

基于自适应分辨率的点对特征点云配准算法

方博,岳晓峰,袁晓磊,马国元
(长春工业大学 机电工程学院,吉林 长春 130012)

摘要:针对基于原始点对特征的三维目标识别算法中存在的内存浪费及过多冗余点导致的配准效率不高问题,提出一种改进的自适应分辨率点云配准算法。通过使用自适应体素网格滤波器对原始点云进行下采样,有效地减少了目标模型哈希表中存在的大量冗余点对,节省了计算机的内存开销并提高了点云配准算法的配准速度。使用迭代最近点方法进一步对聚类结果进行优化处理。实验结果表明:所用算法与传统 ICP 算法相比,在配准误差相同的情况下配准效率提升显著。

关键词:自适应分辨率;稠密点云;体素网格滤波器;哈希表;迭代最近点算法

中图分类号:TP311.5 **文献标志码:**A **文章编号:**1671-5276(2022)03-0113-04

Point-to-feature Point Cloud Registration Algorithm Based on Adaptive Resolution

FANG Bo, YUE Xiaofeng, YUAN Xiaolei, MA Guoyuan

(School of Mechanical and Electrical Engineering, Changchun University of Technology,
Changchun 130012, China)

Abstract: Aimed at the memory waste and the low registration efficiency caused by excessive redundant points in 3D target recognition algorithm based on the original point pair feature, an improved point cloud registration algorithm with adaptive resolution is proposed. An adaptive voxel grid filter is applied to down-sample the original point cloud, which effectively reduces a large number of redundant point pairs in the target model hash table, saves computer memory overhead and improves the point cloud registration algorithm. Iteration closest point is used to further optimize the clustering results. The experimental results show that the proposed algorithm improves registration efficiency significantly under the same registration error in comparison with the traditional ICP algorithm.

Keywords: adaptive resolution; dense point cloud; voxel grid filter; Hash table; ICP

0 引言

点云数据是三维目标模型的重要空间结构数据,由点云驱动的计算机图形学在逆向工程、数字城市、智能机器人、无人驾驶和人机交互等领域发挥着越来越重要的作用^[1-2]。

针对点云配准这一关键性问题,迭代最近点算法(iteration closest point, ICP)^[3]是目前应用最广的点云配准算法,国内外学者多采用特征点粗配准来获取点云间良好的初始配准位置,再采用 ICP 算法进行精确配准。针对特征点的提取,CHEN Y 等^[4]采用待配准点云的点法线与参考点点云的交点来确定对应点,该方法能减少迭代次数并加快算法收敛速度,但在某些情况下鲁棒性较差。RUSU R B 等^[5]提出快速特征直方图(FPFH)特征,目的是通过点周围的平均曲率来编码其邻域的几何属性,该算法能很好地应对邻域中存在不同采样密度或不同噪声水平的点,进而提升配准精度。RUSINKIEWICZ S 等^[6]提出基于法线空间的均匀采样方法,在具有较少特征点的点云

进行配准时,该方法可以提高配准速度。杨玲等^[7]将普氏分析法与 ICP 算法结合,提出了 PICP(procrustes ICP)算法,在获得点云初始位置的情况下,采用普氏分析法求解转换参数,针对不同的点集均获得较好的鲁棒性。随着提取场景点云的复杂程度、稠密程度提升带来的噪声、遮挡等问题,特征点提取的难度及特征描述子的计算量也随之提升,计算效率也因此下降。

本文针对点云配准效率问题进行研究,提出了一种易于实现且鲁棒性高的多分辨率点对特征配准方法。在初始阶段通过本文提出的自适应体素网格滤波器对原始点云进行自适应分辨率采样,合理、有效地减少了点云中的冗余点对,可节省内存开销并提高计算效率。位姿聚类结束后,再通过迭代最近点进行位姿优化提升算法的准确性,进一步改善了原始点对特征算法的识别性能。

1 点对特征

点对特征(point pair feature, PPF)描述的是一对有方向的点的相对位置和姿态,其中包含有 4 种简单几何属性

基金项目:吉林省发展和改革委员会项目(2020C018-3)

第一作者简介:方博(1994—),男,山西海泉人,硕士研究生,研究方向为机器视觉与能检测。

构成的四元数组。如图 1 所示,假设有两个点 m_1 和 m_2 , 法向量分别为 \mathbf{n}_1 和 \mathbf{n}_2 。PPF^[8] 定义为

$\mathbf{F}(m_1, m_2) = (\|d\|_2, \angle(\mathbf{n}_1, d), (\mathbf{n}_2, d), (\mathbf{n}_1, \mathbf{n}_2))$ (1)
式中: $d = m_2 - m_1$; \mathbf{F} 表示 m_1, m_2 的点对特征; $\angle(\mathbf{a}, \mathbf{b})$ 表示两向量间的夹角。由定义可知, \mathbf{F} 是不对称的, 即 $\mathbf{F}(m_1, m_2) \neq \mathbf{F}(m_2, m_1)$ 。

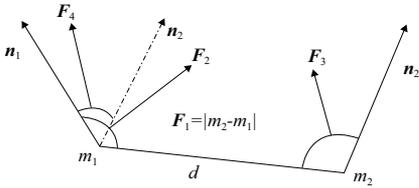


图 1 点对特征图

上述 \mathbf{F} 即为一对点的点对特征, 在点对特征算法中, 需要计算 model 表面所有点对的特征矢量 \mathbf{F} 。

2 体素网格方法

目前, 随着点云采集仪器精度的提高, 需要处理和计算的点云数目不断增加, 传统点对特征算法已经无法适应点云数目巨大的模型点云处理。下采样时, 通常会尽可能地保留源点云的表面特征和保证一定质量的密度分布, 否则将不利于点云的后续处理。若使用随机采样方法, 则点云的轮廓信息会变弱, 且点云分布也会不均匀。而使用体素网格滤波器对源点云进行下采样, 在减少点云数量的同时, 还可以保持源点云的形态特征。

体素网格下采样^[9-13]的主要思路是: 根据原始点云数据创建一个三维体素栅格, 计算出需要划分的三维立方栅格的边长 L , 然后计算每个立方体内所有点的质心来替代这个立方体栅格中的所有点。该方法从整体上对点云数量进行简化, 并可以保持源点云的形态特征。具体计算方法如下所示。

1) 求出初始点云模型中的最大、最小值顶点 $(x_{\max}, y_{\max}, z_{\max})$ 和 $(x_{\min}, y_{\min}, z_{\min})$ 。以这两个顶点构建一个包围盒, 在顶点 $(x_{\min}, y_{\min}, z_{\min})$ 上固定一个坐标系, 方向与欧氏空间的 (x, y, z) 方向相同, 构成体素空间 Z^3 。

2) 确定立方体边长 L , 并将点云分成多个大小为 $L \times L \times L$ 的立方体。 L 的值越小, 采样后的点云分辨率越高; 值越大, 采样后的点云分辨率则越低。采样立方体的个数为 $A \times B \times C$, 计算方法如下:

$$\begin{cases} A = \frac{|x_{\max} - x_{\min}|}{L} \\ B = \frac{|y_{\max} - y_{\min}|}{L} \\ C = \frac{|z_{\max} - z_{\min}|}{L} \end{cases} \quad (2)$$

3) 计算每个立方体的质心, 计算方法如下:

$$\begin{cases} x' = \frac{1}{m} \sum_{i=1}^m x_i \\ y' = \frac{1}{m} \sum_{i=1}^m y_i \\ z' = \frac{1}{m} \sum_{i=1}^m z_i \end{cases} \quad (3)$$

式中 m 为该立方体中点的数目。

3) 以质心 (x', y', z') 代替该立方体中所有的点, 并将所有立方体的质心重新整合成新的点云。

4) 由式(2)不难看出, 采样立方体的个数, 受立方体边长 L 的影响, 而传统体素化网格中的边长 L 在下采样时需要手动设定体素立方体的边长, 这就降低了算法效率。本文提出了一种改进的自适应体素网格滤波器, 可以根据源点云中点云的数量自动计算体素网格的边长, 并可根据自适应系数的调节采样分辨率, 以适应不同尺度的点云集合。

3 自适应体素网格滤波器

针对不同数量的点云, 本文改进了体素网格滤波器, 对边长 L 的立方体运用了自适应采样分辨率更新策略, 方法如下:

$$L = D_{\text{avg}} \times \omega(s) \quad (4)$$

式中: D_{avg} 表示初始输入点云中两点之间的平均距离; $\omega(s)$ 为边长 L 的自适应比例因子, 用以控制边长 L 的增长率。计算方法分别如下:

$$D_{\text{avg}} = \frac{1}{M} \sum_{i=1}^M \sqrt{\|N_i - Nd_i\|^2} \quad (5)$$

式中: M 为初始点云中点云的总数; N_i 为 M 中的第 i 个点; Nd_i 为 M 中离 N_i 距离最近的点。

$$\omega(s) = \frac{e}{\sqrt{\frac{e^s - e^{-s}}{e^s + e^{-s}} + 1}} \quad (6)$$

式中 $s = \sqrt{1/M}$ 。

在立方体边长计算中引入式(6)用以控制立方体边长的增长率。由式(6)可知, 当初始输入点云的数量越多, $\omega(s)$ 的值就越大, 立方体边长自适应变大, 点数减少更快; 当初始输入点云的数量少时, $\omega(s)$ 的值就会变小, 立方体边长自适应变小, 点数减少速度就放慢, 实现了自适应的分辨率调节, 不仅可以有效地减少点云中的冗余点对, 还提高了整体点云配准算法的效率。在加入激活函数后, 实现了立方体边长的自适应控制, 更加简便快捷。

4 算法的实现

本文提出的算法, 在离线训练阶段将点云下采样后得到的具有相同点对特征 \mathbf{F} 的点对集合在一起, 即哈希表的键(key)实际上为 \mathbf{F} , 内存的值为具有相同特征矢量的点集 \mathbf{A} ; 在线识别阶段, 随机选取一点作为种子点 p_s , 搜索与之距离不超过半径 r 的点, 组成点集 $\mathbf{Q}(p_s) = \{q_1, q_2, \dots, q_k\}$, 计算 p_s 与 $\mathbf{Q}(p_s)$ 中每个点的点对特征 \mathbf{F} , 并根据哈希值查找与之匹配的候选点对集合 $\mathbf{D}[\mathbf{F}(p_s, p_k)]$ 。通过场景点对和哈希表中查找到的候选点对进行匹配, 采用投票策略, 选取投票较多的位姿组成新的集合, 得到初步候选位姿集合 \mathbf{C} ; 采用聚类策略对候选位姿集合进行处理, 从中提取出正确的位姿; 最后返回中进行位姿优化得到最终位姿作为目标点云配准的结果。算法流程图如图 2 所示。

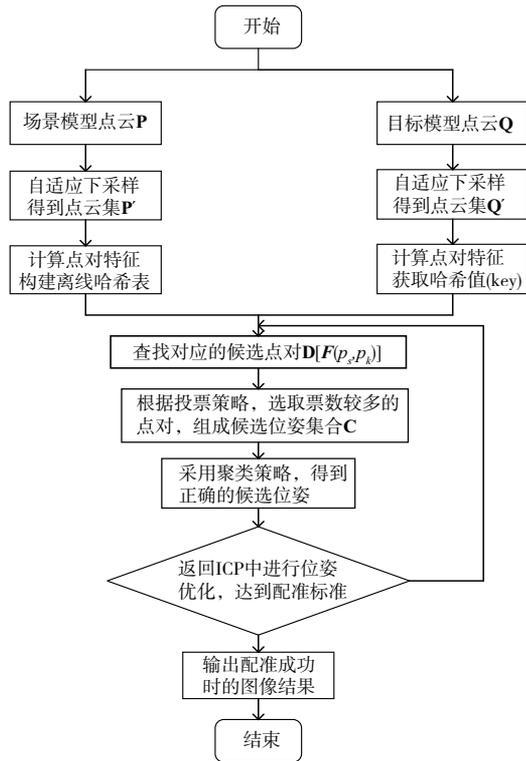


图2 配准流程图

5 实验结果与分析

本文采用的实验平台为 Window10 系统, Visual Studio2019, 配置为启天 M4500-N000, 酷睿™ i5-4590, CPU 3.8 GHz, 8 GB DDR3 的联想台式计算机上。算法没有依赖 GPU 等其他的加速优化工具, 为验证算法的有效性和鲁棒性, 分别与传统 ICP 算法和文献[14]中算法的配准时间和精度进行比较, 实验使用 Stanford University 开放点云数据库的 Bunny (兔子) 和 Dragon (龙) 模型进行对比实验。

图3是兔子和龙的初始点云模型。图4分别为兔子和龙模型在不同分辨率下的点云模型, 由式(4)~式(6)计算可得自适应边长 L 。下采样后的点数设置为不低于 3 000 个。图4(a)为兔子模型下采样一次, 自适应系数为 2.53; 图4(b)为龙模型下采样, 自适应系数为 2.51。表1为兔子和龙在自适应下采样下的采样点数和所耗费时长。下采样结果表明, 使用本文所提自适应下采样体素滤波器采样得到的点云模型, 即使在大量减少点云点数之后, 仍可以保留原始点云的形态特征, 且自适应策略下的下采样所耗费的时间也是极短的。

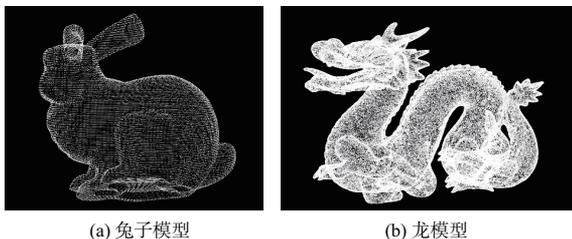


图3

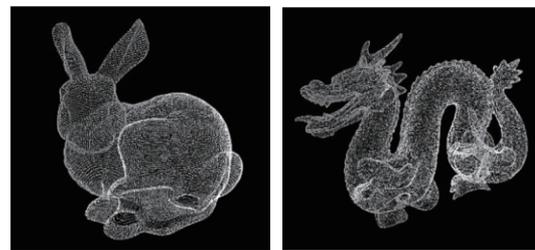


图4 模型简化图

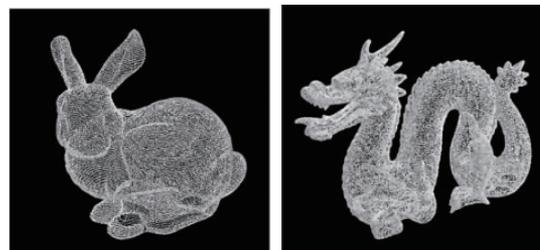
表1 自适应下采样计算耗时和点云简化率

点云名称	原始点云个数/个	自适应系数	体素网格边长	采样后点数	点云简化率/%	简化时间/s
兔子	35 947	2.53	0.005	3 017	92	0.008
龙	437 645	2.51	0.005	4 030	99	0.020

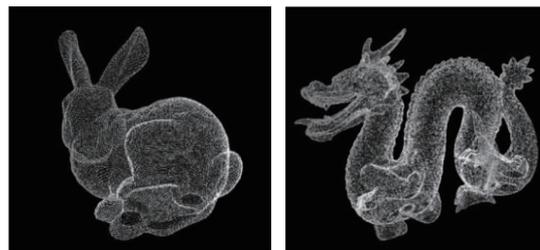
图5为兔子和龙的配准实验结果。可以看出文献[14]的匹配结果效果略差, 部分地方出现了重影, 传统 ICP 算法和本文所提算法配准结果基本一致, 但配准时间花费较长。



(a) 传统ICP算法匹配结果



(b) 文献[14]配准算法匹配结果



(c) 本文改进配准算法匹配结果

图5 配准结果

表2所示为3种算法在兔子和龙模型中的配准时间之间的比较。由表中数据表明: 在配准精度相当的情况下, 本文算法所需时间相对于传统 ICP 算法和文献[14]大大缩短。尤其是在点云较稠密, 数量较多的龙模型中, 更能体现本文算法的快速性和准确性。原始 ICP 算法由

于点数过多,导致配准时长迅速增长,文献[14]虽然将时间缩短至原始 ICP 算法的 2/3,但是耗时仍然很长。

表 2 配准误差相近的情况下算法性能比较

点云模型	点云数目	配准算法	配准误差 $\times 10^{-4}/\text{m}$	配准耗时/s
兔子	35 947	传统 ICP	2.46	2.677
		文献[14]	2.48	1.792
		本文算法	2.48	0.953
龙	437 645	传统 ICP	2.49	114.676
		文献[14]	2.49	63.348
		本文算法	2.46	20.237

为了进一步验证本算法在准确性和省时性上的提升,选取 4 组不同点云数目的点云进行配准实验,配准误差的要求依然不变。在保证配准精度的同时,再次将本文算法、文献[14]和原始传统点对特征算法进行比较,如表 3 所示。本文算法在点云数目较少时,相较于传统 ICP 算法,耗时仅为原先的 36%;在点云数目较多时,相较于传统 ICP 算法,耗时仅为原始 ICP 的 17%。且在各种规模的点云模型中,耗时均少于文献[14]。结合图 6 实验结果表明,本文所提出的自适应分辨率的点对特征配准算法在保证配准精度的同时,还可以快速收敛。

表 3 不同稠密度点云的配准耗时 单位:s

点云数目	配准耗时		
	传统 ICP	文献[14]	本文算法
23 958	2.498	1.278	0.817
67 897	7.012	3.237	1.612
236 708	42.97	20.556	8.267
537 847	141.527	74.548	23.178

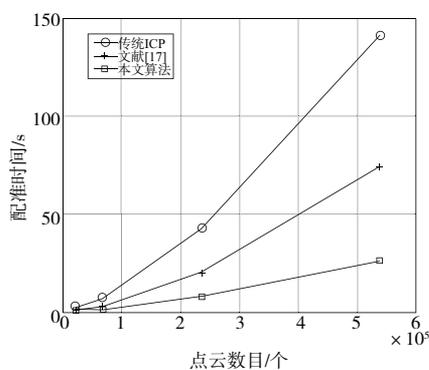


图 6 不同方法配准耗时

6 结语

本文提出了一种改进的自适应分辨率点对特征配准算法。算法通过构造的自适应分辨率体素化网格方法,对原始点云进行下采样处理,在减少冗余点云数量的同时,

有效地保留了点云的原始特征,提升了配准的效率。实验表明:本文算法与传统 ICP 算法相比,在配准误差相同的情况下,配准效率提升显著。随着点云数量和稠密度的提升,本文算法的优势将呈递增的趋势。

参考文献:

- [1] 郭浩.点云库 PCL 从入门到精通[M].北京:机械工业出版社,2019:1-4.
- [2] LIN J R, ZHANG F. Loam livox: a fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV[C]//2020 IEEE International Conference on Robotics and Automation (ICRA).Paris, France: IEEE,2020,3126-3131.
- [3] 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.
- [4] CHEN Y, MEDIONI G. Object modeling by registration of multiple range images [C]//Proceedings of 1991 IEEE International Conference on Robotics and Automation. Sacramento, CA, USA: IEEE,1991:2724-2729.
- [5] RUSU R B, BLODOW N, BEETZ M. Fast point feature histograms (FPFH) for 3D registration [C]//2009 IEEE International Conference on Robotics and Automation. Kobe, Japan: IEEE, 2009:3212-3217.
- [6] RUSINKIEWICZ S, LEVOY M. Efficient variants of the ICP algorithm[C]//Proceedings Third International Conference on 3-D Digital Imaging and Modeling. Quebec City, QC, Canada: IEEE,2001:145-152.
- [7] 杨玲, 谯舟三, 陈玲玲, 等. 结合 Procrustes 分析法和 ICP 算法的 PICP 配准算法[J]. 计算机辅助设计与图形学学报,2017,29(2):337-343.
- [8] 鲁荣荣, 朱枫, 吴清潇, 等. 基于增强型点对特征的三维目标识别方法[J]. 光学学报,2019,39(8):245-254.
- [9] YU HF, WANG R, CHEN J L, et al. Saliency computation and simplification of point cloud data[C]//Proceedings of 2012 2nd International Conference on Computer Science and Network Technology. Changchun, China: IEEE,2012:1350-1353.
- [10] XIAO Z X, HUANG W M. Kd-tree based nonuniform simplification of 3D point cloud[C]//2009 Third International Conference on Genetic and Evolutionary Computing. Guilin, China: IEEE,2009:339-342.
- [11] ORTOS-ESCOLANO S, MORELL V, GARCÍA-RODRÍGUEZ J, et al. Point cloud data filtering and downsampling using growing neural gas [C]//The 2013 International Joint Conference on Neural Networks (IJCNN). Dallas, TX, USA: IEEE,2013:1-8.
- [12] 姜媛媛. 基于超体素区域增长的点云分割算法研究[D]. 西安:西安电子科技大学,2017.
- [13] BENHABILES H, AUBRETON O, BARKI H, et al. Fast simplification with sharp feature preserving for 3D point clouds[C]//2013 11th International Symposium on Programming and Systems (ISPS). Algiers, Algeria: IEEE, 2013:47-52.
- [14] 杨秋翔, 王程远, 杨剑, 等. 基于法矢夹角的改进 ICP 算法[J]. 计算机工程与设计,2016,37(8):2082-2086.

收稿日期:2020-12-29