

三维点云中关键点误匹配剔除方法

熊风光, 霍旺, 韩燮, 况立群

中北大学计算机科学与技术学院, 山西 太原 030051

摘要 三维点云关键点配准与识别过程中存在寻找匹配对不理想、大量误匹配对及配准与识别准确率下降等问题,提出了一种新颖的关键点误匹配剔除方法。在关键点检测阶段,基于边缘点及其邻域点大多分布在同侧的特性,提出了一种边缘点检测算法,剔除处于边缘的关键点,以提高关键点的可重复性和可匹配性,并降低关键点特征匹配的误匹配率。在关键点特征匹配阶段,对经由最近邻算法得到的初始关键点匹配对,通过 Kmeans 算法和分裂法,剔除掉大量错误的关键点匹配对,从而提高三维点云之间关键点的匹配率。实验结果表明,该方法能够剔除完整三维点云匹配完整三维点云、完整三维点云匹配杂乱且有遮挡的三维点云、部分点云匹配部分点云所产生的大量关键点误匹配对,提升了关键点匹配效果;同时在时间上,本文算法较随机取样一致性算法更有效率,是最邻近算法的有益补充。

关键词 图像处理; 三维点云; 关键点; 边缘检测; Kmeans 算法; 分裂法

中图分类号 TP242

文献标识码 A

doi: 10.3788/AOS201838.0210003

Removal Method of Mismatching Keypoints in 3D Point Cloud

Xiong Fengguang, Huo Wang, Han Xie, Kuang Liqun

*School of Computer Science and Technology, North University of China,
Taiyuan, Shanxi 030051, China*

Abstract There are many problems exist in the process of registration and recognition of keypoints in 3D point cloud, such as finding matching mismatch, large number of mismatched pairs, and the decreasing of registration and recognition precision. A novel removal method of mismatching keypoints is proposed. In the stage of keypoint detection, an edge point detection algorithm is put forward based on the feature that the edge points and their neighbor points are mostly distributed on the same side. The proposed method can remove the keypoints existing in the edge to improve the repeatability and match ability, and reduce the mismatching rate in the feature matching of keypoint. In the stage of feature matching of keypoints, the initial keypoint matching pairs obtained by nearest-neighbor algorithm are matched, a lot of mismatching keypoint pairs can be removed according to the methods of Kmeans and splitting, and the matching rate of keypoints between 3D point clouds can be improved. Experimental results show that a large number of the mismatching keypoint pairs can be removed, which generated by a complete point cloud matching a complete point cloud, a complete point cloud matching a point cloud with clutter and occlusion, and a partial point cloud matching a partial point cloud, lots of mismatching keypoint pairs, and the matching effect of keypoints can be significantly improved. At the same time, the proposed algorithm is more efficient than random sample consensus algorithm in the time consumption, which is a good supplement to the nearest-neighbor algorithm.

Key words image processing; Three-dimensional point cloud; keypoint algorithm; edge detection; Kmeans algorithm; splitting method

OCIS codes 100.2960; 100.6890; 110.6880

收稿日期: 2017-08-30; **收到修改稿日期:** 2017-09-27

基金项目: 国家自然科学基金(61672473,61379080)、山西省回国留学人员科研项目(2015-079)

作者简介: 熊风光(1979—),男,博士研究生,讲师,主要从事虚拟现实、计算机视觉方面的研究。

E-mail: xfgncit98@sina.com

导师简介: 韩燮(1964—),女,博士,教授,主要从事仿真与可视化、智能信息处理等方面的研究。

E-mail: hanxie@nuc.edu.cn(通信联系人)

1 引 言

随着新一代低成本 RGBD 相机(微软 Kinect 和 华硕 Xtion)和数据采集技术的发展,以及中央处理器(CPU)和图形处理器(GPU)性能的显著提升,三维(3D)数据获取与处理变得越来越便捷,并极大地促进了三维点云数据处理在工业零部件生产、文物和建筑保护等领域的飞速发展。配准和识别是三维点云数据处理中最热门的两大核心技术,目前较为流行的配准与识别技术大多是基于三维点云的关键点匹配,因其具有运算速度快、效率高的优势。但由于环境及设备的限制,采集的数据中通常带有一定的噪声或孔洞,这使得计算关键点对应关系的结果并不理想,存在误匹配的问题,极大地影响后续配准^[1]与识别^[2]工作的准确性。

三维点云中关键点匹配是三维计算机视觉的核心技术,已广泛地应用于三维对象识别和点云配准等领域。关键点匹配包括关键点检测、关键点描述(特征描述)和关键点特征匹配等 3 部分,每一步均有较成熟的算法完成其特定的功能,如方向直方图特征(LSP)^[3]、内在形态特征(ISS)^[4]、关键点质量(KPQ)^[5]和网格高斯差分(MESHDoG)^[6]等算法用于关键点检测;快速点特征直方图(FPFH)^[7]、旋转投影统计(RoPS)^[8]和方向直方图特征(SHOT)^[9-10]等算法用于关键点描述;最近邻(NN)^[11]和最近邻距离比(NNDR)等算法用于关键点特征匹配。实际应用中,若采集到的三维点云质量高且噪声少,运用这些算法能够快速完成关键点匹配且匹配效果较佳;但是,当三维点云中存在一定噪声或者孔洞、遮挡和杂乱时,按照既有流程、依据现有算法往往难以获得较佳的关键点匹配效果,因此需要对误匹配对进行剔除,以应对配准或识别对匹配效果的要求。

目前针对误匹配剔除也有相关的研究,如在点云配准中,文献[12]、[13]中利用迭代最近点(ICP)^[14-16]精细配准技术对前期误匹配结果进行后期校准,取得了一定效果,但因为 ICP 算法需要整个三维点云参与迭代,而不仅仅是已经匹配好的关键点,因此存在时间效率低的问题;在三维对象识别领域,文献[17]和[18]利用随机取样一致性(RANSAC)算法剔除错误的关键点匹配对,得到了较佳匹配对,提升了三维对象识别的效果,但因为 RANSAC 算法也需要进行大量的迭代,以获取局内点的最大化,这必将对算法的性能造成一定的影响,

而且如果在初始的匹配对中存在较多的误匹配对,RANSAC 算法将难以从中随机选择出较好的 3 对及其以上的匹配对,反而会降低匹配和三维对象识别的效果。此外,这些误匹配剔除方法的生命周期均开始于特征匹配之后,忽视了在关键点检测过程中存在的问题,如因为边缘响应,上述的关键点检测算法通常将边缘点也认定为关键点,而实际上这些边缘点存在信息缺失和特征描述低下的问题,会产生大量的误匹配对。

鉴于目前存在的问题,本文提出一种新颖的误匹配剔除方法,运用到关键点检测和关键点特征匹配两个阶段。在关键点检测阶段,提出边缘点检测算法,并基于该算法,逐一剔除处于边缘的关键点,以期得到高质量、可重复和易匹配的关键点,尽量减少因边缘关键点而产生的误匹配对数量。在关键点特征匹配阶段,提出将 Kmeans 算法和分裂法相结合,对利用 K-D 树进行 NN 查找到的关键点匹配对,进行快速有效的误匹配对剔除。

2 方法论

图 1 所示为通常方法的关键点匹配流程,主要包含关键点检测、关键点描述和关键点特征匹配等。图 2 所示为本文方法关键点匹配流程,在通常的关键点匹配流程的基础上加入了边缘关键点剔除和基于 Kmeans 和分裂法的误匹配对剔除两大内容。

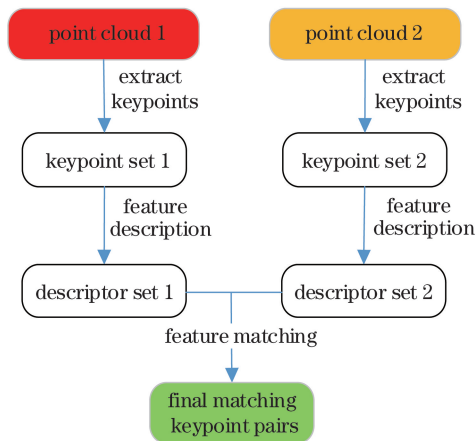


图 1 通常方法的关键点匹配流程

Fig. 1 Keypoint matching process of general method

2.1 边缘关键点剔除算法

2.1.1 算法描述

在部分三维点云杂乱且有遮挡的三维点云中,扫描获取的点云数据普遍存在数据不完整的点云边缘。即使在完整的三维点云中,由于设备的限制或是模型的因素也常有孔洞产生,而在这些孔洞与点

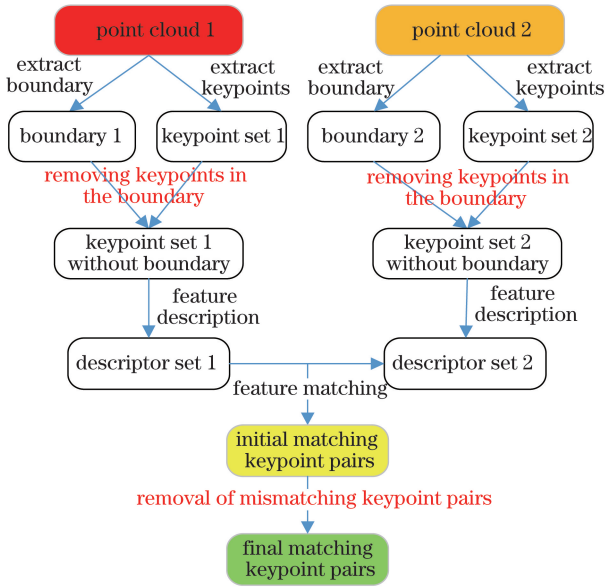


图 2 本文方法的关键点匹配流程

Fig. 2 Keypoint matching process of proposed method

云数据的边缘,由于邻域数据不完整,描述子无法很好地描述出其局部特征,所以会在特征匹配过程中造成误匹配。为更好地进行三维点云匹配,在关键点检测的基础上,要将在孔洞或点云数据边缘的关键点剔除,从而提高匹配率。

根据边缘点邻域内的数据点大多分布在一侧的特性^[19-20],采用如下算法检测三维点云 P_{PC} 中的边缘点。

1) 以任意点 p 为中心,搜索半径 r 范围内的邻域点,记为 $N(p)$, $N(p) = \{p_j \mid p_j \in P_{PC}, \|p_j - p\| \leq r\}$ 。

2) 根据 p 点的法向量 n 与 p 点可以构建 p 点处的切平面;将 $N(p)$ 投影至切平面,其点集记为 $N(q)$;在 $N(q)$ 中查找距离 p 点最近的点,记为 q_i ;最后,以 p 点法向量 n 作为 w 轴, $pq_i / |pq_i|$ 作为 u 轴, $u \times n$ 作为 v 轴, p 点为中心构建局部坐标系,记为 (p, u, v, w) 。

$$q_j = p_j - (n \cdot pp_j) \times n. \quad (1)$$

3) 分别计算由点 p 到其余 k 个邻域内投影点 q_j 的向量 pq_j 与 u 轴之间的顺时针夹角 $S = (\alpha_1, \alpha_2, \dots, \alpha_{k-1})$ 。对其进行升序排列得到新的角度序列 $S' = (\alpha'_1, \alpha'_2, \dots, \alpha'_{k-1})$,由下式计算出角度序列差 L 。显然, L 的几何意义是相邻向量之间的夹角,如图 3 所示。找出 L 中最大角 L_{max} ,若 L_{max} 大于阈值 L_{TH} (本文中设为 $\pi/4$),则认为 p 点为边缘点。

$$L = \alpha'_{i+1} - \alpha'_i, i \in [0, 1, \dots, k-2]. \quad (2)$$

边缘关键点剔除算法以边缘点检测算法为核心,其步骤如下:

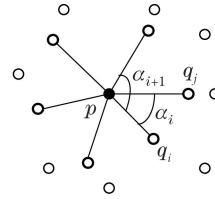


图 3 夹角示意图

Fig. 3 Sketch map of intersection angle

1) 遍历 P_{PC} 中所有的点,根据上述边缘点检测算法进行边缘点检测;如若是边缘点,将其加入到集合 E_{edge} 中。

2) 利用已知的关键点检测算法对 P_{PC} 中的每一个点进行关键点检测,将检测到的关键点加入到集合 $K_{keypoints}$ 中。

3) 搜索集合 $K_{keypoints}$ 中的每一个 kp 距集合 E_{edge} 中点的最近点,其距离记为 d ;若 $d < \delta$ (本文 δ 设为 10) 面片分辨率(单位为 mr),则认定该关键点处于点云边缘,将其从集合 $K_{keypoints}$ 中剔除。

2.1.2 时间复杂度分析

边缘点剔除算法的时间复杂度由边缘点检测算法的时间复杂度和边缘关键点剔除算法的时间复杂度两部分组成,两者的时间复杂度分别为 $O[n \cdot (k + k \lg k)]$ 和 $O(n + m_1 m_2)$,其中 n 为点云中点的个数; k 为每个点的半径 r 范围内邻域点的个数; $k + k \lg k$ 为计算点与其邻域投影点夹角和对夹角进行排序找出最大夹角 L 的时间复杂度; m_1 为关键点的个数; m_2 为边缘点的个数。

2.2 误匹配关系剔除算法

一般情况下,三维点云间进行关键点匹配时,获取的匹配关系中均含有一定量的误匹配对,将对变换矩阵的计算产生严重的干扰,从而影响三维点云配准和识别的准确率。因此,需要剔除这些误匹配对。剔除误匹配对步骤分为:1) 利用 Kmeans 的思想,将匹配对进行聚类划分,保留聚类中匹配对个数最大的聚类,从而剔除部分误匹配对;2) 在获得较多的正确匹配对的基础上,利用分裂法,以标准差和欧氏距离作为标准,进一步剔除误匹配对。采用 Kmeans 算法与分裂法相结合进行误匹配对剔除的原因为:Kmeans 算法通过聚类,设定元素最多的聚类中的匹配对为正确的匹配对,而其他的聚类中的匹配对均被认定为错误的匹配对,这样能够快速剔除大量的误匹配对,实现对初始匹配对进行误匹配对剔除的粗加工;余下的匹配对因为在欧氏距离上比较接近,所有的匹配对均离散地分布在聚类中心(匹配对的均

值)的周围,故而使用聚类的方法难以再进行细分,但也正是这种分布状态,决定了每个匹配对与中心之间的偏离程度会有所不同。分裂法正是基于这种偏离程度,参照匹配对的标准差,只要匹配对与聚类中心偏离程度大于标准差的 τ 倍,就剔除该匹配对,如此不断分裂,最终实现对误匹配对剔除操作的细加工。

2.2.1 Kmeans 剔除误匹配关系对

利用 Mian^[21] 提出的方法,对获取的初始匹配关系集合 C_{crsp}' 中的每一对匹配关系,计算出一个旋转矩阵 \mathbf{R} 和平移向量 \mathbf{t} :

$$\mathbf{R} = \mathbf{F}_{\text{src}}^T \mathbf{F}_{\text{tgt}}, \quad (3)$$

$$\mathbf{t} = \mathbf{p}_{\text{tgt}} - \mathbf{p}_{\text{src}} \mathbf{R}, \quad (4)$$

式中 \mathbf{p}_{src} 与 \mathbf{p}_{tgt} 为一对匹配的关键点, \mathbf{F}_{src} 、 \mathbf{F}_{tgt} 为关键点的局部坐标系。

再将旋转矩阵 \mathbf{R} 转换成四元数,与平移向量 \mathbf{t} 可组成一个 7 维向量。据此, C_{crsp} 中的每一对关键点均可转化为一个 7 维向量。转换后匹配关系形成新的集合,记为 \mathbf{R}_{rts} , 假设其长度为 m 。

利用 Kmeans 算法对 \mathbf{R}_{rts} 进行聚类划分,将其划分成 κ (一般取 3~6) 个聚类,步骤如下:

1) 创建 κ 个聚类集合 $\{\mathbf{c}_{\text{cluster1}}, \mathbf{c}_{\text{cluster2}}, \dots, \mathbf{c}_{\text{cluster}\kappa}\}$ 和 κ 个聚类中心 $\{\mathbf{c}_{\text{center1}}, \mathbf{c}_{\text{center2}}, \dots, \mathbf{c}_{\text{center}\kappa}\}$, 每个聚类中心都是一个 7 维向量。

2) 对 \mathbf{R}_{rts} 中的 m 条向量顺序的分配 0 至 $m-1$ 的标号,并根据“标号 mod κ ”的值将每条向量置入到指定的聚类中。

3) 计算每个聚类中心 $\mathbf{c}_{\text{center}i}$

$$\mathbf{c}_{\text{center}i} = \frac{1}{M} \sum_{j=1}^M \mathbf{R}_{\text{rts}j}, \{\mathbf{R}_{\text{rts}j} \mid \mathbf{R}_{\text{rts}j} \in \mathbf{c}_{\text{cluster}i}\}. \quad (5)$$

4) 遍历 \mathbf{R}_{rts} 中每条向量 $\mathbf{R}_{\text{rts}i}$, 计算出与 $\mathbf{R}_{\text{rts}i}$ 距离最近的聚类中心 $\mathbf{c}_{\text{center}j}$, 将 $\mathbf{R}_{\text{rts}i}$ 从原本的聚类中移动到相应的 $\mathbf{c}_{\text{cluster}j}$ 中。

$$\mathbf{c}_{\text{cluster}j} = \arg \min_j \|\mathbf{R}_{\text{rts}i} - \mathbf{c}_{\text{center}j}\|. \quad (6)$$

5) 循环执行步骤 3) 和 4), 直到 \mathbf{R}_{rts} 中的向量在各个聚类间不再移动, 聚类划分停止。

包含匹配对数量最多的聚类作为最终结果保留下来, 记为 \mathbf{R}'_{rts} (与之相对应的匹配关系集合记为 C'_{crsp}), 其他聚类中的匹配对则被认定为误匹配对而将剔除。通过 Kmeans 方法, 可以初步大量地剔除误差较大的匹配关系, 提高正确匹配对的比率。

2.2.2 分裂法剔除误匹配对

由于 C'_{crsp} 集合中大部分是正确的匹配对, 均离

散地分布在聚类中心的周围, 因此可以借鉴非监督分类中分裂法的思想, 在开始时将所有元素看成一类, 将每个匹配对与中心之间的欧氏距离作为偏离程度的衡量标准, 以匹配对的标准差作为参考条件, 只要标准差大于规定的阈值, 就要分裂, 将偏离程度大的匹配对分离出去。具体步骤如下:

1) 设置参数, 最小标准差的阈值为 S_{TH} (本文 S_{TH} 的值设为 0.013), 剔除倍数 τ 。

2) 计算 \mathbf{R}'_{rts} 的中心 $\mathbf{c}_{\text{center}}$ 与标准差 \mathbf{S} , $\mathbf{c}_{\text{center}}$ 与 \mathbf{S} 都是 7 维向量。若 \mathbf{S} 中的最大分量值小于 S_{TH} , 则剔除步骤结束。

$$\mathbf{c}_{\text{center}} = \frac{1}{N} \sum_{i=1}^N \mathbf{R}'_{\text{rts}i}, \quad (7)$$

$$\mathbf{S} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{R}'_{\text{rts}i} - \mathbf{c}_{\text{center}})^2}. \quad (8)$$

3) 遍历 \mathbf{R}'_{rts} 中每一条向量 \mathbf{R}'_{rts} , 计算 \mathbf{R}'_{rts} 与 $\mathbf{c}_{\text{center}}$ 的距离 d_i , 若 $d_i > \tau \cdot \|\mathbf{S}\|$ 并且 \mathbf{R}'_{rts} 中向量的总数大于 3 (只有匹配对数量大于 3 时才能计算出变换矩阵), 则剔除该向量; 如若 \mathbf{R}'_{rts} 中向量的总数等于 3 时, 则剔除步骤结束。

$$d_i = \|\mathbf{R}'_{\text{rts}i} - \mathbf{c}_{\text{center}}\|. \quad (9)$$

4) 循环执行步骤 2)、3), 直到达到剔除程序结束的条件。

经过上述一系列步骤, 可以得到两个三维点云间的正确匹配关系, 进而可利用这些正确的匹配对计算出准确的三维点云间的变换矩阵, 用于配准或识别。

2.2.3 时间复杂度分析

剔除误匹配对的时间复杂度同样由 Kmeans 算法和分裂法的时间复杂度两部分组成, 分别为 $O(n_1 \cdot k_c \cdot m \cdot T_1)$ 和 $O(n_2 \cdot m \cdot T_2)$, 其中 n_1 和 n_2 分别为执行 Kmeans 算法和分裂法时初始的关键点匹配对的个数; k_c 为 Kmeans 算法的聚类数; m 为 \mathbf{R}'_{rts} 中任一向量的长度, 本文中值为 7; T_1 为 Kmeans 算法中步骤 5) 的循环次数; T_2 为分裂法中步骤 4) 的循环次数。

3 实验结果与分析

3.1 实验基础技术与软硬件环境

关键点匹配方法中的基础技术可采用当前一些成熟的、流行的方法, 其中涉及有: 1) 关键点检测采用 PCL 库中提供的尺度不变特征变换 (SIFT) Keypoint 算法, 该算法是对 Lowe^[22] 的二维 SIFT 算法进行的改进与扩展, 使其适用于三维点云数据

的处理; 2) 关键点描述采用 Tombari^[10] 提出的 SHOT 描述子, 其具有很强的抗噪、抗干扰能力, 运行效率也可满足要求; 3) 关键点的局部坐标系是依据 Petrelli^[23] 提出的方法建立的, 该方法在重复性方面优于其他方法, 并且计算效率非常高; 4) 初始的匹配关系可通过 K-D 树进行 NN 查找而得到, NN 算法的思想是: 假设集合 A 和 B, 对 A 中任一点 a 在 B 中查找最近点 b , 并为 b 在集合 A 中查找最近点 a' , 如果 a 与 a' 为同一点, 则最终认定集合 A 中 a 在集合 B 中的最近邻为 b , 否则集合 A 中 a 在集合 B 中未找到最近邻。NN 算法的时间复杂度为 $O[(n_3 \cdot n_4 \cdot s)^2]$, 其中 n_3 和 n_4 分别为集合 A 和 B 中关键点的个数; s 为 SHOT 描述子的维度, 其值为 352。

软硬件环境: 操作系统为 64 位 windows 7, CPU 为 Intel i7 3.6 GHz, 内存为 16 G, 开发环境为 Visual Studio 2013+PCL 1.8, 开发语言为 C++。

3.2 匹配关系评价标准

在实验分析之前首先定义评价标准。1) 正确匹配对。假设点云 M, M' , 关键点匹配对为 $(K_{\text{keys_srci}}, K_{\text{keys_tgti}})$, $K_{\text{keys_srci}}$ 经过真实 (GT) 矩阵旋转平移变换后为 $K'_{\text{keys_srci}}$; 若 $K'_{\text{keys_srci}}$ 与 $K_{\text{keys_tgti}}$ 的距离在 ν (本文 ν 的值设为 2) mr 内, 则认为这是一对正确的匹配对, 否则是错误的匹配对。2) 旋转平移误差。采用 Mian^[21] 提出的评价方法, 计算出旋转误差 θ_e 和平移误差 t_e :

$$\theta_e = \arccos \left[\frac{\text{tr}(\mathbf{R}\mathbf{R}_{\text{GT}}^{-1}) - 1}{2} \right] \frac{180}{\pi}, \quad (10)$$

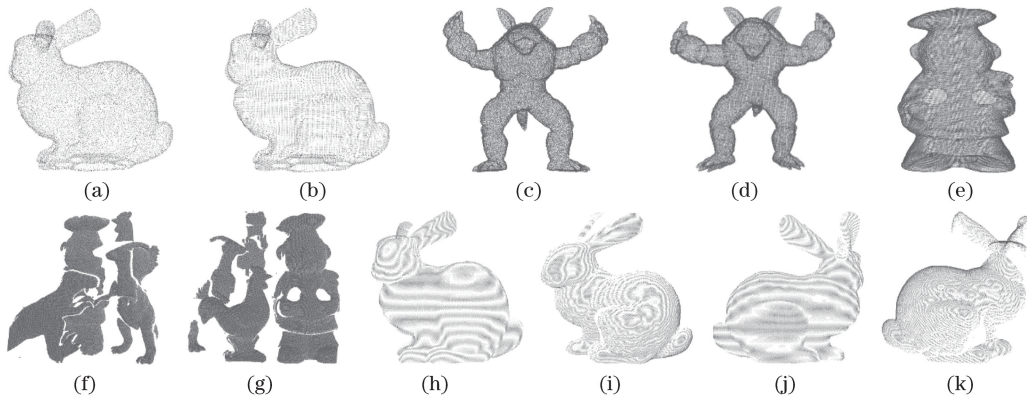


图 4 三维点云图例。(a) Bunny'; (b) Bunny''; (c) Armadillo'; (d) Armadillo''; (e) chef; (f) rs1; (g) rs4; (h) bun000; (i) bun045; (j) top2; (k) bun180

Fig. 4 Examples of 3D point cloud. (a) Bunny'; (b) Bunny''; (c) Armadillo'; (d) Armadillo''; (e) chef; (f) rs1; (g) rs4; (h) bun000; (i) bun045; (j) top2; (k) bun180

3.4 边缘关键点剔除实验

对三类数据集集中的所有三维点云作边缘点检测

$$t_e = \frac{\|t - t_{\text{GT}}\|}{d_{\text{res}}}, \quad (11)$$

式中 \mathbf{R} 和 t 为根据所有的匹配的关键点对, 利用 Arun^[24] 提出奇异值分解 (SVD) 方法计算出的旋转矩阵和平移向量, \mathbf{R}_{GT} 和 t_{GT} 分别为真实的旋转矩阵和平移向量, θ_e 是旋转误差, t_e 为平移误差, d_{res} 是三维点云的面片分辨率。

3.3 实验数据

采用 3 种不同类型的实验数据集进行关键点匹配, 以验证本文方法能否有效剔除关键点误匹配对。数据集分别为: 1) 完整三维点云与完整三维点云; 2) 完整三维点云与杂乱且有遮挡的三维点云; 3) 部分三维点云与部分三维点云。每一类型的数据实验集中均有两组三维点云对。在数据集 1) 中, 以斯坦福大学的标准三维模型库^[25] 中的 Armadillo 和 Bunny 为原型, 分别进行加噪和旋转平移变换而得到; 假设三维点云原型为 M , 将 M 添加方差为 0.3 倍面片分辨率的高斯噪声得到三维点云 M' ; 将 M 进行平移, 并且绕 x 、 y 和 z 轴分别旋转 15° 得到三维点云 M'' ; 2) 组三维点云对分别为 (Bunny', Bunny'') 和 (Armadillo', Armadillo'')。在数据集 2) 中, 采用 UWA^[5] 数据集集中的三维模型, 其中 chef 为完整三维点云, rs1 和 rs4 分别为杂乱且有遮挡的三维点云, 两组三维点云对分别为 (chef, rs1) 和 (chef, rs2)。在数据集 3) 中, 同样采用了斯坦福大学标准三维模型库中提供的不同视角下兔子的三维点云, 两组三维点云对分别为 (bun045, bun000) 和 (top2, bun180), 如图 4 所示。

以及边缘关键点剔除的对比分析。针对 Bunny', Bunny'', Armadillo', Armadillo'' 和 chef, 邻域半径为

5 mr;针对 rs1 和 rs4,邻域半径为 6 mr;剩下的三维点云,邻域半径为 7 mr。表 1 给出每一个三维点云的原始点数量、边缘点数量、边缘点比率(边缘点数量/原始点数量)、初始关键点(执行边缘关键点剔除操作之前的关键点)数量、最终关键点(执行边缘关键点剔除操作之后的关键点)数量、关键点剔除比率[(初始关键点数量-最终关键点数量)/初始关键点数量]、边缘关键点剔除耗时、边缘关键点剔除效率(边缘关键点剔除耗时/原始点数量)。图 5 分别对应图 4 中每个三维点云突出了边缘点、初始关键点和最终关键点的 3 种状态,图 5(a1)~(a3)对应图 4(a)的 3 种状态,其中红色且被放大的点分别为边缘点、初始关键点和最终关键点,其他均类似,不再赘述。

所有三维点云中,Bunny'、Bunny''、Armadillo'、Armadillo''和 chef 为完整三维点云,其结构具有封闭性,而 rs1、rs4、bun000、bun045、top2 和 bun180 为非完整三维点云,其结构非封闭,存在锯齿状的边

缘。根据三维点云结构是否封闭,以及图 5(a1)~(k1)可知,非封闭的三维点云的边缘点更为突出,大量集中在三维点云的边缘;由表 1 可知,非封闭的三维点云的边缘点比率明显高于封闭的三维点云,并且边缘关键点剔除效率高于封闭的三维点云。需要说明的是,在封闭的三维点云中,Armadillo'和 Armadillo''的边缘点比率更低,这是因为其他三个封闭的三维点云中存在孔洞,增加了边缘点的数量,导致边缘点比率升高。另外,通过对比观察图 5 中各三维点云的边缘关键点剔除前后的状态可知,非封闭的三维点云的边缘关键点更容易被剔除,因为非封闭的三维边缘点突出而且其边缘点比率更高,因此更多的关键点被认定为边缘关键点;表 1 中的关键点剔除率也佐证了这一结论。通过计算可知,平均边缘关键点剔除效率为 0.0257 ms,即 10000 点规模的点云,剔除其边缘关键点仅需要 257 ms,能够满足配准或识别对时间的要求。

表 1 三维点云中点的信息和删除边缘关键点的时间

Table 1 Information about points in the 3D point cloud and the time of removing keypoints on the edge

Point cloud	Number of original points	Number of edge points	Rate of edge points /%	Number of initial keypoints	Number of final keypoints	Ratio of removal /%	Time of removal /ms	Efficiency of removal /ms
Bunny'	35947	1013	2.82	2713	1564	42.35	577	0.016
Bunny''	35947	825	2.30	2508	1820	27.43	624	0.017
Armadillo'	172974	836	0.48	771	443	42.54	3120	0.018
Armadillo''	172974	188	0.11	702	656	6.55	3232	0.019
chef	176920	1314	0.74	835	670	19.76	3708	0.021
rs1	114364	13203	11.54	1394	291	79.12	3246	0.028
rs4	135666	13113	9.67	1162	331	71.51	3334	0.026
bun000	40256	2345	5.83	1135	508	55.24	1412	0.035
bun045	40097	2514	6.27	993	348	64.95	1399	0.035
top2	38298	2975	7.77	975	331	66.05	1305	0.034
bun180	40251	2393	5.95	1151	464	59.69	1376	0.034

3.5 误匹配关系剔除实验

为了验证本文算法的有效性,并与 NN、RANSAC 算法作对比分析,将 3 种算法分别融入到关键点匹配流程之中并进行匹配关系评价,对比旋转和平移误差。具体实验方案为:1)基于本文算法的关键点匹配;2)基于 NN 算法的关键点匹配;3)基于 RANSAC 算法的关键点匹配。方案 1)的流程参见图 2,其中特征匹配算法为 NN,误匹配剔除算法为本文算法。方案 2)与方案 1)类似,特征匹配算法为 NN,但不再进行误匹配剔除操作。方案

3)与方案 1)类似,特征匹配算法为 NN,误匹配剔除算法为 RANSAC,RANSAC 算法随机选取的关键点匹配对数量为 3,循环次数为 10000。每组方案均针对数据集 1)、2)和 3)进行关键点匹配。在针对数据集 1)的实验中,聚类个数 $\kappa=3$,剔除倍数 $\tau=2$;针对数据集 2)和 3)的实验中,聚类个数 $\kappa=6$,剔除倍数 $\tau=1$ 。

3.5.1 完整三维点云匹配完整三维点云

图 6、7 所示为该数据集下执行 3 种方案获取到的匹配关系图。因为三维点云之间存在旋转关系,

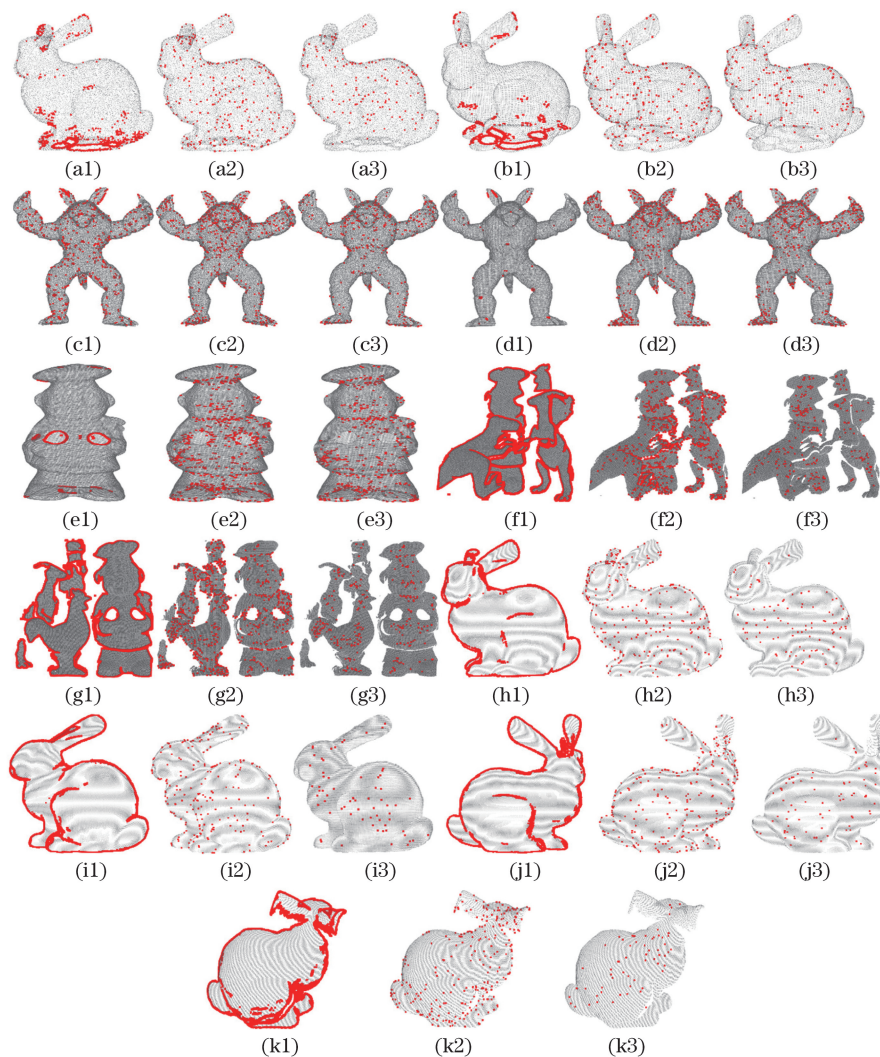


图 5 三维点云状态图

Fig. 5 State chart of 3D point cloud

直接得到的匹配关系连线有交叉,效果并不直观,所以将结果根据真实矩阵旋转使其平行再进行展示。此时,如果两个三维点云之间的平行线越多,则说明正确的匹配对越多。表 2 所示为正确匹配对数量和总匹配对数量,以及对应的旋转和平移误差。

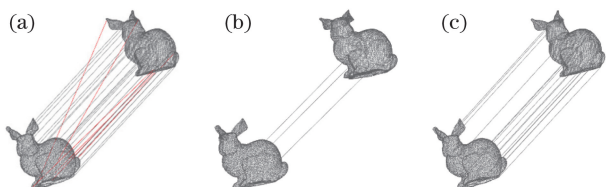


图 6 Bunny'与 Bunny''匹配关系图。(a) NN 算法匹配;

(b) RANSAC 算法匹配;(c)本文算法匹配

Fig. 6 Matching graphs between Bunny' and Bunny''.

(a) Matching by NN algorithm; (b) matching by RANSAC algorithm; (c) matching by proposed algorithm

从图 6(a)和图 7(a)可以看出,直接经过 NN 算法查找到的关键点匹配对存在交叉,即含有一部分

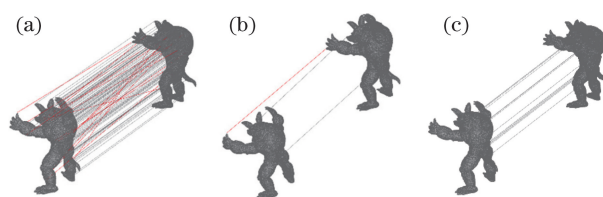


图 7 Armadillo'与 Armadillo''匹配关系图。

(a) NN 算法匹配;(b) RANSAC 算法匹配;

(c) 本文算法匹配

Fig. 7 Matching graphs between Armadillo' and Armadillo''.

(a) Matching by NN algorithm; (b) matching by RANSAC algorithm; (c) matching by proposed algorithm

错误匹配,其中红色连线表示错误的匹配对,黑色连线表示正确的匹配对。从图 6(b)和图 7(b)也可以看到,RANSAC 算法寻找到的最优匹配关系较少,其原因是 RANSAC 算法的随机性无法保证选取到最好的结果。从图 6(c)和图 7(c)可以看出,本文算法匹配结果中,点云之间仅保留正确的匹配关系,效

果较理想。同时从表 2 也可以看出,在完整三维点云匹配完整点云数据集中,基于本文算法的三维点

云关键点匹配方法误差最小、匹配效果最好,优于 NN 和 RANSAC 算法。

表 2 完整三维点云与完整三维点云间的匹配结果

Table 2 Matching results between complete 3D point cloud and complete 3D point cloud

Point cloud 1	Point cloud 2	Algorithm	Number of correct matching pairs	Number of all matching pairs	Error of rotation	Error of translation
Bunny'	Bunny''	NN	25	31	6.098	14.364
		RANSAC	3	3	0.197	0.344
		Proposed	16	16	0.181	0.109
Armadillo'	Armadillo''	NN	96	112	2.126	8.574
		RANSAC	2	3	1.107	1.668
		Proposed	77	77	0.285	0.188

3.5.2 完整三维点云匹配杂乱且有遮挡的三维点云

图 8、9 所示为该数据集下执行 3 种方案获取到的匹配关系图,表 3 所示为正确匹配对数量和总匹配对数量,以及对应的旋转和平移误差。从图 8(a)和图 9(a)可以看出,通过 NN 算法查找到的关键点匹配对中含有大量的误匹配,故而求出的旋转平移误差较大,表 3 中的误差值也给出了有力的佐证;而

RANSAC 算法在此环境中匹配效果亦不佳,其原因在于初始误匹配对比例偏高,RANSAC 算法难以随机选取到正确的匹配关系;而本文算法匹配结果中,匹配关系较好、误差小,剔除了大量的误匹配对。因此,通过图 8、9 以及表 3 的结果可知,在完整三维点云匹配杂乱且有遮挡的三维点云中,本文算法能够较好地剔除关键点误匹配对,提升匹配效果,优于 NN 和 RANSAC 算法。

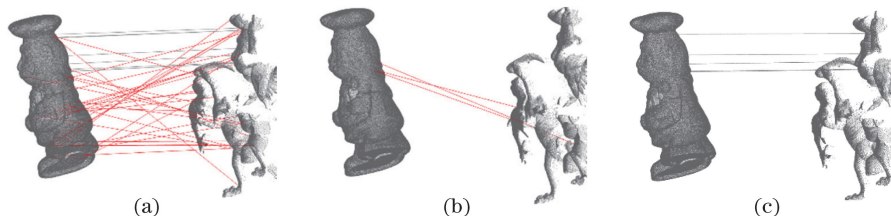


图 8 chef 与 rs1 匹配关系图。(a) NN 算法匹配;(b) RANSAC 算法匹配;(c)本文算法匹配

Fig. 8 Matching graphs between chef and rs1. (a) Matching by NN algorithm; (b) matching by RANSAC algorithm; (c) matching by proposed algorithm

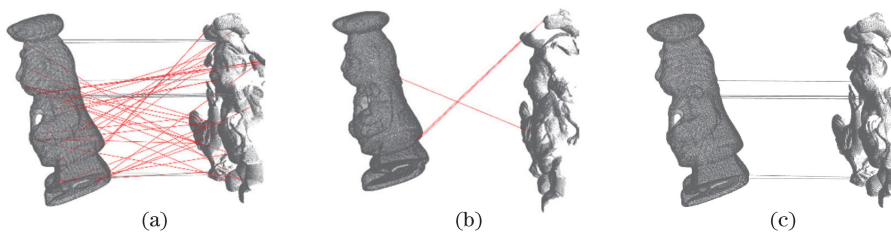


图 9 chef 与 rs4 匹配关系图。(a) NN 算法匹配;(b) RANSAC 算法匹配;(c)本文算法匹配

Fig. 9 Matching graphs between chef and rs4. (a) Matching by NN algorithm; (b) matching by RANSAC algorithm; (c) matching by proposed algorithm

3.5.3 部分点云匹配部分点云

图 10、11 所示为该数据集下执行 3 种方案获取到的匹配关系图,表 4 所示为正确匹配对数量和总匹配对数量,以及对应的旋转和平移误差。从图 10(a)~(c)和图 11(a)~(c)可以看出,本文算法的三维点云关键点匹配效果十分理想,基本剔除了所有误匹配关系;而 RANSAC 算法的匹配结果较

差,其原因仍在于初始误匹配对比例偏高,RANSAC 算法难以随机选取到正确的匹配关系,表 4 的误差数据也佐证了这一结论。通过图 10、11 以及表 4 的结果可知,在部分三维点云匹配部分三维点云中,本文算法能够较好地剔除关键点误匹配对,匹配效果最好,其次是 NN 算法,最差是 RANSAC 算法。

表 3 完整三维点云与杂乱且有遮挡的三维点云的匹配结果

Table 3 Matching results between complete 3D point cloud and 3D point cloud with clutter and occlusion

Point cloud 1	Point cloud 2	Algorithm	Number of correct matching pairs	Number of all matching pairs	Error of rotation	Error of translation
chef	rs1	NN	15	58	104.080	1447.870
		RANSAC	0	3	32.585	426.013
		Proposed	12	12	1.489	18.945
chef	rs4	NN	20	72	88.519	1245.380
		RANSAC	0	3	9.911	159.958
		Proposed	9	9	3.167	43.351

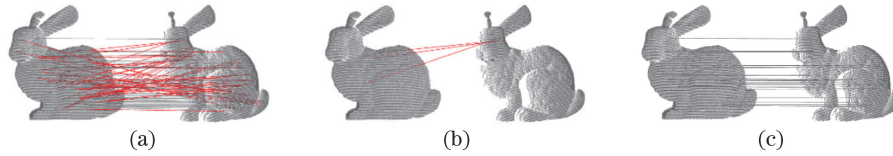


图 10 bun045 与 bun000 匹配关系图。(a) NN 算法匹配;(b) RANSAC 算法匹配;(c) 本文算法匹配

Fig. 10 Matching graphs of bun045 and bun000. (a) Matching by NN algorithm; (b) matching by RANSAC algorithm; (c) matching by proposed algorithm

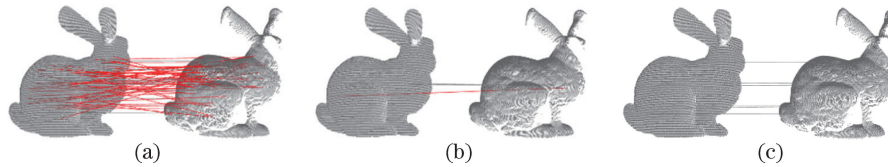


图 11 top2 与 bun180 匹配关系图。(a) NN 算法匹配;(b) RANSAC 算法匹配;(c) 本文算法匹配

Fig. 11 Matching graphs of top2 and bun180. (a) Matching by NN algorithm; (b) matching by RANSAC algorithm; (c) matching by proposed algorithm

表 4 部分 3D 点云间的匹配结果

Table 4 Matching results between partial 3D point cloud and partial 3D point cloud

Point cloud 1	Point cloud 2	Algorithm	Number of correct matching pairs	Number of all matching pairs	Error of rotation	Error of translation
bun045	bun000	NN	49	97	2.5540	5.385
		RANSAC	0	3	7.9000	11.378
		Proposed	20	20	0.7160	0.962
top2	bun180	NN	12	61	31.9090	75.135
		RANSAC	2	3	95.4290	158.701
		Proposed	6	7	0.5203	0.875

3.6 关键点匹配时间实验

对比了 NN 算法、RANSAC 算法和本文算法的匹配结果,统计了关键点匹配所需的时间。由于 3 种匹配方法具有相同的关键点提取和关键点描述,对于相同三维点云匹配所需的时间也相同,因此关键点匹配的快慢由 NN 算法、RANSAC 算法和本文算法来决定。表 5 给出了 3 类数据集基于不同算

法的匹配时间。

由表 5 可知,NN 算法速度最快,其次为本文算法,最慢为 RANSAC 算法。其原因是 RANSAC 算法和本文算法的执行过程中都需包含 NN 算法操作以获取初始的匹配对。而且 RANSAC 算法需要执行大量的迭代,以获取较好的匹配对,实现局内点的最大化,这本质上来说是以牺牲时间效率来换取算

法的可靠性。

表 5 匹配时间对比

Table 5 Comparison of matching time

Point cloud 1	Point cloud 2	Algorithm	Time of keypoint matching /ms
Bunny'	Bunny''	NN	3343
		RANSAC	9294
		Proposed	4550
Armadillo'	Armadillo''	NN	10234
		RANSAC	22680
		Proposed	16692
chef	rs1	NN	15369
		RANSAC	20570
		Proposed	22332
chef	rs4	NN	15248
		RANSAC	23031
		Proposed	21298
bun045	bun000	NN	4699
		RANSAC	14203
		Proposed	6518
top2	bun180	NN	4413
		RANSAC	10368
		Proposed	7097

4 结 论

提出了一种剔除三维点云中关键点误匹配对的方法,将两种方法相结合来剔除关键点的误匹配。在关键点获取阶段,进行了边缘关键点剔除的操作,避免因局部点云数据信息不足而影响关键点的描述,从而降低在关键点特征匹配阶段产生较多的误匹配关键点对的可能性。同时在特征匹配之后,执行了剔除误匹配的操作:先利用 Kmeans 提高正确匹配关系的占比,然后利用分裂法,以标准差和距离为评价标准,将误匹配对全部剔除。实验结果表明,在完整三维点云匹配完整三维点云、完整三维点云匹配杂乱且有遮挡的三维点云、部分点云匹配部分点云等实验环境下,本文算法均能大量的剔除误匹配关键点对,相比 NN 算法和 RANSAC 算法具有更高的匹配效果;在关键点匹配时间实验中,本文算法比 RANSAC 算法时间效率更高,是 NN 算法的有效补充。但是本文算法也存在剔除过量的问题,部分正确匹配对会被作为误匹配对而删除,这也是下一步需要改进之处。

参 考 文 献

[1] Shu C X, He Y T, Sun Q K. Point cloud registration

based on convolutional neural network[J]. *Laser & Optoelectronics Progress*, 2017, 54(3): 031001.

舒程珣, 何云涛, 孙庆科. 基于卷积神经网络的点云配准方法[J]. *激光与光电子学进展*, 2017, 54(3): 031001.

[2] Zhang Q S, Zhou Y, Hu X M, *et al.* Hand vein recognition based on three dimensional point clouds matching[J]. *Acta Optica Sinica*, 2015, 35(1): 0115005.

张祺深, 周雅, 胡晓明, 等. 基于三维点云匹配的手掌静脉识别[J]. *光学学报*, 2015, 35(1): 0115005.

[3] Chen H, Bhanu B. 3D free-form object recognition in range images using local surface patches[J]. *Pattern Recognition Letters*, 2007, 28(10): 1252-1262.

[4] Zhong Y. Intrinsic shape signatures: a shape descriptor for 3D object recognition[C]. 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, 2009: 689-696.

[5] Mian A, Bennamoun M, Owens R. On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes[J]. *International Journal of Computer Vision*, 2010, 89(2/3): 348-361.

[6] Zaharescu A, Boyer E, Varanasi K, *et al.* Surface feature detection and description with applications to mesh matching[C]. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009: 373-380.

[7] Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration[C]. 2009 IEEE International Conference on Robotics and Automation, 2009: 3212-3217.

[8] Guo Y L, Sohel F, Bennamoun M, *et al.* Rotational projection statistics for 3D local surface description and object recognition[J]. *International Journal of Computer Vision*, 2013, 105(1): 63-86.

[9] Salti S, Tombari F, Stefano L D. SHOT: unique signatures of histograms for surface and texture description[J]. *Computer Vision and Image Understanding*, 2014, 125(8): 251-264.

[10] Tombari F, Salti S, Stefano L D. Unique signatures of histograms for local surface description[C]. *European Conference on Computer Vision*, 2010, 6313: 356-369.

[11] Mikolajczyk K, Schmid C. A performance evaluation of local descriptors[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, 27(10): 1615-1630.

[12] Yang B S, Zang Y F. Automated registration of dense terrestrial laser-scanning point clouds using curves[J]. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2014, 95(3): 109-121.

[13] He B W, Lin Z M, Li Y F. An automatic registration algorithm for the scattered point clouds based on the

- curvature feature[J]. *Optics & Laser Technology*, 2013, 46(1): 53-60.
- [14] Besl P J, McKay N D. A method for registration of 3D shapes[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992, 14(2), 239-256.
- [15] Chen Y, Medioni G. Object modelling by registration of multiple range images[J]. *Image and Vision Computing*, 1992, 10(3): 145-155.
- [16] Chen M L, Lu W L, Wan Y C, *et al.* Automatic registration of terrestrial point clouds without additional information[J]. *Chinese Journal of Lasers*, 2016, 43(4): 0414003.
陈茂霖, 卢维欣, 万幼川, 等. 无附加信息的地面激光点云自动拼接方法[J]. *中国激光*, 2016, 43(4): 0414003.
- [17] Taati B, Greenspan M A. Local shape descriptor selection for object recognition in range data[J]. *Computer Vision and Image Understanding*, 2011, 115(5): 681-694.
- [18] Papazov C, Haddadin S, Parusel S, *et al.* Rigid 3D geometry matching for grasping of known objects in cluttered scenes[J]. *International Journal of Robotics Research*, 2012, 31(4): 538-553.
- [19] Xu G Z, Ruan P. Point cloud boundary detection in preprocessor of optical-mechanical integrated simulation[J]. *Infrared and Laser Engineering*, 2016, 45(4): 0428001.
徐广州, 阮萍. 光机集成仿真前处理中点云边界检测技术[J]. *红外与激光工程*, 2016, 45(4): 0428001.
- [20] 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.
- [21] Mian A S, Bennamoun M, Owens R A. A novel representation and feature matching algorithm for automatic pairwise registration of range images[J]. *International Journal of Computer Vision*, 2006, 66(1): 19-40.
- [22] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. *International Journal of Computer Vision*, 2004, 60(2): 91-110.
- [23] Petrelli A, Stefano L D. On the repeatability of the local reference frame for partial shape matching[C]. 2011 IEEE International Conference on Computer Vision, 2011, 24(4): 2244-2251.
- [24] Arun K S, Huang T S, Blostein S D. Least-squares fitting of two 3D point sets[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1987, 9(5): 698-700.
- [25] Curless B, Levoy M. A volumetric method for building complex models from range images[C]//*Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Technique*, 1996, 3: 303-312.