference on Digital Image Computing: Techniques and Applications, IEEE, 2016: 1-8.
- [2] Sun D Z, Fan Z X, Li Y R. Automatic extraction of boundary characteristic from scatter data[J]. Journal of Huazhong University of Science and Technology (Natural Science), 2008, 36(8): 82-84.
孙殿柱, 范志先, 李延瑞. 散乱数据点云边界特征自动提取算法[J]. 华中科技大学学报(自然科学版), 2008, 36(8): 82-84.
- [3] Chen Y R, Wang Y B, Peng Z J, *et al.* Improved algorithm for extraction of boundary characteristic point from scattered point cloud[J]. Computer Engineering and Applications, 2012, 48(23): 177-180.
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): 177-180.
- [4] Ke Y L, Fan S Q. Research on direct extraction of boundary from point clouds[J]. Chinese Journal of Mechanical Engineering, 2004, 40(9): 116-120.

柯映林, 范树迁. 基于点云的边界特征直接提取技术[J]. 机械工程学报, 2004, 40(9): 116-120.

- [5] Demarsin K, Vanderstraeten D, Volodine T, *et al.* Detection of closed sharp edges in point clouds using normal estimation and graph theory[J]. Computer-Aided Design, 2007, 39(4): 276-283.
- [6] Liu H, Li D, Yu Z G, *et al.* Boundary point extraction of point cloud based on range image[J]. Journal of Water Resources and Architectural Engineering, 2013, 11(3): 156-158.
刘浩, 李冬, 于振国, 等. 基于深度图像的点云边界点提取[J]. 水利与建筑工程学报, 2013, 11(3): 156-158.
- [7] Zhang Z J, Wei X, Zhou Z Q, *et al.* 8N-DD-based point cloud edge extraction algorithm [J]. Chinese Journal of Scientific Instrument, 2017, 38(8): 2015-2021.
张志佳, 魏信, 周自强, 等. 一种基于八邻域深度差的点云边缘提取算法[J]. 仪器仪表学报, 2017, 38(8): 2015-2021.
- [8] Xi X, Wan Y, Wang C. Building boundaries extraction from points cloud using an image edge detection method [C] // Geoscience and Remote Sensing Symposium, IEEE, 2016: 1270-1273.
- [9] Liu Z Y, Jiang K Y, Lin J Y. Interactive extraction of boundary of specified target feature on scattered point cloud [J]. Computer Engineering and Applications, 2016, 52(6): 186-190.
刘增艺, 江开勇, 林俊义. 散乱点云特征边缘交互提取[J]. 计算机工程与应用, 2016, 52(6): 186-190.
- [10] Zhou X L, Wan W G. Efficient boundary points extraction method for scattered point cloud of endocardium [J]. Application Research of Computers, 2012, 29(10): 3942-3944.
周学礼, 万旺根. 心内膜散乱点云边界点检测算法研究[J]. 计算机应用研究, 2012, 29(10): 3942-3944.
- [11] Tao Y B, Lin H, Bao H J. Kd-tree based fast ray tracing for RCS prediction [J]. Progress in Electromagnetics Research, 2008, 81: 329-341.
- [12] Liao L Q, Bai J S, Luo D A. Integrated point cloud storage structure based on octree and KD tree[J]. Computer Systems & Applications, 2012, 21(3): 87-90.
廖丽琼, 白俊松, 罗德安. 基于八叉树及 KD 树的混合型点云数据存储结构[J]. 计算机系统应用, 2012, 21(3): 87-90.