

单线激光点云拼接SLAM算法研究

张杰^{1,2*}, 裴东^{1,2}, 高文辉^{1,2}

¹西北师范大学物理与电子工程学院, 甘肃 兰州

²甘肃省智能信息技术与应用工程研究中心, 甘肃 兰州

收稿日期: 2023年4月28日; 录用日期: 2023年5月21日; 发布日期: 2023年5月31日

摘要

针对传统单线激光SLAM算法, 在地图构建过程中, 由于单线激光单帧点云数据量过少, 在某些局部相似的场景中可能出现得分很高的误匹配情况, 本文对点云配准算法的迭代环节经行改进, 并提出一种滑动窗口的点云拼接方法, 利用新拼接的点云组, 进行地图构建和回环检测。分别在仿真与实际环境中对本文所提算法经行验证, 在仿真环境中本文改进的点云配准算法相对于原来的点云配准算法, 迭代次数减少50.22%, 运行时间减少39.67%; 在真实环境中利用机器人平台进行定位与地图构建, 结果表明本文所提算法在定位精度、建图精度和实时性上提升明显, 可以有效解决单线激光运行过程中的误匹配情况。

关键词

SLAM, 单线激光, 点云配准, 点云拼接

Research on Single Line Laser Point Cloud Splicing SLAM Algorithm

Jie Zhang^{1,2*}, Dong Pei^{1,2}, Wenhui Gao^{1,2}

¹College of Physics and Electronic Engineering, Northwest Normal University, Lanzhou Gansu

²Engineering Research Center of Gansu Province for Intelligent Information Technology and Application, Lanzhou Gansu

Received: Apr. 28th, 2023; accepted: May 21st, 2023; published: May 31st, 2023

*通讯作者。

Abstract

Aiming at the traditional single-line laser SLAM algorithm, in the process of map construction, due to the small amount of single-line laser single-frame point cloud data, there may be mismatches with high scores in some locally similar scenes. In this paper, the iterative link of the point cloud registration algorithm is improved, and a sliding window point cloud splicing method is proposed. Using the newly spliced point cloud group, we carry out map construction and loop detection. The proposed algorithm was verified in simulation and real environment respectively. In the simulation environment, compared with the original point cloud registration algorithm, the iteration times and running time of the improved point cloud registration algorithm were reduced by 50.22% and 39.67%. In the real environment, the robot platform is used for positioning and map construction. The results show that the proposed algorithm improves the positioning accuracy, map construction accuracy and real-time performance significantly, and can effectively solve the mismatching situation during the operation of single line laser.

Keywords

SLAM, Single-Wire Laser, Point Cloud Registration, Point Cloud Splicing

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

当机器人处于一个未知环境时,为使移动机器人实现自主行走,必须同时解决定位和地图创建问题,也称(Simultaneous Localization and Mapping, SLAM) [1]问题。激光 SLAM [2] [3]是机器人根据其所携带的激光传感器,确定自身在地图中的位置及姿态,实现机器人自主行走。单线激光 SLAM [4] [5]中由于单帧激光点云数据量过少,在某些局部相似的场景中可能出现得分很高的误匹配情况。针对这个问题,文献[6]在激光传感器的基础上,引入惯性测量单元(Inertial Measurement Unit, IMU),通过 IMU 获得两个扫描点云的初始位姿估计,并提取两个二维点集的线特征来完成快速变化场景的匹配,减少误匹配情况。文献[7]将激光扫描点利用金字塔采样得到不同分辨率的概率网格,提高匹配效率,进一步的减少了局部的累计误差,但增加了计算量。文献[8]结合 WiFi 定位,对 WiFi 信号强度加权匹配,利用卡尔曼滤波器实现激光雷达与 WiFi 的融合定位结果。文献[9]根据机械雷达的固有特性,在匹配阶段,只查找相邻两条激光线之间的内点,提高了近邻点索引的精度和效率。这些方法都在一定程度上提高了算法精度,但都存在一个问题,引入其他传感器或者改进匹配方式都增大了计算量,需要性能更高的硬件平台才能保证 SLAM 系统的实时性。为此本文在不引入其他传感器和不提高硬件平台性能的前提下,对单线激光雷达的多帧激光点云进行拼接使得有效点云数目增多,获得更多的环境信息,提高地图构建与回环检测的精度。

本文算法流程分为以下几步,首先将传感器数据预处理去除噪声点,然后利用点云配准的算法计算出相邻帧之间的位姿,再对点云进行拼接,点云的拼接提出一种滑动窗口拼接的方法,因为激光频率很高,可以在一秒钟提取一帧激光数据,十秒为一组实时更新拼接点云,在室内低速运动场景下鲁棒性就会比较高,利用拼接好的点云组建图,并在最后根据新拼接的点云组与全局地图匹配进行回环检测,把

上述过程结合就是一个完整的二维激光的定位方案。本文 SLAM 系统全局流程如图 1 所示：

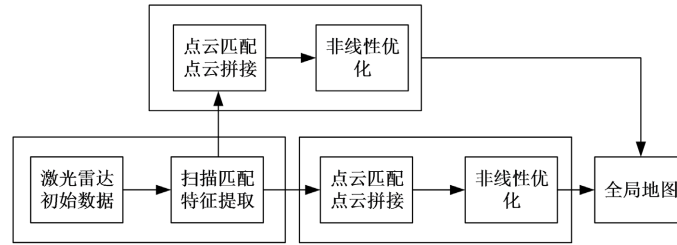


Figure 1. Global process

图 1. 全局流程

2. 点云配准与拼接

激光雷达扫描得到初始数据不仅包含点云数据还有噪声点，需要经过处理才能使用，常见的点云预处理手段有直通滤波，体素滤波，统计滤波，半径滤波等[10] [11]。本文对点云数据的预处理采用体素滤波算法，它的原理是在二维平面上，通过划分小的栅格，用小栅格内的质心代替小栅格内的点云信息，从而简化栅格平面中的点云，使用此方法能够消除点云中的离散点和噪声。

体素滤波后的点云数据，需要通过点云配准算法求解相邻点云间的位姿。针对点云配准的算法也比较多，其中比较著名的有 Besl 提出的迭代最近点(Iterative Closest Point, ICP) [12]算法，ICP 算法利用最小二乘法，求得点与点之间最小的距离误差，该方法的优点是无需对位置进行初始估计，直接通过数学计算求解刚体变换；Fischler 提出的随机样本一致性(RANSAC) [13]方法，可以避免整个点云的迭代，从而显著提高算法效率，在二维和三维数据处理中得到了广泛应用；正态分布变换(NDT) [14]方法用概率密度函数来确定两个点云间的最优的匹配，简化了特征匹配，因为不需要特征计算，所以配准时间较短。本文点云配准选择 ICP 算法，并对 ICP 算法的迭代环节经行改进。具体过程如下：设激光雷达采集得到的相邻两组数据点集为 P_k 和 P_{k+1} ，则相邻两帧点集之间的刚体变换为：

$$P'_{k+1} = RP_{k+1} + T \quad (1)$$

其中 $R = \begin{bmatrix} \cos \Delta\varphi & \sin \Delta\varphi \\ -\sin \Delta\varphi & \cos \Delta\varphi \end{bmatrix}$ 为 P_k 和 P_{k+1} 之间的旋转矩阵， $T = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$ 为 P_k 和 P_{k+1} 之间的平移向量。 $\Delta\varphi$ 为 P_k 和 P_{k+1} 之间的旋转角度， Δx 为 P_{k+1} 在 x 轴方向的位移， Δy 为 P_{k+1} 在 y 轴方向上的位移。

通过参数变换，将 P_{k+1} 中的点集，转换到 P_k 的坐标系下，实现点集匹配。在机器人连续运动采集的情况下，相邻两组激光点集中，实际只有部分点集能够重合，其余点需尽量做到距离最小，表示为：

$$\min(|P_k - P'_{k+1}|) \quad (2)$$

此时激光点集匹配问题可以描述为求解 R 、 T 使得所表示的目标评价函数达到最小值：

$$\min(g(R, T)) = \min\left(\frac{1}{n} \sum_{k=1}^n |P_k - RP_{k+1} - T|^2\right) \quad (3)$$

目标评价函数求解过程如下：首先计算两组点云数据的中心点，分别表示为 μ^{P_k} 与 $\mu^{P_{k+1}}$ ，计算公式为：

$$\begin{cases} \mu^{P_k} = \frac{1}{n} \sum_i^n a_i, a_i \in P_k \\ \mu^{P_{k+1}} = \frac{1}{n} \sum_i^n b_i, b_i \in P_{k+1} \end{cases} \quad (4)$$

其中 $\{a_i\}$ 为 P_k 中的点集, $\{b_i\}$ 为 P_{k+1} 中的点集, 然后对 P_k 和 P_{k+1} 中的数据进行归一化处理:

$$\begin{cases} a'_i = a_i - \mu^{P_k} \\ b'_i = b_i - \mu^{P_{k+1}} \end{cases} \quad (5)$$

得到新的目标函数:

$$\begin{aligned} g(R, T) &= \frac{1}{n} \sum_{i=1}^n |a'_i - Rb'_i|^2 \\ &= \frac{1}{n} \left(\sum_{i=1}^n |a'_i|^2 + \sum_{i=1}^n |Rb'_i|^2 - 2 \sum_{i=1}^n |a'_i Rb'_i|^2 \right) \end{aligned} \quad (6)$$

想要使目标函数最小, 只需 $\frac{1}{n} \sum_{i=1}^n |a'_i Rb'_i|^2$ 最大, 令:

$$\text{err}(\Delta\varphi) = \frac{1}{n} \sum_{i=1}^n |a'_i Rb'_i| \quad (7)$$

其中 $\Delta\varphi$ 为两组点集之间的旋转角度, 对式(7)中的 $\Delta\varphi$ 求导:

$$\frac{\partial \text{err}(\Delta\varphi)}{\partial \Delta\varphi} = \frac{1}{n} \sum_{i=1}^n \left(\cos \Delta\varphi (a_i^x b_i^y - a_i^y b_i^x) - \sin \Delta\varphi (a_i^x b_i^x + a_i^y b_i^y) \right) \quad (8)$$

其中 a_i^x , a_i^y , b_i^x , b_i^y 分别为 a'_i 和 b'_i 的 x , y 分量, 进一步可得到:

$$\frac{\sin \Delta\varphi}{\cos \Delta\varphi} = \frac{\sum_{i=1}^n (a_i^x b_i^y - a_i^y b_i^x)}{\sum_{i=1}^n (a_i^x b_i^x + a_i^y b_i^y)} \quad (9)$$

即:

$$\Delta\varphi = \arctan \frac{\sum_{i=1}^n (a_i^x b_i^y - a_i^y b_i^x)}{\sum_{i=1}^n (a_i^x b_i^x + a_i^y b_i^y)} \quad (10)$$

此时得到 $\Delta\varphi$, 就可以求得旋转矩阵 R , 经过参数变换后的两组点云数据中心重合, 即 $\mu^{P_k} = \mu^{P_{k+1}}$ 通过式(1)可得到平移向量 T :

$$T = \mu^{P_k} - R\mu^{P_{k+1}} \quad (11)$$

第一次得到的 R , T 如果直接使用, 对应点对之间的距离可能不是最小, 还需经过多次迭代才能得到最优解。为了减小迭代次数, 提高算法效率, 本文设计了一种迭代自适应终止策略, 传统 ICP 算法每迭代一次目标点集与参考点集都会不断靠近, 为了能够保证精度的同时不发生过度迭代的问题, 本文利用平均绝对值误差(Mean Absolute Error, MAE)构建了一个迭代自适应终止策略, 并将本文改进后的 ICP 算法命名为(Iterative adaptive ICP, IA-ICP), 具体过程为: 通过计算第 n 次和第 $n-1$ 次对应点之间欧氏距离的 MAE 值, 采用 MAE 的大小来描述第 n 次和第 $n-1$ 次恢复块内容的相似性, MAE 值越小, 表明两

次迭代内容越相似, 进一步迭代带来的进步大, 本文设定当 $\text{MAE}_n / \text{MAE}_{n-1} \geq 0.95$ 时, 停止迭代。其中 MAE 求解方式如式(12)所示:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |a_i - b'_i| \quad (12)$$

其中 $\{a_i\}$ 为 P_k 中的点集, $\{b_i'\}$ 为 P'_{k+1} 中的点集, n 为对应点数目。

求得最优 R 、 T 后, 对相邻激光帧点云进行拼接, 使得有效点云数目增多获得更多的环境信息, 采用一种滑动窗口的策略, 设滑动窗口大小为 10, 利用 IA-ICP 算法依次求出窗口内 10 帧数据相邻之间的位姿, 并把窗口内帧数据转移到第一帧数据坐标系下, 每更新一帧数据, 删除最旧的一帧数据, 并将剩余 9 帧点云数据转移到最新一帧点云数据的坐标下, 每 10 帧作为一个拼接好的点云组。本文点云拼接算法流程如下:

<1> 系统开始运行, 窗口内没有点云数据, 此时将预处理后的待加入帧点云数据直接移入窗口内作为窗口内第 1 帧, 如图 2 所示:

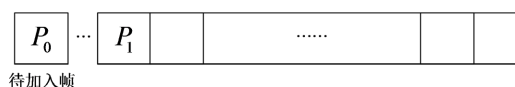


Figure 2. The system just started running
图 2. 系统刚开始运行

然后利用 IA-ICP 算法, 求出下一个待加入帧与窗口内第 1 帧之间的 R 、 T , 将该待加入帧移入窗口作为第 1 帧, 窗口内原来的第 1 帧移动到第 2 帧位置, 但实际过程中由于机器人快速转向等原因会导致丢帧, 出现相邻两帧之间距离过大的问题, 为解决这个问题, 相邻帧之间建立距离约束, 如果新的待加入帧与窗口内第 1 帧对应点云之间的距离满足约束条件, 则将待加入帧移入窗口内, 否则删除, 具体过程为: 设新的待加入帧点云的中心为 μ^b , 窗口内第一帧点云的中心为 μ^a , 两中心点之间的距离为:

$$d = \frac{[\text{dist}(x_0, y_0) - \text{dist}(x_1, y_1)]^2}{[\text{dist}(x_0, y_0) + \text{dist}(x_1, y_1)]^2} \leq \delta_1 \quad (13)$$

式中, δ_1 为设定的距离误差, 若滑动窗口外待加入帧与窗口内第 1 帧距离误差满足式(13), 则将待加入帧移入滑动窗口内作为第 1 帧, 原本窗口内的第 1 帧移到第 2 帧位置, 再利用 IA-ICP 算法, 求出下一个待加入帧与窗口内第 1 帧点云之间的 R 、 T , 依次对待加入帧利用式(13)进行判断直至滑动窗口内满 10 帧, 如图 3 所示:

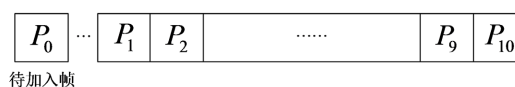


Figure 3. 10 frames full in the window
图 3. 窗口内满 10 帧

当滑动窗口内满 10 帧时, 将滑动窗口内第 2 帧至第 10 帧的数据的通过位姿变换转移到窗口内第 1 帧数据的坐标系下, 拼接形成一个点云组:

$$\begin{cases} P_1 = P_1 \\ P'_2 = R_2 P_2 + T_2 \\ P'_3 = R_2 (R_3 P_3 + T_3) + T_2 \\ \vdots \\ P'_{10} = R_2 (R_3 (\dots (R_{10} P_{10} + T_{10}) \dots) + T_3) + T_2 \end{cases} \quad (13)$$

<2> 当滑动窗口内已有 10 帧数据, 此时窗口内第 10 帧数据将作为待移除帧, 当新的符合条件的待

加入帧进入到窗口内, 此时将窗口内第 10 帧移出窗口外作为废弃帧, 窗口内其余帧依次右移形成新的点云组, 如图 4 所示:

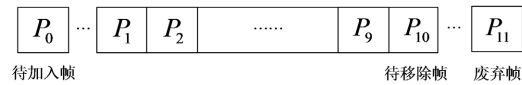


Figure 4. The system is running properly
图 4. 系统正常运行中

<3> 在系统运行过程中可能出现连续多帧待加入帧与滑动窗口内第一帧之间的约束都不满足式(13), 此时设定阈值 k , 如果连续 k 帧待加入帧与滑动窗口内第一帧之间的距离都不满足式(13), 则清空窗口内的点云, 重新按照步骤<1>进行。

3. 实验分析

为验证本文算法的有效性, 搭建移动机器人平台进行测试实验。机器人平台如图 5 所示, 机器人底层由 STM32 控制四个麦克纳姆轮实现全向运动, 并搭载 Hokuyo 激光雷达来获取机器人周围环境信息, 上位机为工控机, 其配置为 Intel i7-1165G7, 8GB 内存, 系统为 Ubuntu18.04, ROS 版本为 Melodic。

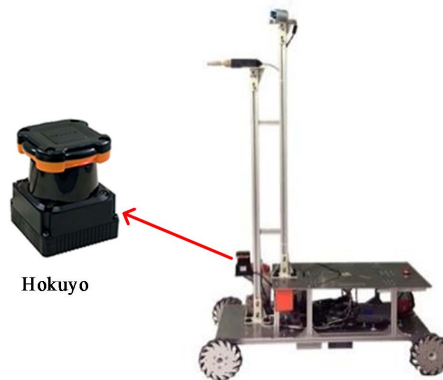


Figure 5. Robot platform
图 5. 机器人平台

实验检验分为三部分, 一: 传统 ICP 算法和 IA-ICP 算法对比; 二: 多种 SLAM 算法轨迹对比; 三: 多种 SLAM 算法建图精度对比。所有实验中机器人的角速度都为 0.1 rad/s, 线速度为 0.2 m/s。

3.1. ICP 算法迭代对比

在相同的条件下, 对传统 ICP 算法和本文提出的 IA-ICP 算法进行比较, 对比迭代次数, 对比结果如图 6 所示, 平均迭代次数与耗时对比如表 1 所示:

Table 1. Performance comparison
表 1. 性能对比

算法	平均迭代次数/次	时间/s
ICP	9.12	1.21
IA-ICP	4.54	0.73

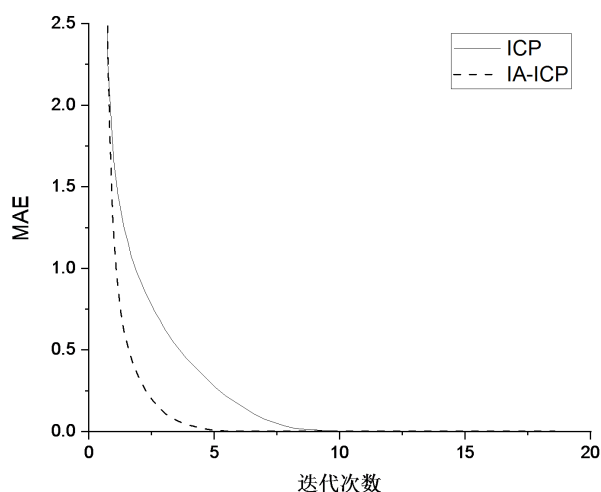


Figure 6. Comparison of iterations
图 6. 迭代次数对比

由图 6 可以明显的看出，本文 IA-ICP 算法比传统 ICP 算法完成迭代所用的次数更少，迭代曲线收敛的更快，由表 1 可知 IA-ICP 算法比较传统 ICP 算法迭代次数平均减少 50.22%；迭代次数减少，会伴随着算法运行时间减少，IA-ICP 算法在运行时间上，相比于传统 ICP 算法减小 39.67%，因此 IA-ICP 算法有更高的效率。

3.2. 轨迹对比：

实验环境为教学楼内走廊，如图 7 所示，具有地面平坦，行人少等优点，适合移动机器人测试，机器人运动路径如图中蓝色箭头所示：

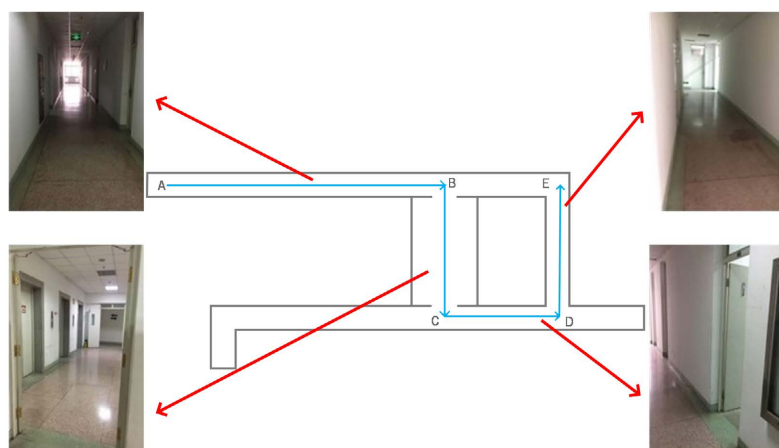


Figure 7. Experimental environment
图 7. 实验环境

在图 7 所示实验场景中，分别使用改进前的方式与改进后的方式，以及 Gmapping 算法进行实验和文献[15]中的方法，每种算法分别按图 7 中蓝色路径运行 5 次，并对 5 次 SLAM 轨求平均值，得到图 8 中的轨迹曲线。

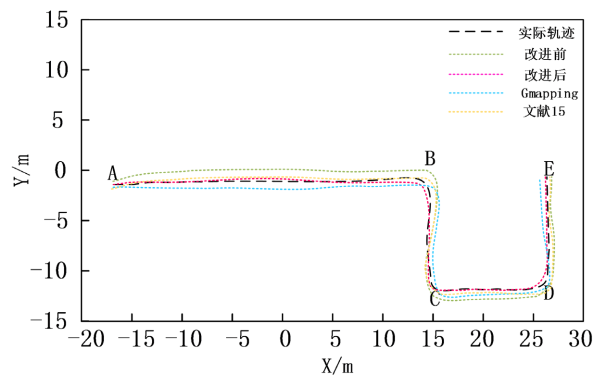


Figure 8. Trajectory comparison
图 8. 轨迹对比

从图 8 轨迹曲线可以看出, 本文所提出的方法最接近真实路径, 其次文献[15]中的方法, 然后是 Gmapping 算法, 与偏离真实轨迹最大的是本文未改进前的算法。由此说明通过滑动窗口拼接能够很好的估计机器人的位姿, 滑动窗口拼接的点云组在前期建图提供了良好的精度, 同时又有利于后期回环检测, 使得机器人运动轨迹与实际轨迹偏差不大, 有效减小了单线激光误匹配情况。三种算法在运行过程中在 x 轴、 y 轴的平均位移偏差如表 2 所示:

Table 2. Mean displacement deviation
表 2. 平均位移偏差

算法	X/m	Y/m
本文改进前	0.42	0.35
本文改进后	0.16	0.13
Gmapping	0.31	0.28
文献[15]	0.28	0.26

从表 2 的数据可以看出, 本文改进后的算法, 误差更小, 精度更高, 在 x 轴方向上, 本文改进后的方法相比 Gmapping 算法的轨迹精度提高 48.39%, 相比文献[15]中的算法轨迹精度提高 42.86%, 相比本文未改进前的算法的轨迹精度提高 61.90%; 在 y 轴方向上, 本文改进后的方法相比 Gmapping 算法的轨迹精度提高 53.57%, 相比文献[15]中的算法轨迹精度提高 50.00%, 相比本文未改进前的算法的轨迹精度提高 62.86%。

3.3. 建图对比

在图 7 所示的实验场景中, 使用本文改进前的算法、本文改进后的算法, 以及 Gmapping 算法, 构建地图结果如图 9 所示。

由表 3 可以看出, 本文改进后的算法相对于 Gmapping 算法构建的整体地图最大误差减小 43.75%, 平均误差减小 19.23%, 匹配耗时减小 28.57%; 相对与本文为改进前的算法构建的整体地图最大误差减小 61.70%, 平均误差减小 43.24%, 匹配耗时减小 36.62%。最终本文算法的准确率达到 95.4%, 相比于 Gmapping 算法的 90.10% 和本文算法未改进前的 86.71% 都有了较大的提升。

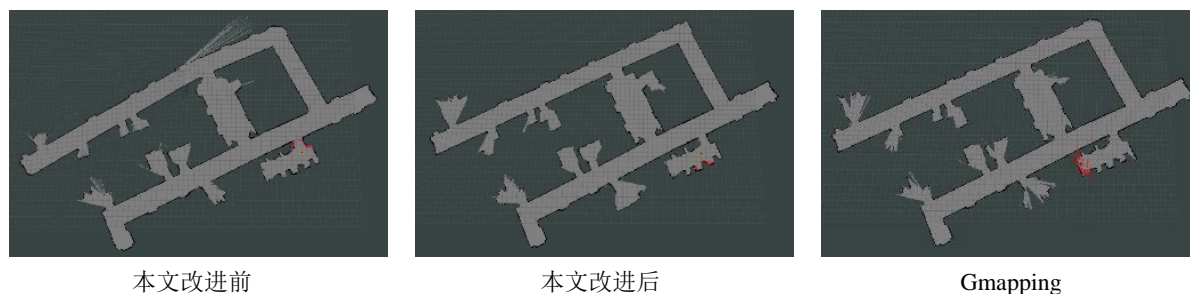


Figure 9. Drawing comparison
图 9. 建图对比

Table 3. Performance comparison of SLAM algorithms
表 3. SLAM 算法性能对比

算法	最大误差(m)	平均误差(m)	匹配耗时(s/次)	准确率/%
本文改进前	0.47	0.37	7.1	86.7
本文改进后	0.18	0.21	4.5	95.4
Gmapping	0.32	0.26	6.3	90.1

4. 结论

本文针对传统单线激光 SLAM 由于单帧点云数据量过少, 在某些局部相似的场景中可能出现得分很高的误匹配情况, 提出了一种基于滑动窗口的点云拼接算法, 并对传统点云配准算法进行改进, 提出一种迭代自适应终止的 ICP 算法提高了点云配准的速度。先利用迭代自适应终止 ICP 算法求出点云间的位姿, 然后通过滑动窗口拼接形成一个局部地图, 利用局部地图进行地图构建和回环检测, 经过实验验证, 本文所提出的方法在室内低速环境下能够准确实现室内定位与地图构建, 具有良好的实用价值。

参考文献

- [1] Chatila, R. and Laumond, J. (1985) Position Referencing and Consistent World Modeling for Mobile Robots. 1985 *IEEE International Conference on Robotics and Automation*, St. Louis, 25-28 March 1985, 138-145.
- [2] 丁元浩, 吴怀宇, 陈洋. 基于 2D 激光扫描到子图匹配的 SLAM 方法[J]. 计算机工程与设计, 2020, 41(12): 3458-3463.
- [3] Singh, R. and Nagla, K.S. (2018) Improved 2D Laser Grid Mapping by Solving Mirror Reflection Uncertainty in SLAM. *International Journal of Intelligent Unmanned Systems*, 6, 93-114. <https://doi.org/10.1108/IJIUS-01-2018-0003>
- [4] 赵若愚, 关志伟, 童敏勇, 等. 基于单线激光雷达的 SLAM 系统功能优化设计[J]. 中国汽车, 2021(2): 4-9, 43.
- [5] 孙琪, 王世峰, 王开鑫, 陈森, 刘传义. 基于单线激光雷达的室内环境建图方法研究[J]. 长春理工大学学报(自然科学版), 2020, 43(4): 37-42.
- [6] Xu, X., Luo, M., Tan, Z. and Zhang, M. (2018) Improved ICP Matching Algorithm Based on Laser Radar and IMU. 2018 *5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, Nanjing, 23-25 November 2018, 517-520. <https://doi.org/10.1109/CCIS.2018.8691139>
- [7] Olson, E. (2015) M3RSM: Many-to-Many Multi-Resolution Scan Matching. 2015 *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, 26-30 May 2015, 5815-5821. <https://doi.org/10.1109/ICRA.2015.7140013>
- [8] 胡钊政, 刘佳蕙, 黄刚, 陶倩文. 融合 WiFi、激光雷达与地图的机器人室内定位[J]. 电子与信息学报, 2021, 43(8): 2308-2316.
- [9] Kovalenko, D., Korobkin, M. and Minin, A. (2019) Sensor Aware Lidar Odometry. 2019 *European Conference on Mobile Robots (ECMR)*, Prague, 4-6 September 2019, 1-6. <https://doi.org/10.1109/ECMR.2019.8870929>
- [10] 焦晨, 王宝锋, 易耀华. 点云数据滤波算法研究[J]. 国外电子测量技术, 2019, 38(11): 18-22.

- [11] 黄德伦, 高波, 白浩, 郭腾. 改进点云滤波算法在地形图测绘中的应用[J]. 测绘技术装备, 2022, 24(4): 65-71.
- [12] Besl, P.J. and McKay, N.D. (1992) A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**, 239-256. <https://doi.org/10.1109/34.121791>
- [13] Fischler, M.A. and Bolles, R.C. (1981) Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, **24**, 381-395. <https://doi.org/10.1145/358669.358692>
- [14] Magnusson, M., Andreasson, H., Nuchter, A. and Lilienthal, A.J. (2009) Appearance-Based Loop Detection from 3D Laser Data Using the Normal Distributions Transform. 2009 *IEEE International Conference on Robotics and Automation*, Kobe, 12-17 May 2009, 23-28. <https://doi.org/10.1109/ROBOT.2009.5152712>
- [15] 夏天, 任彧. 基于加权 SDF 地图的二维激光 SLAM 方法[J]. 计算机应用与软件, 2022, 39(4): 143-148.