

## 三维点云中关键点的配准与优化算法

宋涛<sup>1,2\*\*</sup>, 曹利波<sup>1,3\*</sup>, 赵明富<sup>1,2</sup>, 刘帅<sup>1</sup>, 罗宇航<sup>1</sup>, 杨鑫<sup>1</sup>

<sup>1</sup>重庆理工大学电气与电子工程学院, 重庆 400054;

<sup>2</sup>电梯智能运维重庆市高校工程中心, 重庆 402260;

<sup>3</sup>光纤传感与光电检测重庆市重点实验室, 重庆 400054

**摘要** 传统三维(3D)点云配准过程中存在配准误差高、计算量大及耗时长等问题,针对该问题,提出了一种 3D 点云中关键点的配准与优化算法。在关键点选取阶段,用边缘点检测算法剔除边缘关键点,以提高关键点特征描述的全面性和重复性,降低 3D 点云配准误差。在 3D 点云配准阶段,用 K-维树(KD-tree)加速的最近邻算法和迭代最近点算法剔除粗配准结果中的误配准关键点,降低配准误差,提高 3D 点云配准的速度与精度。实验结果表明,本算法在不同点云数据下,均能获得良好的配准结果。与传统 3D 点云配准算法相比,本算法的平均配准速率提高了 68.725%,平均配准精度提高了 49.65%。

**关键词** 图像处理; 关键点检测; 边缘检测; 三维重建; 点云处理; 误差优化

中图分类号 TP242

文献标志码 A

doi: 10.3788/LOP202158.0415008

## Registration and Optimization Algorithm of Key Points in Three-Dimensional Point Cloud

Song Tao<sup>1,2\*\*</sup>, Cao Libo<sup>1,3\*</sup>, Zhao Mingfu<sup>1,2</sup>, Liu Shuai<sup>1</sup>, Luo Yuhang<sup>1</sup>, Yang Xin<sup>1</sup>

<sup>1</sup>College of Electrical and Electronic Engineering, Chongqing University of Technology, Chongqing 400054, China;

<sup>2</sup>Elevator Intelligent Operation and Maintenance Chongqing Universities Engineering Center, Chongqing 402260, China;

<sup>3</sup>Optical Fiber Sensing and Photoelectric Detection Chongqing Key Laboratory, Chongqing 400054, China

**Abstract** In the traditional three-dimensional (3D) point cloud registration process, there are some problems such as high registration error, large amount of calculation and time-consuming. Aiming at these problems, a registration and optimization algorithm of key points in 3D point cloud is proposed in this paper. In the key point selection stage, the edge point detection algorithm is proposed to eliminate the edge points, improve the comprehensiveness and repeatability of the feature description of key points, and reduce the registration error of 3D point cloud. In the 3D point cloud registration stage, K-dimensional tree (KD-tree) accelerated nearest neighbor algorithm and iterative nearest point algorithm are used to eliminate key misregistration points in the coarse registration results, reduce the registration errors, and improve the speed and accuracy of 3D point cloud registration. Experimental results show that the algorithm can obtain good registration results under different cloud data. Compared with the traditional 3D point cloud registration algorithm, the average registration rate and the average registration accuracy of the algorithm are improved by 68.725% and 49.65%, respectively.

**Key words** image processing; key point detection; edge detection; three-dimensional reconstruction; point cloud processing; error optimization

**OCIS codes** 150.1135; 150.6910; 150.5758

收稿日期: 2020-07-04; 修回日期: 2020-08-07; 录用日期: 2020-08-13

基金项目: 国家自然科学基金青年科学基金(61701056)、重庆市教委基础研究项目(KJQN201901123)、重庆市科技局技术创新与应用发展重点项目(cstc2019jscx-mbxdX0002)、重庆理工大学研究生创新基金(ycx20192051,ycx20192052)

\* E-mail: easton.cao@foxmail.com; \*\* E-mail: tsong@cqut.edu.cn

## 1 引言

点云配准和重建是三维(3D)重建研究中的两大关键技术<sup>[1]</sup>,目前比较成熟的点云配准技术大多基于点云特征相似性匹配<sup>[2]</sup>。在3D点云完整且分辨率较高的情况下,传统3D点云配准算法<sup>[3]</sup>的配准效果较好。受外界条件及设备的限制,获得的点云质量往往参差不齐,导致传统3D点云配准算法存在配准误差大、配准过程耗时长等问题,严重影响了后续的配准和重建<sup>[4]</sup>效果。

3D点云配准算法在点云滤波<sup>[5]</sup>、点云描述、点云配准<sup>[6]</sup>等方面均发展出多种成熟的算法,包括直通滤波、体素滤波、条件滤波、半径滤波等滤波算法;点特征直方图(PFH)<sup>[7-8]</sup>、快速PFH(FPFH)<sup>[9]</sup>、旋转投影统计(ROPS)<sup>[10]</sup>等点云特征描述算法;最近邻(NN)<sup>[11]</sup>、采样一致性初始配准算法(SAC-IA)、迭代最近点(ICP)<sup>[12-13]</sup>、正态分布变换(DNT)<sup>[14-15]</sup>等点云配准及误配准剔除算法。当点云数据噪声少、质量高、数据完整时,上述算法均能得到较好的配准结果。但当点云数据存在大量噪声和边缘、缺陷且数据量大、杂乱等问题时,这些点云配准算法难以准确、快速达到所需要的效果。传统3D点云配准算法往往需要整片点云的参与,当点云数量较大时,算法的计算速度较慢;且工程应用中获取的3D点云通常存在大量的边缘<sup>[16]</sup>,导致部分点云的描述不全面,在后续配准过程中容易出现大量的误配准。传统3D点云配准算法在降低配准误差时需要全部点云参与迭代<sup>[17]</sup>,导致3D点云配准过程耗时长,且3D点云质量较低时,无法有效降低配准误差。

针对上述问题,本文提出了一种3D点云中关键点的配准与优化算法,可应用于点云关键点的选取和剔除、点云边缘计算、关键点配准及加速3D点云配准误差优化阶段。选取点云关键点时,提出计算点云边缘并剔除处于点云边缘的关键点,以提高关键点特征描述的全面性,进而提高关键点的配准质量,降低匹配误差;同时,3D点云配准时只需对关键点进行配准,无需整片点云参与计算,从而提高点云的配准速率。在3D点云配准误差优化阶段,最近邻算法<sup>[18]</sup>通过聚类,用欧氏距离作为标准,快速剔除误配准点对,提高配准点对的质量,实现配准误差优化的粗处理;将提取的高质量配准点对用迭代最近点算法<sup>[19]</sup>进行迭代优化,计算对应点之间的平均距离,通过迭代方法找出精度最高的变换矩阵,最终实现点云配准误差优化的细处理。同时,采用K-维树

(KD-tree)<sup>[20]</sup>对配准误差优化算法进行加速运算,以快速有效地降低3D点云中的配准误差。

## 2 算法原理

常用的3D点云配准算法的流程如图1所示,主要包含点云特征的提取、描述及配准等步骤。

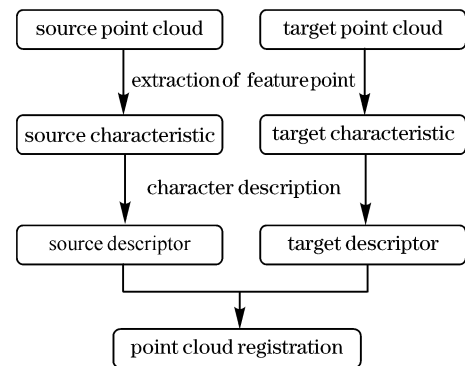


图1 3D点云配准算法的流程图

Fig. 1 Flow chart of the 3D point cloud registration algorithm

图2为本算法的流程图,在传统算法的特征提取中加入了关键点提取和边缘点剔除,以提高点云描述子的效率和全面性。在点云配准中用KD-tree加速结合最近邻和迭代最近点算法的配准误差优化算法,可以降低3D点云配准误差,加快3D点云的配准速率。

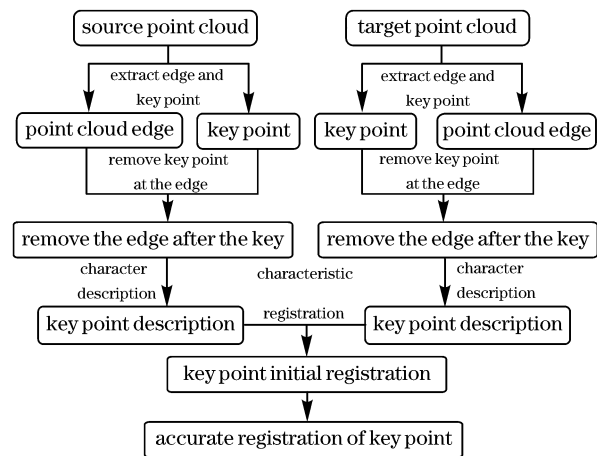


图2 本算法的配准流程图

Fig. 2 Registration flow chart of our algorithm

### 2.1 边缘计算及关键点提取算法

在3D点云重建中,设备获取的点云数据往往存在边缘数据不完整、孔洞等现象<sup>[21]</sup>,导致点云描述子无法全面描述局部特征,进而造成点云间的误配准。同时,点云特征需要计算全部点集,导致计算冗余、配准速度慢等问题。为了实现精确、快速的点

云配准,在提取关键点进行描述的基础上,检测点云边缘点,并剔除处于边缘和孔洞的关键点,进而提高关键点配准的精度和速度。

### 2.1.1 关键点提取算法

对点云关键点进行描述,不仅可以保证描述的全面性还能提高点云的运算速度。关键点检测算法<sup>[22-24]</sup>的具体流程如下。

1) 对点云  $P$  中的每个点  $P_i$  都建立一个局部坐标系,并将阈值  $r_{\text{frame}}$  作为搜索半径。

2) 搜索点云  $P$  中以  $P_i$  为中心,阈值  $r_{\text{frame}}$  为半径内的点  $P_j$ 。计算出中心点与区域内点的权值  $\omega_{ij}$ ,可表示为

$$\omega_{ij} = 1 / \|P_i - P_j\|, \quad \|P_i - P_j\| < r_{\text{frame}}, \quad (1)$$

式中,  $\| \cdot \|$  为欧氏距离。

3) 点云  $P$  中每个点  $P_i$  的协方差矩阵可表示为

$$\text{cov}(P_i) = \frac{\sum_{\|P_i - P_j\| < r_{\text{frame}}} \omega_{ij} (P_i - P_j)(P_i - P_j)^T}{\sum_{\|P_i - P_j\| < r_{\text{frame}}} \omega_{ij}}. \quad (2)$$

4) 计算每个点  $P_i$  协方差矩阵  $\text{cov}(P_i)$  的特征值  $\{\lambda_i^1, \lambda_i^2, \lambda_i^3\}$ ,并将特征值按从大到小的顺序排列。

5) 设置阈值  $\epsilon_1$  和  $\epsilon_2$ ,当对应点为关键点时,特征值需满足

$$\begin{cases} \lambda_i^2 / \lambda_i^1 \leq \epsilon_1 \\ \lambda_i^3 / \lambda_i^2 \leq \epsilon_2 \end{cases}. \quad (3)$$

6) 重复上述步骤,直到遍历整个点云  $P$ ,搜索出点云  $P$  中全部关键点并保存。

### 2.1.2 边缘关键点的剔除

3D 点云的边缘点往往处于最外围,边缘点周围的点大多处在边缘点的同一侧,判断点云  $P$  边缘点集的过程如下。

1) 对点云  $P$  中任意点  $P_i$  设定搜索半径  $r$ ,并将搜索半径  $r$  内的邻域点记为集合  $N(P_i)$ ,即  $N(P_i) = \{P_j | P_j \in P, \|P_i - P_j\| < r\}$ 。

2) 设定曲面方程  $ax + by + cz = d$  ( $d \geq 0$ ),  $a^2 + b^2 + c^2 = 1$ ,取  $P_i$  对应的集合  $N(P_i)$ ,计算  $N(P_i)$  到曲面的距离  $d_i = |ax + by + cz - d|$ 。求解  $\sum_j d_i^2$  最小时对应的特征向量  $\lambda$ ,该特征向量  $\lambda$  即为该点的法向量  $n$ 。

3) 根据  $P_i$  和其法向量  $n$  做该点的切平面  $\phi$ ,将集合  $N(P_i)$  内的点投影到切平面  $\phi$  上,记为  $N(P'_i)$ 。在  $N(P'_i)$  中选取一点  $P'_j$ ,以  $P_i P'_j / |P_i P'_j|$

为  $u$  轴, $n$  为  $\omega$  轴, $u \times \omega$  为  $v$  轴,以  $P_i$  为坐标中心构建局部坐标系,记为  $(P_i, u, v, \omega)$ 。

4) 分别计算集合  $N(P'_i)$  中其他点  $P'_n$  到点  $P_i$  的向量  $P_i P'_n$  与坐标轴  $u$  的顺时针夹角  $\Omega = (\theta_1, \theta_2, \theta_3, \dots, \theta_n)$ ,对相邻夹角两两作差得到夹角集合  $\Omega' = (\theta'_1, \theta'_2, \theta'_3, \dots, \theta'_{n-1})$ ,其中

$$\theta'_{n-1} = \theta_n - \theta_{n-1}. \quad (4)$$

可以发现,  $\Omega'$  为相邻两向量之间的夹角。如图 3 所示,将集合  $\Omega'$  中的元素降序排列,找出其中最大的夹角  $\theta'_{\text{max}}$ ,当  $\theta'_{\text{max}}$  大于阈值(实验取  $\pi/2$ ) 时,认为该点为边缘点。

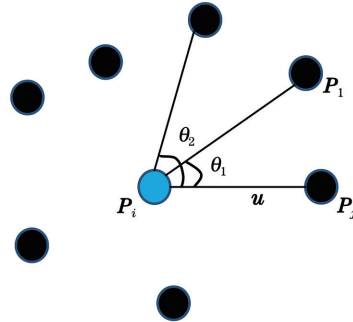


图 3 3D 点云边缘点的选取示意图

Fig. 3 Schematic diagram of the selecting edge points of the 3D point cloud

边缘关键点剔除算法的核心是边缘检测算法,具体流程:1)根据边缘检测算法对点云  $P$  进行边缘检测,当检测到的点为边缘点时,将该点放入集合  $Q_{\text{edge}}$  中。2)根据关键点检测算法对点云  $P$  进行关键点检测,将检测到的关键点放入集合  $P_{\text{keypoints}}$ ;3)分别计算关键点集合  $P_{\text{keypoints}}$  中的点到边缘点的距离  $d$ ,当某个关键点的最小距离  $d$  小于阈值  $\epsilon$  (实验取  $20 \text{ mr}$ ,  $\text{mr}$  为面片分辨率)时,认为该关键点为边缘关键点,剔除该点。

## 2.2 点云描述与配准

### 2.2.1 点云描述子

基于点特征的描述子主要根据点的坐标、法向量及曲率<sup>[25]</sup>等特征描述周围点之间的几何关系,存在特征信息之间的关系、全局特征信息少等缺点,因此实验采用 FPFH 对特征点进行描述。若  $P_i$  和  $P_j$  的法向量为  $n_i$  和  $n_j$ ,则源点  $P_s$  可表示为

$$P_s = \begin{cases} P_i, & n_i \cdot (P_j - P) \leq n_j \cdot (P_i - P) \\ P_j, & n_i \cdot (P_j - P) > n_j \cdot (P_i - P) \end{cases}. \quad (5)$$

根据源点  $P_s$  建立局部坐标系  $(P_s, u, v, \omega)$ ,如图 4 所示,以源点  $P_s$  的法向量  $n_s$  为  $u$  轴,结合非源点  $P_t$  为  $v = (P_t - P_s) \times u$  轴, $u \times v$  为  $\omega$  轴。

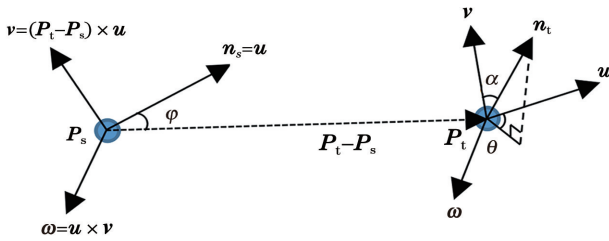


图 4 局部坐标系

Fig. 4 Local coordinate system

计算源点法向量  $n_s$  和非源点法向量  $n_t$  与  $(P_s, u, v, \omega)$  局部坐标系之间的角度  $(\alpha, \varphi, \theta)$ , 可表示为

$$\begin{cases} \alpha = v \cdot n_t \\ \varphi = u \cdot \frac{P_t - P_s}{d} \\ \theta = \arctan(\omega \cdot n_t, u \cdot n_t) \end{cases} \quad (6)$$

以某关键点  $P_q$  为源点坐标,  $k$  为邻域半径, 计算源点和邻域内非源点之间的角度  $(\alpha, \varphi, \theta)$ , 记为  $X_{\text{SPFH}}$ , 将每个关键点的特征描述作为配准依据, 可表示为

$$X_{\text{FPFH}}(P_q) = X_{\text{SPFH}}(P_q) + \frac{1}{k} \sum_{i=1}^k \frac{1}{\omega_k} X_{\text{SPFH}}(P_k), \quad (7)$$

式中,  $\omega_k$  为  $P_q$  和  $P_k$  的欧氏距离。描述子的影响范围如图 5 所示。

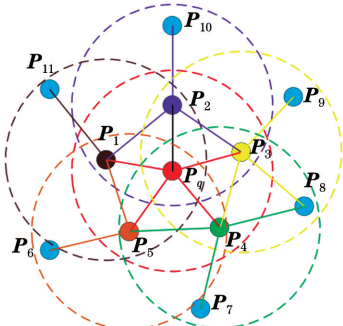


图 5 描述子的影响范围

Fig. 5 Scope of influence of the descriptor

### 2.2.2 点云配准

点云配准是点云处理中的关键步骤, 配准结果直接影响了点云的后续处理结果。实验用 SAC-IA<sup>[26]</sup> 进行点云配准, 该算法依赖点云特征的描述, 可实现点云对之间的粗配准, 原理如下。

1) 设定阈值  $s$ , 在源点云中采样  $x$  个点, 为了保证采样点之间具有不同的特征描述子, 各采样点之间的欧氏距离需大于阈值  $s$ 。

2) 在目标点云中查找同源点云中具有相似特

征描述子的采样点, 分别从源点云采样点和目标点云采样点中随机选取一个点作为对应点。通过该对应点, 计算点云对之间的变换矩阵。

3) 求解配准误差, 用 Huber 罚函数判定点云配

准的性能, 记为  $\sum_{y=1}^n H(l_y)$ , 可表示为

$$H(l_y) = \begin{cases} \frac{1}{2} l_y^2, & \|l_y\| < m_l \\ \frac{1}{2} m_l (2 \|l_y\| - m_l), & \|l_y\| > m_l \end{cases}, \quad (8)$$

式中,  $m_l$  为预设值,  $l_y$  为经变换后第  $y$  组对应点之间的距离差。重复上述步骤, 寻找最小配准误差的变换矩阵。

SAC-IA 可计算出点云的变换矩阵, 但其配准精度较差, 存在大量的误配准, 不能实现点云的精确配准, 只能用于点云的粗配准, 因此还需进行点云的误配准剔除及配准精度的优化。

### 2.3 配准优化算法

3D 点云粗配准后可以将两片点云基本重合, 但配准结果存在误差。为了进一步降低误差, 提高 3D 点云配准精度, 需对 3D 点云粗配准后的结果进行优化处理, 实现 3D 点云的精配准, 具体步骤: 1) 利用 KD-tree 对最近邻和迭代最近点算法进行加速, 以提高算法运算速率; 2) 通过最近邻算法用欧氏距离对点云进行聚类, 保留欧氏距离小于阈值的点, 实现误配准点的剔除; 3) 在获取的可靠配准点基础上, 用迭代最近点算法以配准点对之间的平均距离作为标准进行循环迭代, 寻找距离最小时的变换矩阵, 进一步提高配准精度。其中, 最近邻算法通过聚类, 用欧氏距离作为标准, 当距离小于设定阈值时, 认为配准点对为正确配准, 剩余点对均被认为是错误配准并被快速剔除, 从而提高配准点对的质量, 实现精度优化的粗处理。提取的高质量配准点对在欧氏距离上都低于设定阈值, 因此用迭代最近点算法进行迭代优化。

#### 2.3.1 KD-tree 构建

KD-tree 是一种对于多维欧氏空间分割构造的二叉树, 可将数据划分为多个维度。主要特点是不用对比全部的  $q$  维数据, 只选择某一个维度进行对比, 并根据该维度进行空间划分, 具有运算速度快的特点, 具体流程如下。

1) 寻找  $M$  维数据方差最大的维度  $L$ , 查询维度  $L$  上所有数据项的中位数  $m$ 。以  $m$  为基准, 将数据集一分为二, 分别记为  $D_L$  和  $D_H$ 。建立树节点,



存储数据的划分情况。

2) 对数据集  $D_L$  和  $D_H$  按照步骤 1) 再次划分, 将新的树节点作为上次划分的左右孩子。

3) 设置阈值  $\tau$ , 递归进行步骤 1) 和步骤 2), 直到数据项的数量小于设定阈值  $\tau$ , 停止划分。将最后划分的数据保存在节点中, 称为叶子节点。

### 2.3.2 最近邻剔除算法

通过构建的 KD-tree 实现最近邻算法, 对粗配准后的关键点进行近邻搜索。以欧氏距离为判断标准, 剔除欧氏距离大于阈值的配准关键点, 保存配准精度高的关键点, 具体步骤如下。

1) 寻找目标点的叶节点: 从根节点递归向下搜索 KD-tree, 如果目标点当前维度的坐标小于切分点坐标, 则转移到左子节点; 反之, 转移到右子节点。进入叶节点后, 将该叶节点内与待配准点之间距离最小的点作为当前最近点。

2) 递归向上回溯: 如果该节点保存的实例点比当前最近点距离目标点更近, 则更新当前最近点; 反之, 检查另一子节点区域。如果该区域与以目标点为球心, 以最近点间的距离为半径的球相交, 则另外一个子节点区域内可能存在距离目标点更近的点, 此时转移到另一子节点, 继续递归搜索。如果不相交, 则向上回溯。

3) 直到退回根节点, 搜索结束, 当前搜索的最近点为目标点的近邻点。

4) 设定阈值, 当目标点与最近点之间的欧氏距离小于阈值时, 认为配准精度高, 保留该关键点。反之, 则剔除该关键点。

根据上述原理, 可快速剔除配准精度低的关键点, 仅保留配准质量高的关键点, 实现关键点的优

化。通过剔除关键点中配准误差较大的点以及优化剩余关键点的质量, 实现配准点云精度优化的粗处理, 为后续细处理提供基础。

### 2.3.3 迭代最近点算法

粗处理后关键点的迭代处理是实现配准精度优化的关键步骤, 通过对剩余关键点进行迭代, 找出误差最小时的转换矩阵, 从而实现点云对之间的高精度配准, 具体步骤如下。

1) 在优化后的目标关键点云  $P$  中选取点集  $P_i \in P$ 。

2) 在优化后的源关键点云  $Q$  中使用 KD-tree 快速搜寻  $P_i$  的对应点  $Q_i \in Q$ , 使  $\|Q_i - P_i\|$  最小。

3) 通过对应关键点, 计算点云对的变换矩阵 (包含旋转矩阵  $R$  和平移矩阵  $t$ ), 满足误差函数值最小。用该变换矩阵计算目标关键点集  $P_i$  经变换后的对应点集  $P'_i = R * P_i + t, P_i \in P$ 。

4) 计算变换后点集  $P'_i$  与对应点集  $Q_i$  的平均距离  $d$

$$d = \frac{1}{z} \sum_{i=1}^z \|P'_i - Q_i\|^2. \quad (9)$$

5) 迭代计算, 寻找平均距离最小时的变换矩阵。设定阈值和迭代次数, 当误差小于阈值或迭代次数大于设定值时, 停止迭代。通过反复迭代计算, 求解出点云对平均距离最小时的变换矩阵, 实现点云对之间的配准优化处理, 使点云配准的精度更高。

## 3 实验结果及分析

### 3.1 实验数据与实验条件

用两种不同类型的数据集验证本算法的有效性, 图 6 为原始数据集中的 3D 点云图。数据集包

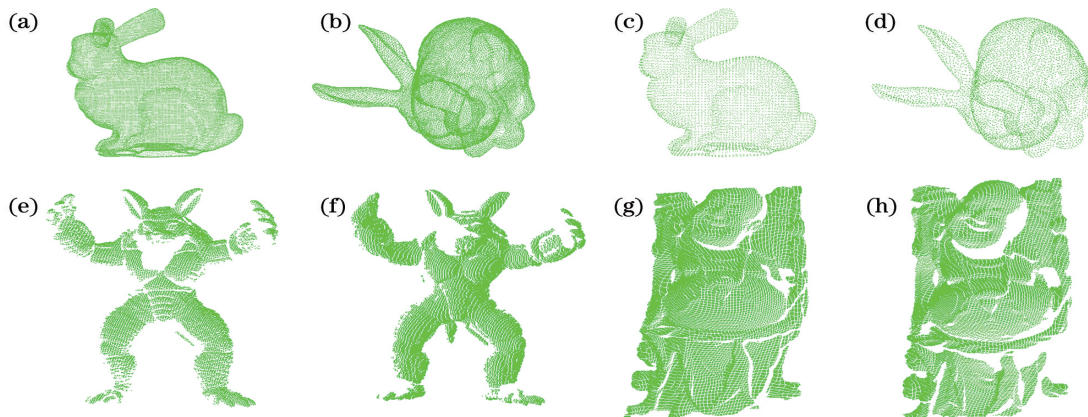


图 6 原始的 3D 点云图像。(a) bun\_zipper; (b) bun\_zipper\_tran; (c) bun\_zipper\_res2; (d) bun\_zipper\_res2\_tran; (e) ArmadilloSide\_60; (f) ArmadilloSide\_90; (g) happySideRight\_0; (h) happySideRight\_24

Fig. 6 Image of the original 3D point cloud. (a) bun\_zipper; (b) bun\_zipper\_tran; (c) bun\_zipper\_res2; (d) bun\_zipper\_res2\_tran; (e) ArmadilloSide\_60; (f) ArmadilloSide\_90; (g) happySideRight\_0; (h) happySideRight\_24

括:1)完整点云配准完整点云,以斯坦福大学标准数据集集中的 bun\_zipper 和 bun\_zipper\_res2 为原型,分别将数据集沿 X、Y 和 Z 轴旋转 45°,再沿 X 轴平移 2.5 m,沿 Y 轴平移 6.5 m(其中,单位 m 为坐标轴定义的数值),形成待配准的目标点云 bun\_zipper\_tran 和 bun\_zipper\_res2\_tran;2)部分点云配准部分点云,以斯坦福大学标准数据集集中的 Armadilloside\_60、Armadilloside\_90 和 happySideRight\_0、happySideRight\_24 作为配准数据。

实验的软硬件环境为:操作系统为 Ubuntu 16.04,MYNT CUBE 小觅魔方 NV1-TX2(基于 NVIDIA Jetson TX2 平台开发),外置 256 G 内存卡,开发环境为 Kdevelop 4.7,开发语言为 C++。

### 3.2 误差得分评价标准

采用误差得分和配准耗时对实验结果进行评

价,误差得分可表示为配准后源点云与目标点云之间最近点的距离平方和,可表示为

$$X_{\text{Score}} = \sum_i^n (p_i - q_i)^2, \quad (10)$$

式中,  $p_i$  和  $q_i$  分别为源点云和目标点云之间对应的最近点。用(10)式循环计算源点云与目标点云之间全部最近点的距离平方和,获得的数值作为误差得分。可以发现,误差得分越小,3D 点云配准的效果越好,精度越高。

### 3.3 关键点选取与边缘点剔除实验

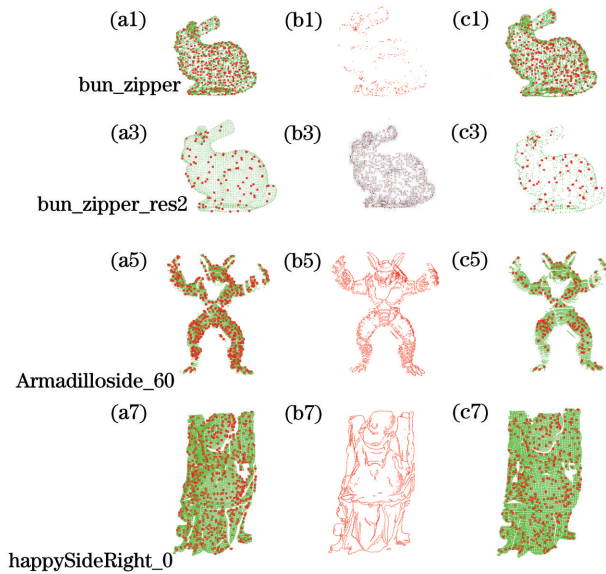
用关键点检测算法和边缘点剔除算法对两种数据集进行实验分析,结果如表 1 所示,包括 3D 点云的原始点数、检测的关键点数、检测的边缘点数及剔除的边缘关键点数。

表 1 关键点的选取与边缘点的剔除结果

Table 1 Results of the key points selection and edge point elimination

Point cloud	Original point	Detection key point	Detection edge point	Remove edge key point
bun_zipper	35947	479	3371	72
bun_zipper_tran	35947	479	14848	226
bun_zipper_res2	8171	107	1637	30
bun_zipper_res2_tran	8171	107	2853	45
Armadilloside_60	204800	566	8917	352
Armadilloside_90	204800	652	9706	371
happySideRight_0	164604	718	9629	266
happySideRight_24	154700	632	9348	258

分别采用边缘检测算法、关键点提取算法以及剔除边缘关键点算法对两种数据集进行处理,结果



如图 7 所示。可以发现,对于完整点云和部分点云,关键点检测算法都能很好地根据其周围点的特征检

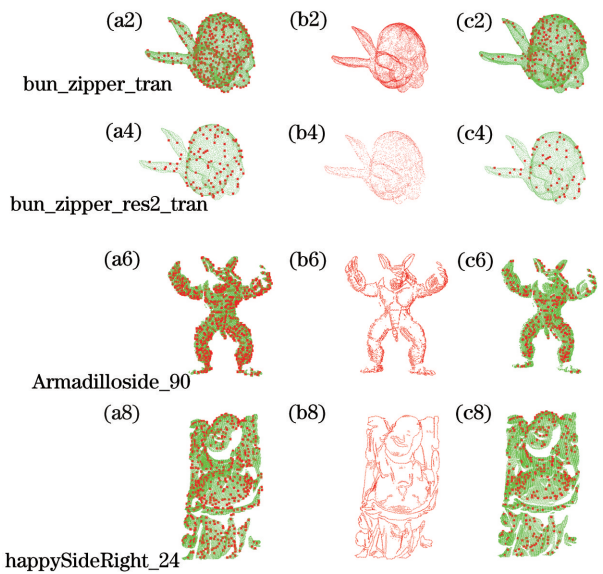


图 7 边缘检测与剔除边缘关键点的 3D 点云。(a) 关键点的提取结果; (b) 边缘检测结果; (c) 剔除边缘关键点的结果  
Fig. 7 3D point cloud with edge detection and removal of key points on the edge. (a) Result of the extraction of key points; (b) result of the edge detection; (c) result of the removing key points of the edge

测出关键点;而部分点云具有不完整性,存在锯齿状边缘,边缘检测算法也可以有效测出处于边缘的点云。如图 7(b5)~图 7(b8)中,边缘检测算法可以有效地区分边缘点云与非边缘点云。

从表 1 可以发现,关键点检测算法能检测出上百个关键点,在普通点云中,能在保证点云关键点数量的前提下保证点云配准速率,避免重复点云计算导致的运算速度慢、资源占用多问题。针对不同类型的 3D 点云,可得到不同的边缘检测量,这表明边缘检测算法对于不同点云真实有效。结合图 7 可以看出,边缘检测算法能有效检测出点云边缘。边缘点云周围具有明显的不规则形状(如锯齿状),易产生关键点,但这些关键点在描述时往往具有不全面性。从边缘关键点的剔除数量可以看出,处于边缘的关键点数据都会被边缘关键点剔除算法检测并剔除掉。

### 3.4 三维点云配准实验

为验证本算法的有效性,分别用完整点云和部分点云数据集,用传统 3D 点云配准算法与本算法进行对比实验。

#### 3.4.1 完整点云配准完整点云

针对完整点云配准完整点云,采用数据集 1)进行实验,结果如图 8 所示。可以发现,相比传统 3D 点云配准算法,本算法的配准效果更好。如在斯坦福兔子胸部和前爪部分的配准结果中,本算法的表

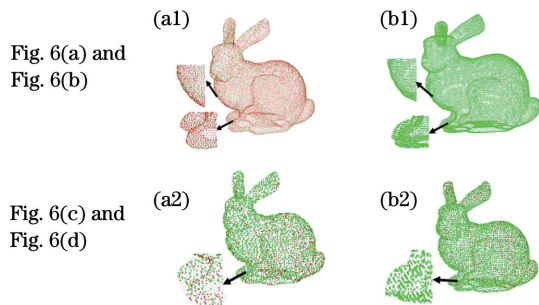


图 8 完整点云配准优化的结果。(a)本算法;(b)传统算法  
Fig. 8 Result of the complete point cloud registration optimization. (a) Our algorithm; (b) traditional algorithm

现更好。传统算法在配准前爪时容易出现重叠,但本算法配准的前爪具有明显的层次,能区分出两只前爪,且没有产生重叠。

#### 3.4.2 部分点云配准部分点云

针对部分点云配准部分点云,采用数据集 2)进行实验,获得的结果如图 9 所示。可以发现,在点云不完整时,相比传统配准算法,本算法的配准效果更好。如在 Armadillo 耳部、手部以及脚部的配准结果均优于传统算法,且本算法的配准结果无重影,而传统算法配准结果存在明显的重影和错位。

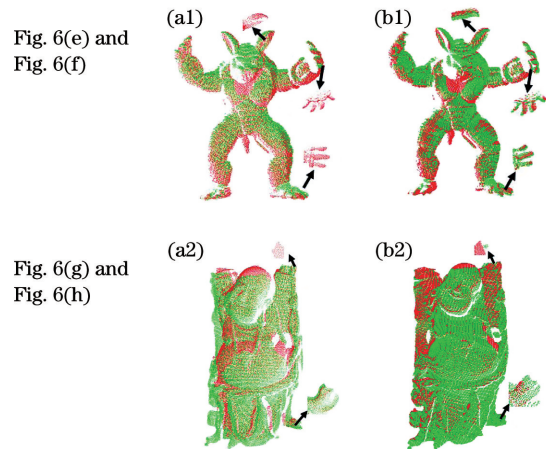


图 9 部分点云配准优化的结果。(a)本算法;(b)传统算法  
Fig. 9 Result of partial point cloud registration optimization. (a) Our algorithm; (b) traditional algorithm

### 3.5 配准误差与耗时实验

为了验证本算法的有效性,将传统 3D 点云配准算法与本算法的实验结果进行对比分析。根据评价标准,计算两种算法配准后的误差得分及耗费时间,结果如表 2 所示。可以发现,相比传统算法,本算法进行配准时的运算速度平均提高了 68.725%,配准精度平均提高了 49.65%。由于传统算法在进行配准优化时将整个点云带入迭代计算,本算法只选取优良的关键点(可剔除处于边缘的边缘关键点、提高关键点的描述全面性)进行配准优化,同时结合最近邻算法与迭代最近点算法对误差进行优化,用

表 2 两种算法配准后的误差得分及时间

Table 2 Error scores and time after registration of the two algorithms

Algorithm	Traditional algorithm		Our algorithm	
	Error score	Time /s	Error score	Time /s
Fig. 6(a) and Fig. 6(b)	$2.40371 \times 10^{-12}$	2.46881	$1.47501 \times 10^{-12}$	1.63467
Fig. 6(c) and Fig. 6(d)	$2.41583 \times 10^{-12}$	0.48333	$1.29984 \times 10^{-13}$	0.03180
Fig. 6(e) and Fig. 6(f)	$6.52757 \times 10^{-6}$	1.10415	$4.14221 \times 10^{-6}$	0.29744
Fig. 6(g) and Fig. 6(h)	$1.52925 \times 10^{-6}$	2.70751	$1.08890 \times 10^{-6}$	0.68695



KD-tree 加速运算,提高了 3D 点云配准的运算速度和配准精度。对于点云分辨率较低的完整点云,相比传统算法,本算法的配准效果有明显提升。原因是传统算法需要大量点云进行迭代优化以降低误差,在点云分辨率较低时,点云量较少,配准误差较大。而本算法可以克服这类问题,在点云分辨率较低时,相比传统算法,本算法在配准精度和配准优化速率上均提高超过 90%。

## 4 结 论

提出了一种 3D 点云中关键点配准和误差优化算法,在关键点检测阶段,用边缘关键点剔除算法提取处于非边缘的关键点组成性能优异的待配准点集,避免了处于边缘的关键点描述不全面的问题,从而降低关键点的配准误差。在关键点配准后,执行配准误差优化算法,结合最近邻算法和迭代最近点算法优化粗配准误差,同时采用 KD-tree 对误差优化算法进行加速。实验结果表明,在不同情况的数据集下,相比传统算法,本算法的配准效果更好,在配准速率上平均提高了 68.725%,在配准精度上平均提高了 49.65%;且在完整点云分辨率较低的情况下,本算法的提升效果更加明显。

## 参 考 文 献

- [1] Zheng T X, Huang S, Li Y F, et al. Key techniques for vision based 3D reconstruction: a review[J]. *Acta Automatica Sinica*, 2020, 46(4): 631-652.  
郑太雄, 黄帅, 李永福, 等. 基于视觉的三维重建关键技术研究综述[J]. *自动化学报*, 2020, 46(4): 631-652.
- [2] Xu J Z, Wang J R. Auto-registration method of ground based building point clouds based on line features and iterative closest point algorithm [J]. *Journal of Computer Applications*, 2020, 40(6): 1837-1841.  
徐景中, 王佳荣. 基于线特征及迭代最近点算法的地基建筑物点云自动配准方法[J]. *计算机应用*, 2020, 40(6): 1837-1841.
- [3] Ge B Z, Peng B, Tian Q G. Registration of three-dimensional point-cloud data based on curvature map [J]. *Journal of Tianjin University*, 2013, 46(2): 174-180.  
葛宝臻, 彭博, 田庆国. 基于曲率图的三维点云数据配准[J]. *天津大学学报*, 2013, 46(2): 174-180.
- [4] 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.
- [5] Dai S J, Ren Y C, Zhang H B. Study on smooth denoising of 3D scattered point clouds with anisotropic diffusion filtering[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2018, 30(10): 1843-1849.  
戴士杰, 任永潮, 张慧博. 各向异性扩散滤波的三维散乱点云平滑去噪算法[J]. *计算机辅助设计与图形学学报*, 2018, 30(10): 1843-1849.
- [6] Wang B, Liu L, Hou Y Q, et al. Three-dimensional cardiac point cloud registration by improved iterative closest point method [J]. *Optics and Precision Engineering*, 2020, 28(2): 474-484.  
王宾, 刘林, 侯榆青, 等. 应用改进迭代最近点方法的三维心脏点云配准[J]. *光学精密工程*, 2020, 28(2): 474-484.
- [7] Rusu R B, Marton Z C, Blodow N, et al. Learning informative point classes for the acquisition of object model maps[C]//2008 10th International Conference on Control, Automation, Robotics and Vision, December 17-20, 2008, Hanoi, Vietnam. New York: IEEE Press, 2008: 643-650.
- [8] Rusu R B, Blodow N, Marton Z C, et al. Aligning point cloud views using persistent feature histograms [C]//2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, September 22-26, 2008, Nice, France. New York: IEEE Press, 2008: 3384-3391.
- [9] Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration [C]//2009 IEEE International Conference on Robotics and Automation, May 12-17, 2009, Kobe, Japan. New York: IEEE Press, 2009: 3212-3217.
- [10] 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.
- [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] Shen B W, Yin F, Chou W S. A 3D modeling method of indoor objects using kinect sensor [C]//2017 10th International Symposium on Computational Intelligence and Design (ISCID), December 9-10, 2017, Hangzhou, China. New York: IEEE Press, 2017: 64-68.
- [13] Besl P J, McKay N D. A method for registration of 3-D shapes [J]. *IEEE Transactions on Pattern*



- Analysis and Machine Intelligence, 1992, 14(2): 239-256.
- [14] Chen Y, Medioni G. Object modelling by registration of multiple range images [J]. Image and Vision Computing, 1992, 10(3): 145-155.
- [15] Chen M L, Lu W X, 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.
- [16] 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.
- [17] Liu J, Bai D. 3D point cloud registration algorithm based on feature matching [J]. Acta Optica Sinica, 2018, 38(12): 1215005.  
刘剑, 白迪. 基于特征匹配的三维点云配准算法[J]. 光学学报, 2018, 38(12): 1215005.
- [18] Liu R T, Liang J C. Reverse nearest neighbor query based on new index structure [J]. Journal of Computer Research and Development, 2020, 57(6): 1335-1346.  
刘润涛, 梁建创. 基于新型索引结构的反最近邻查询[J]. 计算机研究与发展, 2020, 57(6): 1335-1346.
- [19] Yang J D, Sun L M, Shao Y J, et al. Efficient iterative closest point localisation algorithm for autonomous robots [J]. Journal of Mechanical Engineering, 2018, 54(15): 15-20, 30.  
杨晶东, 孙磊明, 邵雨婕, 等. 一种有效的自主机器人迭代最近点定位算法[J]. 机械工程学报, 2018, 54(15): 15-20, 30.
- [20] Li J F, Tan Y H, Liao S H. Highly parallel SAH-KD-tree construction for raytracing [J]. Journal of Hunan University (Natural Sciences), 2018, 45(10): 148-154.  
李建锋, 谭耀华, 廖胜辉. 用于光线跟踪的高并行度表面积启发式(SAH)KD树构建[J]. 湖南大学学报(自然科学版), 2018, 45(10): 148-154.
- [21] Xiong F G, Huo W, Han X, et al. Removal method of mismatching keypoints in 3D point cloud [J]. Acta Optica Sinica, 2018, 38(2): 0210003.  
熊风光, 霍旺, 韩燮, 等. 三维点云中关键点误匹配剔除方法[J]. 光学学报, 2018, 38(2): 0210003.
- [22] Yamakawa S, Shimada K. Polygon crawling: feature-edge extraction from a general polygonal surface for mesh generation [M]//Hanks B.W, et al. Proceedings of the 14th International Meshing Roundtable. Berlin, Heidelberg: Springer-Verlag, 2005: 257-274.
- [23] Daniels J I, Ha L K, Ochotta T, et al. Robust smooth feature extraction from point clouds [C]//IEEE International Conference on Shape Modeling and Applications 2007 (SMI '07), June 13-15, 2007, Lyon, France. New York: IEEE Press, 2007: 123-136.
- [24] Park M K, Lee S J, Lee K H. Multi-scale tensor voting for feature extraction from unstructured point clouds [J]. Graphical Models, 2012, 74(4): 197-208.
- [25] Ho H T, Gibbins D. Curvature-based approach for multi-scale feature extraction from 3D meshes and unstructured point clouds [J]. IET Computer Vision, 2009, 3(4): 201-212.
- [26] Lu J, Shao H X, Wang W, et al. Point cloud registration method based on key point extraction with small overlap [J]. Transactions of Beijing Institute of Technology, 2020, 40(4): 409-415.  
陆军, 邵红旭, 王伟, 等. 基于关键点特征匹配的点云配准方法[J]. 北京理工大学学报, 2020, 40(4): 409-415.