

# 一种改进的二维 ICP 点云配准算法

谢小鹏<sup>1</sup>, 古家威<sup>2</sup>

(1. 东莞理工学院城市学院, 广东 东莞 523419; 2. 季华实验室, 广东 佛山 528000)

**摘要:**为解决传统 ICP 算法存在的运行不稳定、处理速度慢等问题,本文分析了传统 ICP 算法的缺点并对其进行改进。首先,针对传统 ICP 算法容易出现的迭代方向错误问题,本文打乱了目标点集的点序号,然后在目标点寻找最近点时选择一对一的方式进行。算法的处理速度上,本文通过设置动态阈值,剔除了误匹配点对,减少了 ICP 算法的迭代次数,进而提高了算法的处理速度。

**关键词:**二维 ICP; 乱序; 一对一; 动态阈值

**中图分类号:**TP391 **文献标识码:**A **DOI:**10.3969/j.issn.1001-5078.2021.07.020

## An improved two-dimensional ICP point cloud registration algorithm

XIE Xiao-peng<sup>1</sup>, GU Jia-wei<sup>2</sup>

(1. City College of Dongguan University of Technology, Dongguan 523419, China;  
2. Jihau Laboratory, Foshan 528000, China)

**Abstract:** In order to solve the problems of unstable operation and slow processing speed of the traditional ICP algorithm, the shortcomings of the traditional ICP algorithm are analyzed and improved. First of all, in order to solve the problem of iterative direction error that the traditional ICP algorithm is prone to, the point number of the target point set is disrupted, and then a one-to-one approach is selected when finding the closest point at the target point. In terms of the processing speed of the algorithm, the mismatched point pairs are eliminated by setting a dynamic threshold, the number of iterations of the ICP algorithm are reduced, and the processing speed of the algorithm is improved.

**Keywords:** two-dimensional ICP; serial number randomization; one to one; dynamic threshold

### 1 引言

点云配准技术在逆向工程、机器人导航及无人驾驶等领域具有广泛的应用,点云配准一般来说包括两个步骤,粗配准和精配准,粗配准主要实现目标点集与参考点集之间的快速配准,但是精度不高,常用的方法有 NDT (Normal Distribution Transform) 方法<sup>[1]</sup>,王庆闪等实现了 NDT 与 ICP 结合的点云配准方法<sup>[2]</sup>,精配准的实现方法主要是迭代最近点 (ICP, Iterative Closest Point) 算法<sup>[3]</sup>,由 Besl 于 1992 年提出。近些年来,很多学者对 ICP 算法进行了改

进,Censi 采用点到线的度量方式进而提出了 PL-ICP (Point-to-Line ICP) 方法<sup>[4]</sup>,该方法提高了 ICP 算法的精度;解则晓改进了点到面 ICP 即 P-ICP (Point-to-Plane ICP)<sup>[5]</sup>;Chetverikov 提出了 Trimmed ICP<sup>[6]</sup>,该方法将  $n$  个匹配点对的欧氏距离进行排序,去除距离最大的  $\eta n$  ( $0 < \eta < 1$ ) 个匹配点对,该方法需要对  $\eta$  值做出平衡,过大过小都会影响最终结果的精度;孙翌将<sup>[7]</sup>  $k$  邻域搜索方法应用到 ICP 算法中,提高了查找最近点的效率,进而加快了 ICP 算法的处理速度。

**基金项目:**东莞理工学院城市学院重大科技培育项目 (No. 2018YZD001Z);广东省科技厅 (No. 2019A1515110352) 资助。

**作者简介:**谢小鹏 (1961 -),男,博士,教授,博导,从事机械电子方面的研究。E-mail:66976575@qq.com

**收稿日期:**2020-09-08; **修订日期:**2020-09-22

本文针对传统 ICP 算法存在的问题进行了改进,首先介绍了传统 ICP 的主要步骤,发现传统 ICP 算法有以下几种问题:迭代方向错误,迭代优化过程中易出现局部最优的情况,迭代次数过多,增加了算法的处理时间。本文提出了改进方案,首先打乱目标点集中点序号的顺序,然后在接下来的寻找最近点过程中采用一对一的方式进行,增强了算法的稳定性,对每个匹配点进行筛选,使用动态阈值去过滤那些欧氏距离过大的一些匹配点对,加快了算法收敛速度。

## 2 传统二维 ICP 算法

传统的二维 ICP 算法的一般步骤如下:

步骤一:选取参考点集和目标点集。

步骤二:遍历目标点集中所有的点,在参考点集中选择欧氏距离最小的一个点。

步骤三:建立目标函数,对目标函数进行优化求解,得到目标点集的旋转矩阵  $R_j$  和平移矩阵  $T_j$ ,进而得到新的目标点集。

步骤四:判断是否达到了最大迭代次数或者误差条件,是则停止迭代,输出最终的结果,否则转到步骤一,继续迭代。

在步骤三中,需要建立目标函数并优化求解,下面是目标函数的建立和优化过程:

设目标点集中的任意一点为  $M_i(x_i, y_i)$ ,在参考点集中找到了距离自身最小的点为  $N_i(x'_i, y'_i)$ ,对点  $M_i(x_i, y_i)$  作平面旋转平移变换得到点  $M'_i(x'_i, y'_i)$ ,变换关系如下:

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = R_j \begin{pmatrix} x_i \\ y_i \end{pmatrix} + T_j \quad (1)$$

其中,  $R_j = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$  为旋转矩阵,  $T_j =$

$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$  为平移矩阵。

将目标点集与参考点集中所有对应点对的欧氏距离的平方和累加起来得到目标函数,并使之达到最小,于是目标函数为:

$$\begin{aligned} f(\theta, t_1, t_2) &= \sum_{i=1}^n [(x'_i - x'_i)^2 + (y'_i - y'_i)^2] \\ &= \sum_{i=1}^n [(x'_i - \cos\theta x_i - \sin\theta y_i - t_1)^2 + \end{aligned}$$

$$(y'_i + \sin\theta x_i - \cos\theta y_i - t_2)^2] \quad (2)$$

接下来对目标函数进行求解:

对目标函数中的 3 个待定系数分别求偏导并令其为 0:

$$\begin{aligned} \frac{\partial f}{\partial \theta} &= \sum (y'_i + \sin\theta x_i - \cos\theta y_i - t_2) \cdot \\ &(\cos\theta x_i + \sin\theta y_i) + \sum (x'_i - \cos\theta x_i \\ &- \sin\theta y_i - t_1)(\sin\theta x_i - \cos\theta y_i) = 0 \quad (3) \end{aligned}$$

$$\frac{\partial f}{\partial t_1} = -2 \sum (x'_i - \cos\theta x_i - \sin\theta y_i - t_1) = 0 \quad (4)$$

$$\frac{\partial f}{\partial t_2} = -2 \sum (y'_i - \cos\theta y_i + \sin\theta x_i - t_2) = 0 \quad (5)$$

对上述 3 个方程联立即可得到 3 个待定系数的值,进而得到本次迭代的旋转矩阵  $R_j$  和平移矩阵  $T_j$ ,将目标点集中所有的点代入到式(1)中作旋转平移变换即可得到新的目标点集。

当算法结束后,最后一次迭代得到的目标点集即为最终的目标点集,最终的旋转矩阵和平移矩阵由公式(6)得到:

$$\begin{pmatrix} R & T \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} R_m & T_m \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R_{m-1} & T_{m-1} \\ 0 & 0 & 1 \end{pmatrix} \dots \begin{pmatrix} R_j & T_j \\ 0 & 0 & 1 \end{pmatrix} \dots \begin{pmatrix} R_1 & T_1 \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

其中,  $m$  是迭代次数。

## 3 改进的二维 ICP 算法

传统的二维 ICP 算法容易出现以下一些问题:

(1)在寻找最近点的过程中,会出现目标点集与参考点集多对一的情况,进而使得算法的迭代方向发生错误,最终的误差无法收敛。在图 1 中,箭头

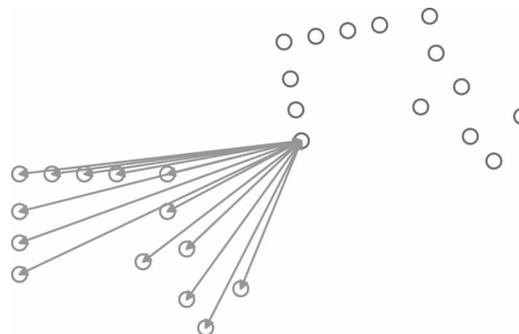


图 1 当目标点集与参考点集以多对一方式匹配时  
Fig. 1 The target point set matches the reference point set in a many-to-one manner

所指圆圈指的是目标点集,其它圆圈指的是参考点集,绿色带箭头线段指的是目标点集与最近点的配对关系,可以发现目标点集中所有的点都指向了同一个最近点。

(2) 本文将目标点集中的某个点与其找到的最近点称作匹配点对,寻找到最近点后,有些匹配点对的欧式距离过大,会影响算法的迭代方向,增加算法的迭代次数。

在图 2 中黑色连线的匹配点对间的欧式距离远大于其他匹配点对的欧式距离,这将大大地改变 ICP 算法的迭代方向。

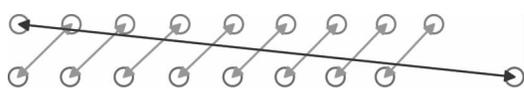


图 2 当出现匹配点对间欧式距离过大的情况

Fig. 2 The Euclidean distance between matching points is too large

(3) 传统的二维 ICP 算法在优化过程中容易陷入局部最优的情况,导致求解出来的结果错误。

本文提出的一种改进的二维 ICP 扫描匹配算法针对传统的二维 ICP 算法进行了改进。

(1) 为避免出现目标点集与参考点集多对一的情况,采取目标点集与参考点集一对一的方式,具体实施方法则是当目标点集中的某个点在参考点集中找到最近点后,将参考点集中的这个最近点排除在参考点集之外。

(2) 打乱目标点集的顺序。一般来说,用于 ICP 匹配的点集的点序号存在着某种规律性,比如从上到下或者从左到右排列,目标点集中序号最前的点能够在参考点集中正确找到最近点,由于序号靠前的点具有优先选择权,目标点集中序号靠后的点找到的最近点有很大的可能性不是真实的最近点。由于点集客观存在的排列规律性,会导致大多数点都找不到真实的最近点,为了让目标点集中大多数点都能找到真实的最近点,本文将目标点集中的点序号打乱,增加了目标点集中能找到真实最近点的个数,如图 3 所示。

在图 3(a) 中目标点集的点序号按照从上往下的顺序排列,由此形成了图中的匹配点对,可以发现这些匹配点对间的欧式距离都相对要更大,对于这些欧式距离过大的匹配点对,算法会对其进

行筛选,进而发生大范围误过滤的情况,这将影响算法的最终效果。在图 3(b) 中目标点集的点序号经过了打乱,此时目标点集中有更多的点找到了真实最近点。

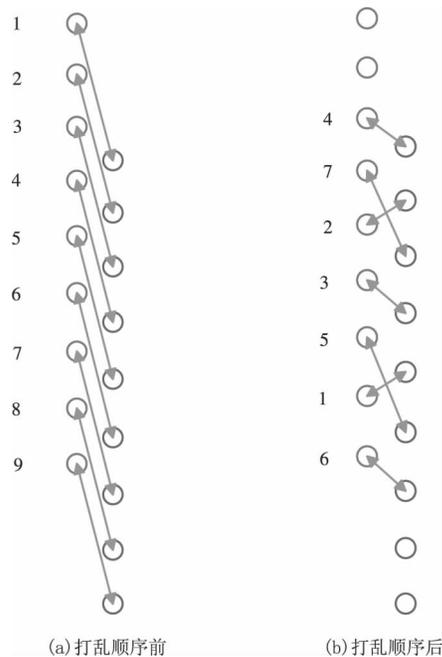


图 3 目标点集打乱顺序前后对比

Fig. 3 Before and after the target point set is shuffled

(3) 某些匹配点对间的欧式距离  $d_{\min}$  与其他正常的匹配点对欧式距离相差过大,会影响 ICP 算法的优化方向,于是本文将这些欧式距离过大的匹配点对剔除,此时本文选择一个阈值  $d_T$  作为筛选匹配点对的阈值,当匹配点对的欧式距离大于该阈值时将被剔除,然而 ICP 算法是目标点集与参考点集不断靠近的一个过程,每一次迭代都会使得这两个点集之间的距离减小,因此阈值  $d_T$  也应该随着迭代次数的增加而减小的,即在 ICP 每次迭代过程中,阈值  $d_T$  都是不同的。为了能过滤掉更多的错误匹配点对,同时保证不发生误过滤的情况,本文将阈值  $d_T$  与上一次迭代时得到所有匹配点对间的平均欧式距离  $\bar{d}_j$  相关联,如式,其中  $k$  是一个大于 1 的系数,  $j$  指的是当前迭代次数,ICP 算法每次迭代过后,  $\bar{d}_j$  都会更新一次。按照本文提出的改进方案,本文提出了如图 4 所示的算法流程图。

$$d_T = k \bar{d}_j \quad (7)$$

$$\bar{d}_j = \sum_{i=1}^n d_i / n \quad (8)$$

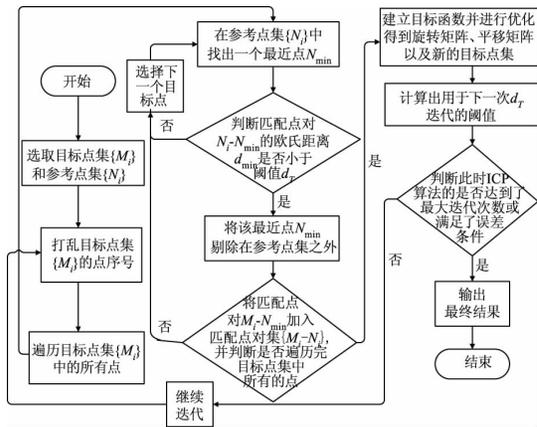


图 4 改进的二维 ICP 算法流程图

Fig. 4 Improved two-dimensional ICP algorithm flowchart

4 实验

为验证本文改进算法的优点,进行对比实验。实验中点集数据来自于一款旋转二维激光扫描器在实际环境中所采集的数据,该款激光扫描器每转一圈获取 360 个数据。实验环境的测试环境如下:CPU 为 Intel (R) Core (TM) i5 - 6500U, 8G 内存, Windows10 系统,编程环境是 MATLAB7.0。

图 5 是传统二维 ICP 算法的 ICP 迭代过程,图中的点集数据来源于二维激光扫描器,图中的点集数据来源于二维激光扫描器,图中有两簇点集,分别是目标点集及参考点集,其中参考点集由目标点集经过旋转平移变换所得到,连线表示表示匹配点对间的匹配关系。迭代过程中,目标点集与参考点集不断靠近,最终几乎完全重合。图 6 是本文改进的二维 ICP 算法的 ICP 迭代过程,采用的数据与图 5 中的数据相同,都是从初始状态到最终目标点集与参考点集的无限接近,可以发现本文算法所需的迭代次数明显要更少,特别是在无限接近阶段,本文算法能够迅速地收敛。

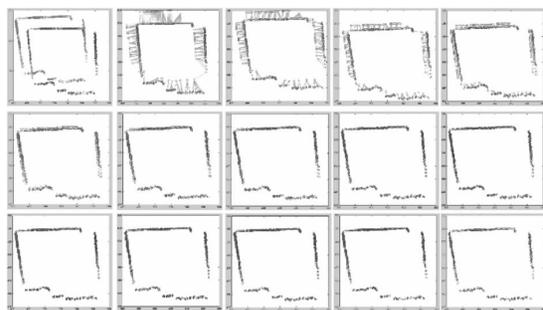


图 5 传统二维 ICP 算法的迭代过程

Fig. 5 Iterative process of traditional two-dimensional ICP algorithm

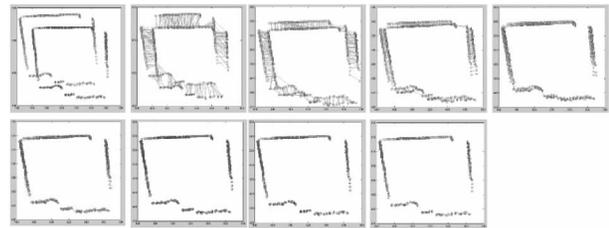


图 6 本文改进的二维 ICP 算法的迭代过程

Fig. 6 Iteration process of the improved two-dimensional ICP algorithm

为具体量化本文改进 ICP 算法的优势,在获得多组点集数据之后,使用这些数据在传统 ICP 算法与本文改进 ICP 算法上分别处理,表 1 是传统 ICP 算法与本文改进 ICP 算法在平均迭代次数与平均处理时间上的对比,可以发现本文所需的迭代次数要更少,处理时间也要更短。本文改进 ICP 算法处理较快的原因在于算法的后续收敛阶段,正常的匹配点对的欧氏距离趋于 0,因此一些误匹配点对的出现会极大地影响算法的收敛,而本文算法设有自适应变化的阈值用来过滤一些误匹配点对,从而加快了算法收敛,减少了迭代次数。

表 1 传统 ICP 算法与本文改进 ICP 算法的迭代次数与处理时间

Tab. 1 The iteration number and processing time of the traditional ICP algorithm and the improved ICP algorithm

|             | 平均迭代次数/次 | 平均处理时间/s |
|-------------|----------|----------|
| 传统 ICP 算法   | 14.571   | 0.315    |
| 本文改进 ICP 算法 | 8.857    | 0.245    |

5 结论

本文针对传统 ICP 算法出现的一些问题,对其进行了改进。本文采用目标点集与参考点集一对一的方式进行最近点配对,并对目标点集的点序号进行打乱,增强了 ICP 算法的稳定性;在 ICP 算法每次迭代过后,设置一个新的阈值用于过滤一些误匹配点对,进而减小算法的迭代次数,加快了算法处理速度,实验结果表明,本文改进的 ICP 算法在处理速度具有明显地提升。

参考文献:

[1] Xie Zexiao, Xu Shang. A survey on the ICP algorithm and

- its variants in registration of 3D point clouds[J]. Periodical of Ocean University of China, 2010, 40(1): 99 - 103. (in Chinese)
- 解则晓, 徐尚. 三维点云数据拼接中 ICP 及其改进算法综述[J]. 中国海洋大学学报, 2010, 40(1): 99 - 103.
- [2] Wang Qingshan, Zhang Jun, Liu Yuansheng, et al. Point cloud registration algorithm based on combination of NDT and ICP [J]. Computer Engineering and Applications, 2020, 56(7): 88 - 95. (in Chinese)
- 王庆闪, 张军, 刘元盛, 等. 基于 NDT 与 ICP 结合的点云配准算法[J]. 计算机工程与应用, 2020, 56(7): 88 - 95.
- [3] Besl P J, McKay N D. Method for registration of 3-D shapes[C]//Sensor fusion IV: control paradigms and data structures. International Society for Optics and Photonics, 1992, 1611: 586 - 606.
- [4] Censi A. An ICP variant using a point-to-line metric [C]//2008 IEEE International Conference on Robotics and Automation. Ieee, 2008: 19 - 25.
- [5] Xie Zexiao, Xu Shang, Li Xuyong. A fine registration method for 3D point clouds in reverse engineering [J]. China Mechanical Engineering, 2009, 20(13): 1577 - 1581. (in Chinese)
- 解则晓, 徐尚, 李绪勇. 逆向工程中三维点云数据精确拼接方法 [J]. 中国机械工程, 2009, 20(13): 1577 - 1581.
- [6] Chetverikov D, Svirko D, Stepanov D, et al. The trimmed iterative closest point algorithm[C]//Object Recognition Supported by User Interaction for Service Robots. IEEE, 2002, 3: 545 - 548.
- [7] Sun Yi, Bao Jiandong. Application of improved ICP algorithm based on fast algorithm for k-nearest neighbor search in pipeline inspection [J]. Journal of Test and Measurement Technology, 2019, 33(2): 99 - 103. (in Chinese)
- 孙翌, 包建东. 基于 k 领域快速搜索的改进 ICP 算法在管道检测中的应用 [J]. 测试技术学报, 2019, 33(2): 99 - 103.