

# 基于2D激光雷达的移动机器人 同步定位与地图构建算法 研究

汤玉春

上海理工大学光电信息与计算机工程学院, 上海

收稿日期: 2023年4月12日; 录用日期: 2023年6月21日; 发布日期: 2023年6月30日

## 摘要

文中提出一种将激光雷达和轮式里程计相融合的方法, 用于提升SLAM (Simultaneous Localization and Mapping)算法前端点云配准效率和点云畸变矫正效果。在点云配准方面, 对PL-ICP算法做出改进, 首先对激光点云进行预处理去除无效点, 再利用自适应体素滤波方法对激光点云进行下采样, 在保留点云特征的同时将点云稀疏化, 从而减少点云配准的计算量, 利用轮式里程计的测量值为点云配准提供初值, 提高点云配准的效率和定位效果。在点云畸变矫正方面, 按照轮式里程计测量的机器人位姿的时间戳, 利用拉格朗日线性插值法对点云配准配得到的机器人位姿进行线性插值, 利用EKF算法融合轮式里程计测量的位姿和点云配准插值得到的位姿对轮式里程计的测量误差做出矫正, 然后为激光点云提供运动补偿, 从而去除点云畸变提升SLAM算法定位和建图效果。利用ROS搭建仿真环境验证了本文提出的算法的有效性。

## 关键词

SLAM, 轮式里程计, PL-ICP, EKF, 多传感器融合, 点云畸变矫正

# Simultaneous Localization and Map Construction Algorithm for Mobile Robots Based on 2D LiDAR

Yuchun Tang

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Apr. 12<sup>th</sup>, 2023; accepted: Jun. 21<sup>st</sup>, 2023; published: Jun. 30<sup>th</sup>, 2023

## Abstract

This paper proposes a method for fusing data from a lidar and wheel odometry to improve the efficiency of front-end point cloud registration and distortion correction in SLAM (Simultaneous Localization and Mapping) algorithms. Specifically, in terms of point cloud registration, the paper improves the PL-ICP algorithm by first preprocessing the lidar point cloud to remove invalid points, and then using an adaptive voxel filtering method to down sample the lidar point cloud, which reduces the computational load of point cloud registration while preserving the point cloud features. In addition, the paper uses the measurement values from the wheel odometry to provide an initial value for point cloud registration, which improves the efficiency and accuracy of point cloud registration. In terms of point cloud distortion correction, the paper uses the timestamp of the robot pose measured by the wheel odometry, and applies Lagrangian linear interpolation to linearly interpolate the robot pose obtained from point cloud registration. The paper then uses the EKF algorithm to fuse the robot pose measured by the wheel odometry and the pose obtained from point cloud registration to correct for the measurement errors of the wheel odometry. Finally, the paper provides motion compensation for the lidar point cloud to remove point cloud distortion and improve the localization and mapping effects of the SLAM algorithm. The paper verifies the effectiveness of the proposed algorithm by using ROS to build a simulation environment.

## Keywords

SLAM, Wheel Odometry, PL-ICP, EKF, Multi-Sensor Fusion, Point Cloud Distortion Correction

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

SLAM 的关键技术根据传感器在陌生环境中获取自身的位姿信息和环境信息, 估计自身运动的位置进行实时定位并利用这些信息增量式的构建环境地图[1], SLAM 算法按照传感器不同可分为基于相机的视觉 SLAM [2] [3]基于激光雷达的激光 SLAM [4] [5]。相机使用便捷、能够获取丰富的环境信息、但容易受到外部环境的干扰。激光雷达抗干扰能力强、受光照影响弱、能直接获取深度信息。其中 2D 激光雷达因其售价便宜, 对算力要求低, 易于构建栅格地图等特点而被广泛应用于机器人领域。

仅使用激光雷达进行定位与建图时, 在变速场景和特征较少的退化场景中点云配准效率低、定位效果差, 在移动机器人快速运动时激光点云易发生畸变, 最终导致建图效果差, 因此多传感器的融合[6] [7]成了近几年的研究热点。Chen S 等人[8]将激光雷达和相机结合, 通过构建视觉相似性的词袋模型提升回环检测的效果, 以提高 SLAM 算法的后端优化性能, 但在纹理缺失的场景鲁棒性较差。章弘凯等人[9]提出将 IMU 和激光雷达相结合, 用 IMU 的预积分结果为点云配准提供初值提升点云配准效率, 同时将其积分结果作为约束条件加入到 SLAM 算法的后端优化中减小轨迹漂移的影响提高载体的定位精度, 但 IMU 对线加速度进行二次积分的误差较大。Hess W 等人[10]提出了基于图优化的 Cartographer 算法, 经过不断的发展已经能和 GPS、IMU、轮式里程计等多种传感器融合, 但仅将辅助传感器的数据用于未来帧的位姿推, 并未用于点云畸变矫正。

激光点云畸变会使环境地图发生偏移, 进而影响到后端优化和回环的效果最终影响移动机器人定位

精度和建图效果。点云畸变的矫正最直接的方法是利用高频率的激光雷达，在激光雷达频率较高情况下点云畸变可以忽略，但这会带来高昂的硬件成本。M. Roh 等人[11]利用 GPS/INS 传感器对点云信息进行校正，在每次扫描的同时获取激光雷达载体的确切位置和方向，但是其测量方式依赖卫星，因此难以应用于室内或隧道等封闭环境。基于帧间配准的纯运动估计法因其不需要额外辅助的传感器而被广泛的应用于点云畸变矫正[12] [13] [14]。如果激光雷达载体在一个扫描周期内的运动状态非匀速或则激光频率太低，使用该方法对点云矫正的效果并不理想。Yang W 等人[15]提出一种激光雷达和相机融合方法，通过全速度估计来进行点云畸变矫正。激光雷达在径向方向上能精确测量距离信息，但只能使用稀疏的角度信息，而相机作为补充传感器可以提供密集的角度分辨率，此外，还利用概率卡尔曼滤波方法融合估计的速度提升点云矫正效果，但相机提供的信息易受环境干扰。Zhang B 等人[16]提出利用 UKF 融合激光雷达和 IMU 数据的方式进行点云畸变矫正，UKF 用于融合 IMU 的角速度、线加速度和激光雷达的位姿，得到 IMU 的角速度偏差、线加速度偏差和激光雷达的线速度，然后得到激光雷达在一个扫描周期内各时刻的位姿，从而消除点云的失真得到激光雷达在一个扫描周期内各时刻的位姿，从而消除点云的失真。虽然 IMU 能测量激光雷达载体的角速度和线加速，但线速度累计误差较大。

轮式里程计能够直接测量移动机器人的位移和角度，具有较高的局部位置测量精度和局部角度测量精度，并且具有较高的位姿更新频率，通常能达到 100~200 Hz 左右，本文将轮式里程计和激光雷达相融合用于点云畸变矫正，和提升在不同场景下的点云配准效率和定位效果。

## 2. 移动机器人模型与点云配准

### 2.1. 移动机器人运动模型

轮式里程计通过测量移动机器人两主动轮的转速得到移动机器人的线速度  $v$  和角速度  $w$ ，从而推算出移动机器人在世界坐标系下的位姿变化信息。在世界坐标系下移动机器人两轮差分底盘运动学模型如图 1 所示。

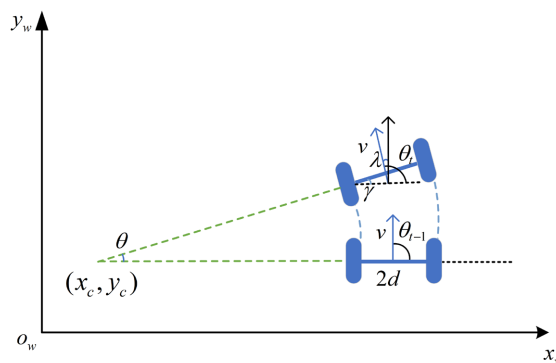


Figure 1. Two-wheel differential chassis kinematic model  
图 1. 两轮差分底盘运动学模型

设左轮转速为  $w_l$ ，半径为  $r_l$ ，右轮的转速为  $w_r$ ，半径与左轮相同，两轮间的距离为  $2d$ ，移动机器人的角速度为和线速度分别为  $w$  和  $v$ ，计算如式(1)。

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{r_l}{2} & \frac{r_r}{2} \\ -\frac{r_l}{2d} & \frac{r_r}{2d} \end{bmatrix} \begin{bmatrix} w_l \\ w_r \end{bmatrix} = J \begin{bmatrix} w_l \\ w_r \end{bmatrix} \quad (1)$$

在间隔时间  $\Delta t_0$  极短时，假设其线速度角速度恒定，移动机器人运动变化的圆心角  $\theta$  的计算如式(2)。

$$\begin{cases} w = J_{21}w_l + J_{22}w_r \\ \theta = w\Delta t \end{cases} \quad (2)$$

$$\begin{cases} v = J_{11}w_l + J_{12}w_r \\ x_c = x_{t-1} - \frac{v}{w}\sin(\theta_{t-1}) \\ y_c = y_{t-1} + \frac{v}{w}\cos(\theta_{t-1}) \end{cases} \quad (3)$$

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} -\frac{v}{w}\sin\theta_{t-1} + \frac{v}{w}\sin(\theta_{t-1} + w\Delta t) \\ \frac{v}{w}\cos\theta_{t-1} - \frac{v}{w}\cos(\theta_{t-1} + w\Delta t) \\ w\Delta t \end{bmatrix} \quad (4)$$

设  $p_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})^T$  表示移动机器人上一时刻在世界坐标系中的位姿，由几何关系可得角  $\theta$ 、 $\gamma$  和  $\lambda$  相等， $(x_c, y_c)$  表示其运动半径的圆心所在的位置，其计算如式(3)，则当前时刻移动机器人在世界坐标系的位姿  $p_t = (x_t, y_t, \theta_t)^T$  计算如式(4)。

## 2.2. 激光点云配准原理

激光 SLAM 建图主要分为前 endpoint 云配准，后端优化，和回环检测等三个部分。点云配准即寻求一组旋转和平移参数使得对齐后的上一帧点云和参考点云在同一参考坐标系下达到最大的重叠，通过求解的旋转和平移参数恢复出两帧点云数据扫描期间的移动机器人的位姿变化信息。常用的二维激光点云配准算法有 ICP [17]算法和 PL-ICP [18]算法，ICP 算法是基于点对点的距离扫描匹配算法，构造点到点的距离误差函数。 $C^j = \{c_1^j, c_2^j, c_3^j, \dots, c_i^j\}$  表示当前帧的扫描点云， $C^{j-1} = \{c_1^{j-1}, c_2^{j-1}, c_3^{j-1}, \dots, c_i^{j-1}\}$  表示与之对应的上一帧点云， $N$  表示一帧点云中激光点的个数。构建误差方程如式(5)寻求一组解  $(R(\theta), t)$  使得误差方程的值最小。

$$E(R(\theta), t) = \frac{1}{N} \sum_{i=1}^N \|c_i^{j-1} - R(\theta)c_i^j - t\|^2 \quad (5)$$

PL-ICP 则是在 ICP 的基础上利用点到线的距离作为误差取代了点到点的距离作为误差。激光点实际上是对真实环境中曲面的离散采样，最好的误差尺度是激光点到实际曲面的距离，而 PL-ICP 采用分段线性的方法对实际曲面进行近似，用当前帧点云中的激光点到上一帧点云中对应最近两激光点连线的距离来近似实际激光点到曲面的距离。PL-ICP 的误差方程更贴近实际情况，因为比起直接计算点到点的距离，其考虑了点云的局部结构，因此精度更高不容易陷入局部最优，对场景的适应性和收敛精度都高于 ICP。

## 2.3. 点云配准初值获取

无论使用哪种点云配准方法，若没有为点云配准提供准确的初值，点云配准的定位性能会快速劣化甚至完全失效。文中将轮式里程计和激光雷达相融合对 PL-ICP 算法做出改进，提高点云匹配的效率和对不同场景的适应性，具体流程如图 2 所示。

在长廊和特征物极少的退化场景中，由于没有参照物激光雷达的观测数据在某一维度高度相似，如图 3 所示。随着机器人的移动雷达坐标系由  $O_{C^{j-1}}$  移动到  $O_{C^j}$  处，但在两个坐标系下的激光点  $c_i^j$  和  $c_i^{j-1}$  的

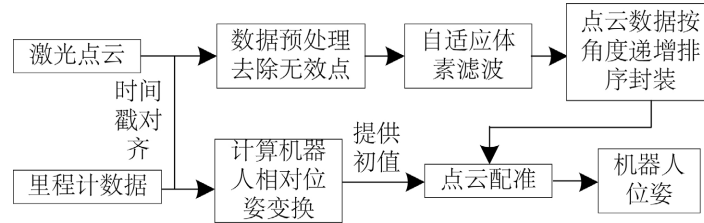


Figure 2. Flow chart of the improved PL-ICP algorithm  
图 2. 改进的 PL-ICP 算法流程图

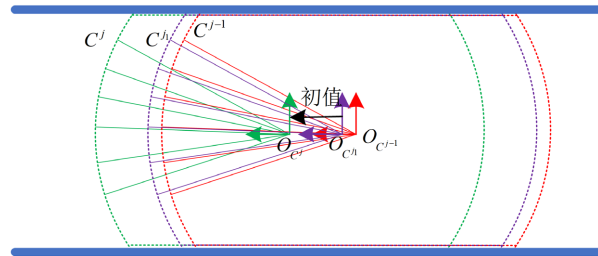


Figure 3. Lidar coordinate transformation  
图 3. 激光雷达坐标变化

距离信息完全一样，在这种情况下通过点云配准得到的机器人位姿变化信息接近为零。在有初值的情况下点云  $C^j$  变换到点云  $C^i$ ，参考坐标系由  $O_{C^j}$  变换到  $O_{C^i}$ ，再由点云  $C^i$  继续和上一帧点云  $C^{j-1}$  进行配准，配准误差可以明显减小，且初值越准确点云配准效果越好。获取点云配准初值常用的方法是利用匀速预测模型，由上一次激光点云配准的结果基于匀速假设计算激光雷达载体的运动速度，并将计算的结果用于推测未来帧的激光雷达载体在世界坐标系下的位姿，计算出激光雷达的相对位姿变化信息，然后将其用于点云配准的初值。但在变速场景和雷达频率较低的场景中匀速预测模型提供的初值误差较大，不能准确反应出激光雷达载体的位姿变换情况，最终导致点云配准的结果无法快速收敛和定位失败。

轮式里程计的频率远高于激光雷达，在一帧激光扫描时期内可以多次测量机器人位姿，因此将轮式里程计的测量值用于点云配准的初值能准确的反应出在一帧扫描时间内机器人的位姿变化。点云配准初值  $b_k = (t_k, \theta_k) = (t_{t_s, t_e}, \theta_{t_s, t_e})$ ，由轮式里程计测量的移动机器人在世界坐标系下的位姿计算相对位姿变换矩阵得到。已知一帧激光点云扫描的起止时间为  $t_s$  和  $t_e$ 。轮式里程计测量机器人位姿的频率远高于激光雷达，在  $t_s$  和  $t_e$  之间存在多帧机器人位姿信息，两个传感器的测量值存在不同的时间戳，如图 4 所示，因此需要进行时间同步。

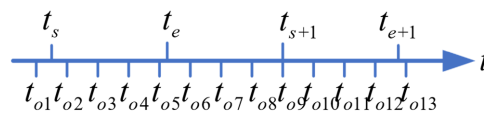


Figure 4. Different sensor timelines  
图 4. 不同传感器时间轴

用  $t_l = \{t_s, t_e, t_{s+1}, t_{e+1}, \dots, t_{s+n}, t_{e+n}\}$  表示激光点云的时间轴， $t_o = \{t_{o1}, t_{o2}, t_{o3}, t_{o4}, \dots, t_{o(j-1)}, t_{oj}\}$  表示里程计的时间轴。利用两个双端容器  $deq_l$ ， $deq_o$  分别存储点云数据和里程计信息进行时间同步，每拿到一帧点云数据则在  $deq_o$  容器中搜索对应时间的机器人位姿，若  $t_s$ 、 $t_e$  时刻没有对应的机器人位姿则分别搜索最接近  $t_s$ 、 $t_e$  时刻的位姿，然后利用线性插值得到对应的位姿，为了增加搜索效率每次对  $deq_o$  中小于  $t_e$  时刻

的位姿信息进行删除。假设  $t_s$  时刻对应的前后两帧里程计的时间戳为  $t_{o1}$ 、 $t_{o2}$ ，对应机器人位姿为  $p_{t_{o1}}$ 、 $p_{t_{o2}}$ ，则  $t_s$  时刻的机器人位姿  $p_{t_s}$  计算如式(6)同理可计算得到  $p_{t_e}$ 。轮式里程计测量得到的移动机器人位姿为  $1 \times 7$  的矩阵组成，由世界坐标系下的距离信息和四元数组成如式(7)。将  $p_{t_s}$  和  $p_{t_e}$  转换成  $4 \times 4$  矩阵形式用  $T_{t_s}$  和  $T_{t_e}$  表示，其相对位姿变换矩阵用  ${}^{t_s}T_{t_e}$  表示，计算如式(8)，计算  $t_{t_s,t_e}$  和  $\theta_{t_s,t_e}$  的值如式(9)。

$$p_{t_s} = p_{t_{o2}} \frac{t - t_{o1}}{t_{o2} - t_{o1}} + p_{t_{o1}} \frac{t_{o2} - t}{t_{o2} - t_{o1}} \quad (6)$$

$$p = [X, Y, Z, x, y, z, w] \quad (7)$$

$${}^{t_s}T_{t_e} = T_{t_s}^{-1} T_{t_e} \quad (8)$$

$$\begin{cases} t_{t_s,t_e} = \begin{bmatrix} {}^{t_s}T_{t_e}(0,3) \\ {}^{t_s}T_{t_e}(1,3) \end{bmatrix} \\ \theta_{t_s,t_e} = a \tan 2 \left( {}^{t_s}T_{t_e}(1,0), {}^{t_s}T_{t_e}(0,0) \right) \end{cases} \quad (9)$$

### 2.4. 自适应体素滤波

一帧点云激光点较多，如果将所有激光点都用于点云匹配计算量较大，并非所有激光点数据都有效，有的导致配准结果无法收敛，如激光打在桌子边缘导致返回的激光强度低、小于或大于有效距离而产生的 inf 点和 nan 点，这些噪声点可能会导致两帧点云无法通过旋转平移重叠。因此需对点云数据进行预处理去除噪声点以提高点云配准效率。设定最大测距阈值  $d_{max}$  和最近离阈值  $d_{min}$ ，初始扫描点  $c_i^j$  与  $d_{max}$  和  $d_{min}$  进行距离比较去除无效点。距离激光雷达近的物体可以扫描得到更多的激光点，这就导致点云比较密集，同时也会出现大量的重叠点，而远处的点则相对稀疏。为了降低计算量基于体素滤波对点云进行下采样，在保存点云特征的同时降低点云的稀疏性，同时能够在一定程度上去除离群点。已知点云数据在 X、Y、Z 轴上分布的最大最小值记为  $x_{max}$ 、 $x_{min}$ 、 $y_{max}$ 、 $y_{min}$ 、 $z_{max}$ 、 $z_{min}$ ，计算体素栅格的边长  $l_x$ 、 $l_y$ 、 $l_z$  如式(10)。

$$\begin{cases} l_x = x_{max} - x_{min} \\ l_y = y_{max} - y_{min} \\ l_z = z_{max} - z_{min} \end{cases} \quad (10)$$

设置体素小栅格的边长为  $l_v$ ，将 X、Y、Z 三轴均分为  $l$ 、 $w$ 、 $h$  份，则可以划分为  $l * w * h$  个体素小栅格，其中  $sum$  表示体素小栅格的个数  $sum = l * w * h$ ， $\lfloor \cdot \rfloor$  表示向下取正，其中  $l$ 、 $w$ 、 $h$  的计算如式(11)。对每个激光点所属的体素小栅格进行编号确定每个激光点所属的体素小栅格，设编号为  $(f, j, k)$ ，其中  $f$ 、 $j$ 、 $k$  的计算如式(12)。

$$l = \left\lfloor \frac{l_x}{l_v} \right\rfloor, \quad w = \left\lfloor \frac{l_y}{l_v} \right\rfloor, \quad h = \left\lfloor \frac{l_z}{l_v} \right\rfloor \quad (11)$$

$$f = \left\lfloor \frac{x_i - x_{min}}{l_v} \right\rfloor, \quad j = \left\lfloor \frac{y_i - y_{min}}{l_v} \right\rfloor, \quad k = \left\lfloor \frac{z_i - z_{min}}{l_v} \right\rfloor \quad (12)$$

$n$  表示每个小栅格内的激光点数，计算每个体素小栅格的质心  $g_i$  替代栅格内的其他激光点，其计算如式(13)。

$$g_i^j = \frac{1}{n} \sum_i^n C_i^j \quad (13)$$



得到滤波后的点云  $C^j = \{g_1^j, g_2^j, g_3^j, \dots, g_i^j\}$ , 若质心点不存在则用距离质心最近的点替代其他点。滤波阈值  $l_v$  的选取对点云的配准效果至关重要,  $l_v$  选取过大会使点云过于稀疏导致有效特征点太少而使得点云配准失败。 $l_v$  选取太小, 虽然能保留足够的激光点, 但并不能达到减少点云配准计算量的目的。机器人在移动过程中通常会遇到空间环境变化明显的情况, 如从走廊进入室内, 如果使用固定的  $l_v$  值进行体素滤波, 在不同的场景下滤波后的激光点会相差数倍, 采用固定滤波阈值会影响到点云配准的效率。因此本文使用自适应体素滤对滤波阈值的选取做出改进, 利用固定的激光点数取代固定的滤波阈值。设置最大滤波阈值  $l_{\max}$  和最小滤波阈值  $l_{\min}$  进行体素滤波, 如果滤波后的激光点数远小于或大于目标点数则利用二分法继续滤波, 如果体素滤波后激光点数接近目标点数, 则输出滤波后的点云。滤波后的平均激光点数与原始激光点数的比值用  $\gamma$  表示。在点云配准搜索对应点的时候, 找到其中一个点后, 另一个点通常在其附近, 但滤波后的激光点为无序状态, 为了提升搜索对应点的效率, 将滤波后的激光数据按照角度递增的关系进行排序。

## 2.5. 点云配准迭代

计算当前帧的激光点在上一帧激光点云所在激光雷达坐标系下的位置  $g_i^w$  如式(14), 其中  $b_k = (t_k, \theta_k)$ 。对激光点  $g_i^w$  在上一帧点云  $C^{j-1}$  中寻找距离它最近的两个激光点, 其下标记为  $i$  和  $i+1$ , 设  $n_i$  为  $g_i^{j-1}$  和  $g_{i+1}^{j-1}$  连线的法向量, 构建误差方程如(15)求解  $b_{k+1}$  使得误差方程的值最小。

$$g_i^w \triangleq g_i \oplus b_k = R(\theta_k)g_i + t_k \quad (14)$$

$$E(b_{k+1}) = \sum_i \left( n_i^T [R(\theta_{k+1})g_i^w + t_{k+1} - g_{i+1}^{j-1}] \right)^2 \quad (15)$$

将  $b_{k+1}$  作为下一次迭代的初值计算得到新的点云  $C^h$  继续和上一帧点云  $C^{j-1}$  进行配准, 求解结果用  $b_{k+2}$  表示, 循环迭代  $n$  次以后直到  $n$  等于配准次数的阈值或则  $b_{k+n}$  小于配准误差阈值则停止迭代。将每次求解的结果用矩阵  $T_{b_{k+i}}$  表示, 则最终两帧点云的配准结果  ${}_{c^{j-1}}^c T = T_{b_{k+n}} T_{b_{k+n-1}} \dots T_{b_k}$ , 迭代过程如图 5 所示。机器人初始时刻在世界坐标系下的位姿矩阵为  $T_0$ , 经过  $j$  帧点云扫描配准后定位得到的移动机器人在世界坐标系中的位姿矩阵  $T_j = T_0 {}_{c_1}^c T {}_{c_2}^c T \dots {}_{c_{j-1}}^c T$ 。改进 PL-ICP 算法的伪代码如表 1。

## 3. 激光点云畸变矫正

### 3.1. 点云畸变原理分析

激光雷达通过旋转能够覆盖  $360^\circ$  的视野, 在一次测量中, 每束激光产生一个包含距离和角度信息的单点, 一圈扫描结束所有激光点的集合构成一帧点云信息。常用的二维激光雷达的频率在  $5\sim 20$  Hz, 这就意味着激光雷达旋转一周获取周围环境的信息需要  $0.2\text{ s}\sim 0.05\text{ s}$ , 如果激光雷达在一个扫描周期内是静止的则一帧点云中每个激光点对应的移动机器人位姿是相同的, 都为激光雷达扫描开始时的第一个激光

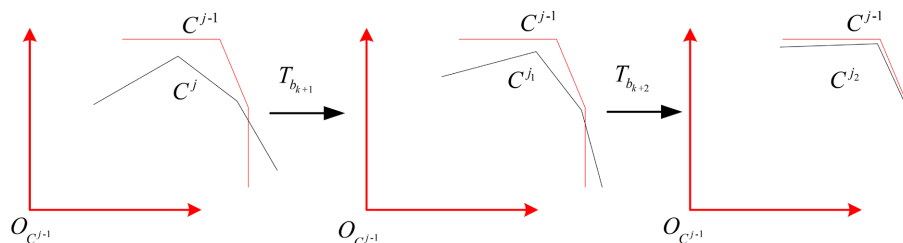


Figure 5. Point cloud alignment iterative process  
图 5. 点云配准迭代过程

**Table 1.** Improved PL-ICP algorithm pseudocode  
**表 1.** 改进 PL-ICP 算法伪代码

```

1 input:点云数据  $C^j$ , 里程计数据  $p_i$ 
2 output:一帧点云扫描期间的机器人位姿变换信息矩阵  ${}_{c^{j-1}}^j T$ 
3 begin:
4 do
5  $deq_o \leftarrow p_i, deq_l \leftarrow C^j$  //将轮式里程计数据和点云数据存入对应的双端队列
6 while( $deq_l.size < 2$ ) //防止没有对应的里程计数据, 缓存两帧雷达数据程序再继续执行
7 FindCorrespondPose ( $deq_o, deq_l, p_{t_s}, p_{t_e}$ ) //按每帧点云的开始和结束的时间戳  $t_s, t_e$  在  $deq_o$  中寻找对应的里程计位姿, 若没有对应位姿则利用附近时刻的位姿通过式(6)进行线性位姿插值。
8  $T_{t_s} \leftarrow p_{t_s}, T_{t_e} \leftarrow p_{t_e}$  //位姿矩阵变换
9  ${}_{t_s}^{t_e} T = T_{t_s}^{-1} T_{t_e}$  //计算相对位姿变换即  $t_e$  时刻机器人在  $t_s$  时刻机器人坐标系下的位姿
10  $b_k = (t_k, \theta_k) = (t_{t_s, t_e}, \theta_{t_s, t_e})$  //得到点云配准初值
11 if( $c_i^j.distance < d_{min} \parallel c_i^j.distance > d_{max}$ ) delete //数据预处理去除无效点
12 do
13 num1=VoxelFilter( $C^j, l_{max}, l_{min}$ ) //利用体素滤波对点云进行下采样 num1、num2 分别表示滤波后的激光点数和目标激光点数
14 mid=( $l_{max} + l_{min}$ )/2
15 if(num1<num2)  $l_{max} = mid$  else  $l_{min} = mid$ 
16 While num1<num2 || num1>1.1*num2
17 Sort( $C^j$ ) //利用选择排序将滤波后的激光点按照角度递增的关系进行排序
18 do
19 for i= 1:n
20 FindClosePoint( $C^j, C^{j-1}$ )
21  $E(b_{k+1}) = \sum_i (n_i^T [R(\theta_{k+1})g_i^w + t_{k+1} - c_{i+1}^{j-1}])^2$ 
22  $C^j = T_{b_{k+1}} C^j; {}_{c^{j-1}}^j T = T_{b_{k+1} c^{j-1}} T; g_i^w = c_i^j$ 
23 while  $b_{k+1} > \varepsilon$  //直到配准误差小于设定的值停止迭代
24 return  ${}_{c^{j-1}}^j T$ 
25 end

```

点所对应的移动机器人位姿。激光雷达测量到的物体的位置信息由激光雷达坐标系转换到机器人坐标系再转换到世界坐标系从而得到被测物体在整个地图中的位置。激光雷达在进行 360 度扫描时会伴随着移动机器人的移动, 这导致单个激光点在空间中有不同的参考位置, 但激光雷达的点云数据在进行封装时会默认所有激光点的信息是在同一机器人位姿下瞬时获取的, 因此点云数据会失真。如图 6 所示假设激光雷达的频率是 10 Hz, 逆时针方向旋转, 搭载激光雷达的移动机器人沿  $x$  方向运动, 线速度为 1 m/s, 机器人由位置 A 在激光雷达的四分之一个扫描周期内经过位置 B 运动到了位置 C, AC 的距离为 2.5 cm。激光雷达在位置 A 处扫描得到激光点  $s$ , 在位置 B 处扫描得到激光点  $p$ , 在位置 C 处扫描得到激光点  $t$ , 但所有激光点都会以 A 的位置为参考坐标系将其转换到世界坐标系下, 因为忽略了点 C 所在的位置, 最终呈现在地图中激光点  $t$  的位置会在点  $f$  处, 以此类推当前帧的整个点云都会发生偏移如图中的点云  $C^2$ , 而点云  $C^1$  才是激光雷达扫描得到的真实点云, 点云  $C^2$  为由于机器人移动引起的畸变的点云。点云矫正的目的就是将点云  $C^2$  还原成点云  $C^1$ 。



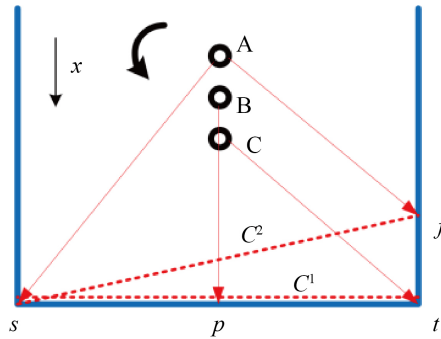


Figure 6. Point cloud distortion principle  
图 6. 点云畸变原理

### 3.2. 点云矫正流程

消除点云畸变的关键是还原出激光雷达载体在一帧点云扫描周期内准确的位姿信息。基于匀速预测模型的纯估计法对激光点云进行畸变矫正的主要步骤：1) 对于一帧激光点云中的每一激光点都赋予对应的时间戳，假设激光雷达载体匀速运动，通过帧间匹配的位移除以时间求出激光雷达载体的移动速度；2) 根据每一个激光点对应的时间戳进行运动补偿。由于运动控制器噪声的存在移动机器人运动并非匀速的，且在实际应用中移动机器人的使用不局限于匀速场景。轮式里程计能够直接测量移动机器人的位移和角度，具有较高的局部角度测量精度和局部位置测量精度，并且更新频率远高于激光雷达，相比于匀速预测模型轮式里程计的测量值能够更加准确的反应机器人的实际运动情况。设当前帧激光雷达扫描的起始时间为 $t_s$ 终止时间为 $t_e$ ， $\Delta t_l = t_s - t_e$ ，激光雷达的频率为 $f_l$ ，一帧里程计的时间为 $\Delta t_o$ ，频率为 $f_o$ ， $k = f_o \div f_l$ 。算法的整体流程如下：

- 1) 通过改进的 PL-ICP 算法配准得到 $\Delta t_l$ 时间内移动机器人的相对位姿变化信息，通过线性插值得到 $\Delta t_l$ 时间内 $k$ 个机器人位姿 $T_{l_i}$ ，计算如式(16)。
- 2) 使用 EKF 算法融合改进 PL-ICP 算法配准后插值得到的机器人位姿和轮式里程计测量的位姿，得到矫正后的机器人位姿信息，对矫正后的位姿进行线性插值，使每个激光点对应一个移动机器人位姿。
- 3) 已知机器人位姿通过坐标变换把所有激光点转换到以第一个激光点对应雷达位姿为基准的坐标系下，得到矫正后的点云数据。

$$T_{l_i} = \begin{pmatrix} t - t_s \\ t_e - t_s \end{pmatrix} T_{t_s}^{c^j} T^{c^{j-1}} \quad (16)$$

### 3.3. EKF 数据融合

在 SLAM 算法中卡尔曼滤波算法[19]常用来做多传感器数据的融合，以达到传感器载体位姿矫正的目的，其原理是利用上一时刻矫正后的载体位姿通过状态转移方程得到当前时刻的预测位姿，同时利用当前时刻的观测值修正当前时刻的预测值，得到当前时刻矫正后的位姿。但卡尔曼滤波不适用于非线性系统，为了解决非线性问题扩展卡尔曼滤波算法[20]在卡尔曼滤波算法的基础上对状态转移方程和观测方程进行一阶泰勒展开达到线性化的目的。移动机器人的运动往往是非线性的，因此本文利用扩展卡尔曼滤波算法进行数据融合。输入为轮式里程计测量位姿 $p_t$ 和激光雷达点云配准插值的位姿 $l_t$ ，输出为矫正后的机器人位姿 $\omega_t$ ，其中 $l_t$ 由 $T_{l_i}$ 转换为 $1 \times 3$ 的矩阵得到。将移动机器人的运动模型分解为一个无噪声分量和一个随机噪声分量如式(17)，式中 $u_t = (v_t, w_t)$ 表示控制量，对式(17)进行一阶泰勒展开线性化得到式(18)。  $G_{p,t}$ 和 $G_{u,t}$ 表示方程 $g(\cdot)$ 关于 $p$ 和 $u$ 求导的雅克比矩阵。 $G_{p,t}$ 和 $G_{u,t}$ 的计算如式(19)和(20)。

$$p_t = g(u_t, p_{t-1}) + N(0, R_t) \tag{17}$$

$$g(u_t, p_{t-1}) \approx g(u_t, \omega_{t-1}) + G_{p,t}(p_{t-1} - \omega_{t-1}) \tag{18}$$

$$G_{p,t} = \frac{g(u_t, \omega_{t-1})}{\partial p_{t-1}} = \begin{bmatrix} \frac{\partial x'}{\partial \omega_{t-1,x}} & \frac{\partial x'}{\partial \omega_{t-1,y}} & \frac{\partial x'}{\partial \omega_{t-1,\theta}} \\ \frac{\partial y'}{\partial \omega_{t-1,x}} & \frac{\partial y'}{\partial \omega_{t-1,y}} & \frac{\partial y'}{\partial \omega_{t-1,\theta}} \\ \frac{\partial \theta'}{\partial \omega_{t-1,x}} & \frac{\partial \theta'}{\partial \omega_{t-1,y}} & \frac{\partial \theta'}{\partial \omega_{t-1,\theta}} \end{bmatrix} \tag{19}$$

$$= \begin{bmatrix} 1 & 0 & -\frac{v_t}{w_t} \cos \omega_{t-1,\theta} + \frac{v_t}{w_t} \cos(\omega_{t-1,\theta} + w_t \Delta t) \\ 0 & 1 & -\frac{v_t}{w_t} \sin \omega_{t-1,\theta} + \frac{v_t}{w_t} \sin(\omega_{t-1,\theta} + w_t \Delta t) \\ 0 & 0 & 1 \end{bmatrix}$$

$$G_{u,t} = \frac{g(u_t, \omega_{t-1})}{\partial u_t} = \begin{bmatrix} \frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial w_t} \\ \frac{\partial y'}{\partial v_t} & \frac{\partial y'}{\partial w_t} \\ \frac{\partial \theta'}{\partial v_t} & \frac{\partial \theta'}{\partial w_t} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-\sin \theta + \sin(\theta + w_t \Delta t)}{w_t} & \frac{v_t (\sin \theta - \sin(\theta + w_t \Delta t) + w_t \cos(\theta + w_t \Delta t) \Delta t)}{w_t^2} \\ \frac{\cos \theta - \cos(\theta + w_t \Delta t)}{w_t} & \frac{-v_t (\cos \theta - \cos(\theta + w_t \Delta t) + w_t \sin(\theta + w_t \Delta t) \Delta t)}{w_t^2} \\ 0 & \Delta t \end{bmatrix} \tag{20}$$

扩展卡尔曼滤波的预测方程如式(21) (22)，当前时刻移动机器人预测位姿期望值  $\bar{p}_t$  由上一时刻校正后的位姿期望值和无误差项的移动机器人运动模型得到， $\Sigma_t$  表示位姿的协方差矩阵。 $G_{u,t} M_t G_{u,t}^T$  表示控制空间运动噪声向状态空间运动噪声的近似映射。

$$\bar{p}_t = \omega_{t-1} + \begin{bmatrix} -\frac{v}{w} \sin \theta_{t-1} + \frac{v}{w} \sin(\theta_{t-1} + w \Delta t) \\ \frac{v}{w} \cos \theta_{t-1} - \frac{v}{w} \cos(\theta_{t-1} + w \Delta t) \\ w \Delta t \end{bmatrix} \tag{21}$$

$$\bar{\Sigma}_t = G_{p,t} \Sigma_{t-1} G_{p,t}^T + G_{u,t} M_t G_{u,t}^T \tag{22}$$

观测模型  $z = h(p_t, l_t, a)$  描述的是  $p_t$  和  $l_t$  的相对位置信息，如式(23)， $a$  是传感器的误差  $(a_r, a_\beta) \sim N(0, A)$ 。其中  $z = (r, \beta)^T$ ， $r$  是距离， $\beta$  是方位角。

$$Z = \begin{bmatrix} \sqrt{(p_{t,x} - l_{t,x})^2 + (p_{t,y} - l_{t,y})^2} \\ \tan^{-1}(p_{t,y} - l_{t,y}) / (p_{t,x} - l_{t,x}) - \theta \end{bmatrix} + \begin{bmatrix} a_r \\ a_\beta \end{bmatrix} \quad (23)$$

$$A = \begin{bmatrix} \sigma_r^2 & \\ & \sigma_\beta^2 \end{bmatrix} \quad (24)$$

观测模型的泰勒展开线性近似如式(26)，式(27)为  $h$  关于机器人位姿的雅克比矩阵。

$$q = (l_{t,x} - \omega_{t,x})^2 + (l_{t,y} - \omega_{t,y})^2 \quad (25)$$

$$h(p_t, l_t) \approx h(\omega_t, l_t) + H_t(p_t - \bar{p}_t) \quad (26)$$

$$H_t = \frac{\partial h(\bar{p}_t, l_t)}{\partial p_t} = \begin{bmatrix} -\frac{l_{t,x} - \bar{p}_{t,x}}{\sqrt{q}} & -\frac{l_{t,y} - \bar{p}_{t,y}}{\sqrt{q}} & 0 \\ \frac{l_{t,y} - \bar{p}_{t,y}}{q} & -\frac{l_{t,x} - \bar{p}_{t,x}}{q} & -1 \end{bmatrix} \quad (27)$$

$$H_w = \frac{\partial h(\bar{p}_t, l_t)}{\partial z} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (28)$$

计算卡尔曼增益矩阵  $K$  如式(29)

$$\begin{cases} S_t = H_t \bar{\Sigma} H_t^T + H_w A H_w^T \\ K_t = \bar{\Sigma} H_t^T S_t^{-1} \end{cases} \quad (29)$$

$$z = \begin{bmatrix} \sqrt{(\bar{p}_{t,x} - l_{t,x})^2 + (\bar{p}_{t,y} - l_{t,y})^2} \\ \tan^{-1}(\bar{p}_{t,y} - l_{t,y}) / (\bar{p}_{t,x} - l_{t,x}) - \theta \end{bmatrix} \quad (30)$$

状态更新得到矫正后的机器人位姿的期望值和协方差如式(31)。

$$\begin{cases} \omega_t = \bar{p}_t + K_t(z_t - \bar{z}_t) \\ \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \end{cases} \quad (31)$$

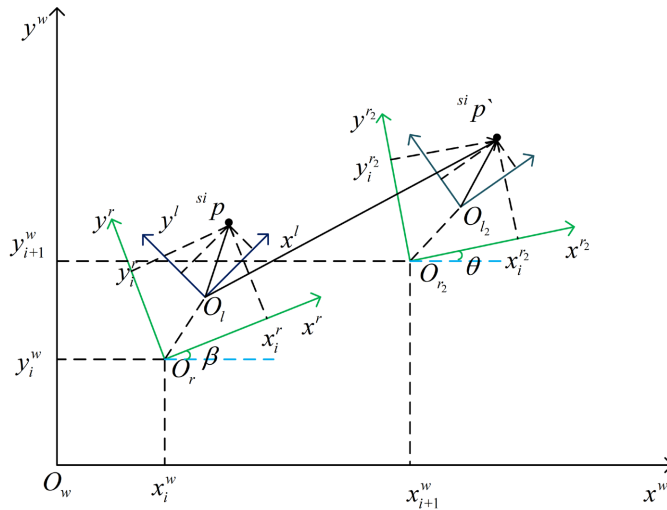
### 3.4. 点云数据矫正

从  $t_s$  到  $t_e$  时间段一共有  $k$  个机器人位姿  $\{\omega_{t_s}, \omega_{t_{s+1}}, \omega_{t_{s+2}}, \dots, \omega_{t_{s+k-2}}, \omega_{t_e}\}$ ，位姿由 EKF 融合轮式里程测量位姿和帧间配准插值的位姿得到。在  $k$  个机器人位姿之间进行线性插值，设  $\omega_{t_s}$  和  $\omega_{t_{s+1}}$  之间有  $n-2$  个位姿  $\{\omega_{t_s}, \omega_{t_{s1}}, \omega_{t_{s2}}, \dots, \omega_{t_{s(n-2)}}, \omega_{t_{s+1}}\}$ ，插值计算如式(32)。

$$\omega_t = \omega_{t_{s+1}} \frac{t - t_s}{t_{s+1} - t_s} + \omega_{t_s} \frac{t_{s+1} - t}{t_{s+1} - t_s} \quad (32)$$

在获取每个激光点对应的位姿信息后需要将所有激光点转换到以第一个激光点为基准的坐标系下。

如图 7 所示， $O_w$  表示世界坐标系， $O_r$  表示机器人坐标系， $O_l$  表示雷达坐标系。设  $p_i(x_i^w, y_i^w, \beta)$  为某一帧激光进行信息采集时第一束激光采集信息时机器人所在的位置，该位置所对应的激光雷达坐标系也作为基坐标系。在一帧点云扫描期间激光束扫描到障碍物时产生激光点  $^s p$  的距离和角度信息，此时机器人所在位置为  $p_{i+1}(x_{i+1}^w, y_{i+1}^w, \theta)$ ，此时的机器人坐标系用  $O_{r_2}$  表示，雷达坐标系用  $O_{l_2}$  表示， $O_{l_2}$  坐标系下的激光点  $^s p$  的数据信息存储一帧点云中，存储数据时参考的雷达坐标系为  $O_{l_1}$ ，点  $^s p$  的位置经过坐标的旋转和平移变换后相对于世界坐标系  $O_w$  的位置在  $^s p$  处。在已知  $p_{i+1}$  的情况下将  $^s p$  的位置转换到  $O_{l_1}$  坐标



**Figure 7.** Point cloud coordinate transformation  
**图 7.** 点云坐标变换

系下即可得到校正后的激光点，其中  $p_{i+1} = \omega_i$ 。设  ${}^{ri}_s t$  和  ${}^{ri}_s R$  分别表示从激光雷达坐标系到机器人坐标系的平移向量和旋转矩阵， ${}^{wi}_r t$  和  ${}^{wi}_r R$  分别表示从机器人坐标系到世界坐标系的平移向量和旋转向量。 ${}^r p$  和  ${}^w p$  分别表示激光点在机器人坐标系和世界坐标系下的位置，其计算如式(33) (34)。

$${}^r p = {}^{ri}_s R {}^s p + {}^{ri}_s t \tag{33}$$

$${}^w p = {}^{wi}_r R {}^r p + {}^{wi}_r t = {}^{wi}_r R ({}^{ri}_s R {}^s p + {}^{ri}_s t) + {}^{wi}_r t \tag{34}$$

将其转换为齐次形式如式(35) (36)

$${}^w \tilde{p} = {}^{wi}_r T {}^{ri}_s T {}^s \tilde{p} \tag{35}$$

$${}^{wi}_r T = \begin{pmatrix} {}^{wi}_r R & {}^{wi}_r t \\ \mathbf{0}^T & 1 \end{pmatrix} \quad {}^{ri}_s T = \begin{pmatrix} {}^{ri}_s R & {}^{ri}_s t \\ \mathbf{0}^T & 1 \end{pmatrix} \quad {}^s \tilde{p} = \begin{pmatrix} {}^s p \\ 1 \end{pmatrix} \tag{36}$$

再将激光点由世界坐标系转换到以  $O_i$  为原点的坐标系下用  ${}^s p_s$  表示，计算如式(37)，最终得到校正后的点云新信息。

$${}^s p_s = {}^{wi}_r T^{-1} ({}^w \tilde{p} - \tilde{p}_0) \tag{37}$$

## 4. 实验与分析

### 4.1. 实验说明

基于 ROS (Robot Operating System) 操作系统进行仿真实验，验证本文提出的算法。利用 Gazebo 平台构建 3D 仿真室内环境和仿真移动机器人如图 8 所示，整体为矩形长 40 m，宽 20 m，在室内和过道均布置有不同的障碍物构成结构化场景如图 8，世界坐标系原点在图中蓝色直线处，垂直向左为 x 轴正方向，垂直向下为 y 轴正方向。所用平台为 Ubuntu20.04，ROSkinetic，搭载处理器为 i7-6500U 运行内存为 4G 的计算机。仿真机器人底盘为四轮模型，底盘为左右两个主动轮和前后方各一个万向轮，搭载有单线激光雷达和里程计。轮式里程计频率为 100 Hz，测量误差为  $\pm 0.1\%$ ，激光雷达频率为 10 Hz，测量最远距离 8 m，最小 0.05 m，水平视角  $360^\circ$ ，角度分辨率为  $0.36^\circ$ ，测量误差  $\pm 0.02$  m。

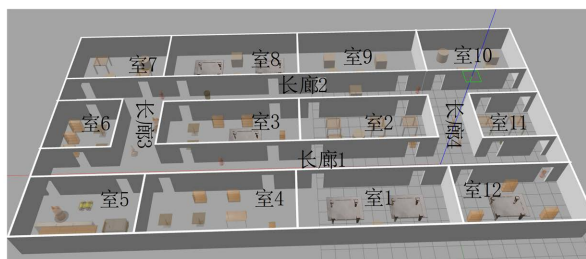


Figure 8. Experimental scenes

图 8. 实验场景

实验分为三个部分：1) 在室内结构化场景中和退化场景中，机器人处于变速运动的情况下，对比本文提出的改进的点云配准算法和其他点云配准算法在点云配准效率和定位效果上的差异。2) 讨论点云稀疏性对点云配准效率和定位效果的影响。3) 将本文提出的融合轮式里程计进行点云畸变矫正的方法移植到 Cartographer 算法中分别对比无点云畸变矫正、基于匀速预测模型进行点云畸变矫正的 Cartographer 算法在回环误差和场景还原度方面的差异。

1) 点云配准效率：以点云配准的平均耗时和平均迭代次数为准。

2) 点云配准定位误差：以激光雷达数据集停止录制时移动机器人在世界坐标系中的真实位置与点云配准定位位置的欧式距离作为定位误差评判标准。回环误差计算原理与定位误差计算方式相同，不同的是回环的起点与终点真值相同，均为世界坐标系坐标原点。

3) 场景还原度：场景还原度使用 SSIM (Structural Similarity Index Measure) 算法[21]进行计算，用 Python 对原始场景地图和 SLAM 算法生成的地图分别计算 SSIM 值，SSIM 的值越大说明算法对场景的还原度越高，点云畸变矫正效果越好。

## 4.2. 点云配准实验

### 4.2.1. 结构化场景中的点云配准实验

机器人从世界坐标系坐标原点出发进入室 2 后从另外一个门出，录制数据集共 482 帧点云信息，机器人最大移动速度为 1.5 m/s，最大加速度为 2 m/s<sup>2</sup>，最大角速度为 1 rad/s，整体运动状态为变速行驶。针对该数据集分别运行文本提出的改进的 PL-ICP 算法(在实验中记为 ours)，ICP 算法，PL-ICP 无初值提供的算法，PL-ICP 以匀速运动模型预测初值的算法(在本文中记为 PL-ICP+)，点云配准迭代的误差阈值均为 0.000001 m，经过参数调试最终  $\gamma$  值选取为 0.72，即滤波后的激光点数与原激光点数的平均比值。点云配准平均迭代次数的小梯图如图 9 所示，在同样没有提供初值的情况下，ICP 算法的平均迭代次数为 19 次，PL-ICP 算法的平均迭代次数为 33 次，这说明在点云配准的过程中相对于 ICP 算法 PL-ICP 算法对初值更加敏感，但从图 10 中可以看出 ICP 算法的平均配准耗时远远大于 PL-ICP 算法的平均耗时，ICP 算法的平均耗时为 0.1298 s，而一次点云扫描的时间 0.1 s，这最终会导致在点云配准的过程中出现掉帧的情况。PL-ICP 算法的平均耗时为 0.0341 s，这是因为 PL-ICP 算法本身是二次收敛的，所以尽管收敛到相同的误差精度 PL-ICP 算法需要更多的迭代次数但其耗时仍远小于 ICP 算法。PL-ICP+算法点云配准的平均耗时为 0.0263 s，平均迭代次数为 20 次，本文的算法点云平均配准耗时为 0.0209 s，平均迭代次数为 14 次。从图 9、图 10 中可以看出本文的算法具有更少的离群值且大多数值都集中在均值以下，这说明本文的算法对变化的场景具有更强的适应性和稳定性。离群点的出现是因为在结构化场景中机器人移动速度较快或短时间内转角较大的情况下环境特征变化明显，导致点云配准的过程中寻找对应点的效率降低，所以点云配准时间和迭代次数会增加。本文提出的改进的点云配准算法利用轮式里程计的测量



值为点云配准提供的初值比基于匀速预测模型提供的初值更加准确，且能适用于移动机器人变速运动的场景，同时利用自适应体素滤波法能够将点云稀疏化且能去除点云中的部分离群点，所以点云配准效率得到了显著提升。

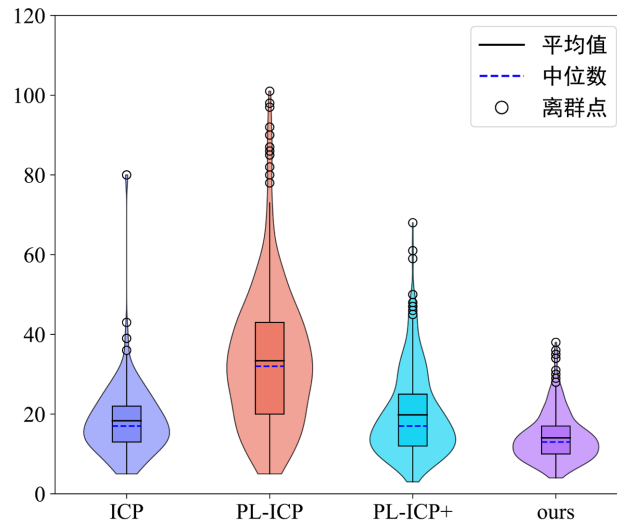


Figure 9. Number of point cloud alignment iterations  
图 9. 点云配准迭代次数

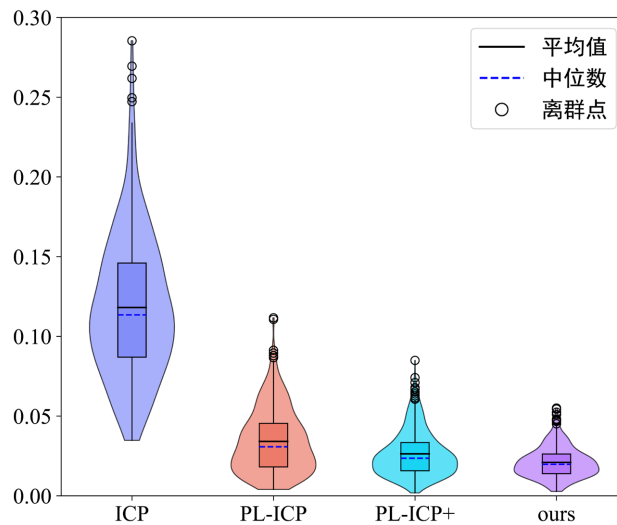


Figure 10. Point cloud alignment time consuming  
图 10. 点云配准耗时

在数据集停止录制时机器人真实位姿为(10.0355 m, -01739 m, 1.3997 rad)。点云配准定位位姿和误差如表 2 所示，其中 ICP 算法的定位误差最大为 0.8612 m 远大于 PL-ICP 算法的 0.4386 m，但其角度误差小于 PL-ICP 算法。相比于 PL-ICP 算法 PL-ICP+算法的定位误差和角度误差得到减小，但 x 轴误差却更大了。本文的算法无论在 x 轴和 Y 轴上的误差，还是定位误差和角度误差上均为最小，相比于 PL-ICP 算法和 PL-ICP+算法角度误差分别减少了 78.6%和 53.2%，定位误差分别减少了 64.5%和 51.3%。

**Table 2.** Point cloud alignment errors in structured scenes  
**表 2.** 在结构化场景中的点云配准误差

算法	点云配准位姿( $x, y, \theta$ )	x 轴绝对误差/m	y 轴绝对误差/m	角绝对误差/rad	定位误差/m
ICP	(10.8258, -0.5162, 1.2967)	0.7903	-0.3423	-0.1030	0.8612
PL-ICP	(10.1159, -0.6051, 1.2272)	0.0804	-0.4312	-0.1725	0.4386
PL-ICP+	(10.2959, -0.3596, 1.3207)	0.2604	-0.1857	-0.0790	0.3198
ours	(10.0533, -0.3285, 1.3627)	<b>0.0178</b>	<b>-0.1546</b>	<b>-0.0370</b>	<b>0.1556</b>

#### 4.2.2. 退化场景中的点云配准实验

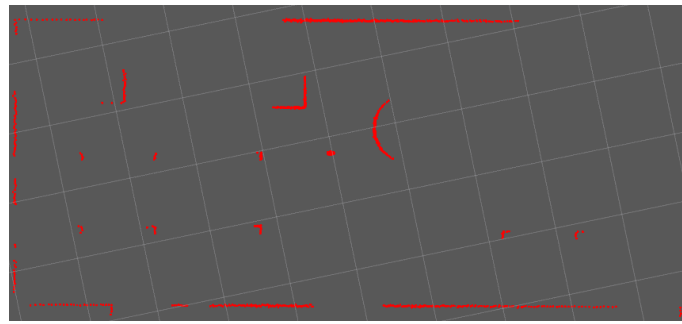
将长廊 1 中的障碍物去除，门设置为关闭状态，重新构造退化场景，录制数据集共 124 帧点云信息，在数据集停止录制时移动机器人真实位姿为(12.1455 m, 0.0625 m, 0.0053 rad)机器人最大行驶速度为 2 m/s，最大加速度 2 m/s<sup>2</sup>，最大角速度为 1 rad/s，整体运动状态为变速行驶。在退化场景激光雷达扫描得到的点云数据呈高度相似，所以各算法的点云配准迭代次数和耗时无明显差异，因此本小节仅讨论点云配准的定位误差。点云配准定位位姿和误差如表 3 所示，因为机器人是沿直线行驶所以 y 轴和角度的误差很小。主要参考 x 轴的误差，通过 ICP 算法和 PL-ICP 算法可以看出在无初值提供的情况下定位信息完全丢失。以匀速预测运动模型提供初值的 PL-ICP 算法的定位误差为 5.6576 m 远大于本文提出的算法的定位误差 0.7107 m，说明匀速预测运动模型并不适用于变速的场景，而本文的算法在退化的场景中和变速场景具有更好的鲁棒性。综上所述，本文提出的改进的 PL-ICP 算法在点云配准效率和定位效果方面均有所提升且适用于变速场景和退化场景。

**Table 3.** Point cloud alignment errors in degraded scenes  
**表 3.** 在退化场景中的点云配准误差

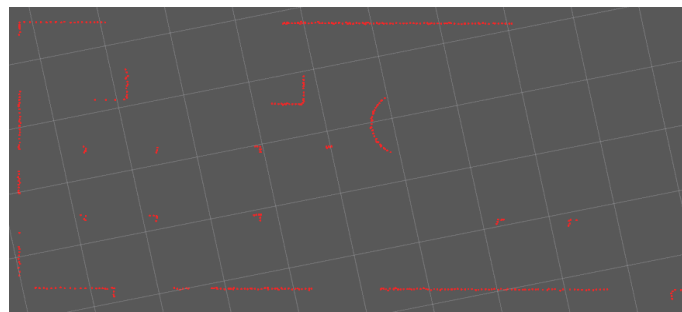
算法	点云配准位姿( $x, y, \theta$ )	x 轴绝对误差/m	y 轴绝对误差/m	角绝对误差/rad	定位误差/m
ICP	(0.2077, 0.0541, 0.0046)	-11.9378	-0.0084	-0.0007	11.9378
PL-ICP	(0.2298, 0.0644, 0.0051)	-11.9157	-0.0019	-0.0002	11.9157
PL-ICP+	(6.4879, 0.0743, 0.0078)	-5.6576	0.0059	0.0025	5.6576
ours	(11.4349, 0.0734, 0.0044)	-0.7106	0.0109	-0.0011	<b>0.7107</b>

#### 4.3. 讨论点云稀疏性对点云配准耗时和定位效果的影响

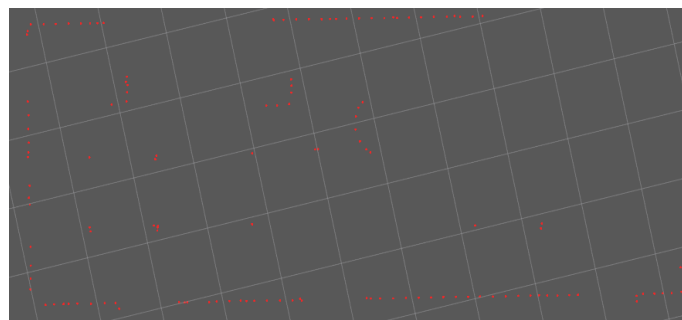
所用数据集和 4.2.1 所用数据集相同，验证  $\gamma$  (滤波前后一帧点云中激光点数的比值)的取值对点云配准的影响， $\gamma$  取值越小点云越稀疏。当  $\gamma = 1$  时点云为滤波前的原始点云，通过 rviz 可视化点云数据如图 11 所示，对应场景为图 8 中的室 2，通过图片可以看出在距离激光雷达较远的地方激光点比较密集出现大量重叠，而远处的点则相对稀疏。当  $\gamma = 0.60$  时无论距离远近点云的分布整体较为均匀，当  $\gamma = 0.21$  时点云已经非常稀疏只保留了部分几何特征。点云配准的效率和定位效果与点云的稀疏性有关，稀疏化的点云虽然能提升点云的配准速度，但点云过于稀疏会导致特征丢失且不利于寻找对应点，最终使得定位效果变差。如图 12 和图 13 所示，当  $\gamma$  值大于 0.72 时随着  $\gamma$  值的减小，点云配准耗时逐渐减少且定位误差也在不变小，但是当  $\gamma$  值在小于 0.72 以后点云配准的定位误差会逐渐增大。因此在实际应用中需要权衡实时性与定位精度选择合适的  $\gamma$  值。



(a)  $\gamma = 1$



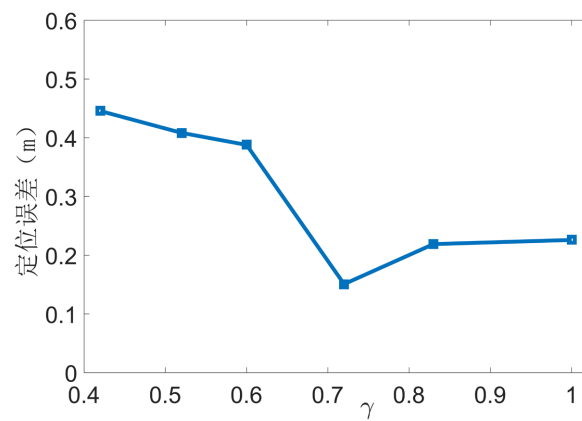
(b)  $\gamma = 0.61$



(c)  $\gamma = 0.2$

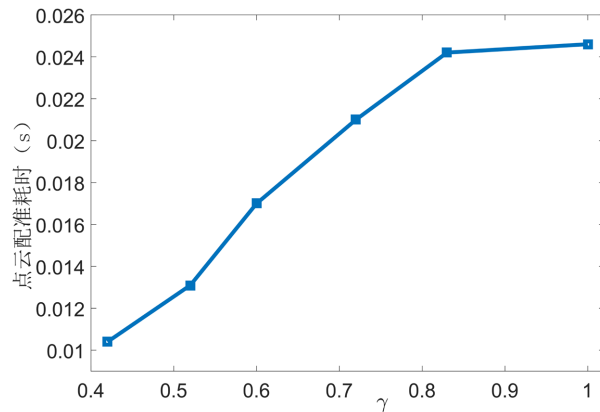
**Figure 11.** Point cloud plot with different  $\gamma$  values

**图 11.** 不同  $\gamma$  值的点云图



**Figure 12.** Positioning error for different  $\gamma$  values

**图 12.** 不同  $\gamma$  值的点云配准定位误差

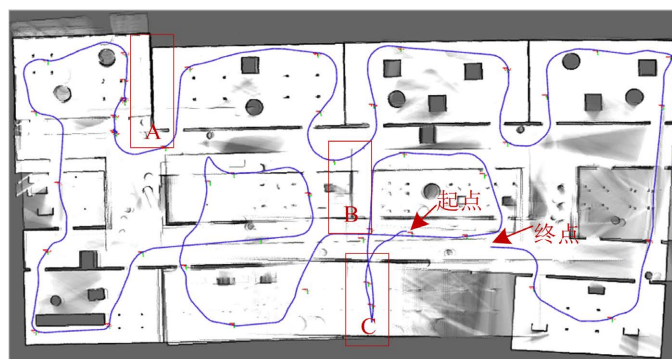


**Figure 13.** Scan match error for different  $\gamma$  values  
**图 13.** 不同  $\gamma$  值的点云配准耗时

#### 4.4. 点云畸变矫正实验

移动机器人从世界坐标系坐标原点出发遍历整个室内环境最后又回到起始位置，录制数据集共 2223 帧激光点云信息，机器人最大移动速度为 1.5 m/s，最大加速度为 2 m/s<sup>2</sup>，最大角速度为 1 rad/s，整体运动状态为变速行驶。分别使用无点云畸变矫正的 Cartographer 算法、利用匀速模型进行点云畸变矫正的 Cartographer 算法、利用本文提出的融合轮式里程计进行点云畸变矫正的 Cartographer 算法分别运行该数据集。构建的二维地图如图 14~图 16 所示，图中的蓝色线条为机器人的运动轨迹，轨迹上的点为机器人的位姿节点。

无点云畸变矫正时 Cartographer 算法构建的地图如图 14，场景 A 对应的真实场景为图 8 中对应的室 7 和室 8 交界处，场景 B 对应的是室 2 和室 3 交界处，场景 C 对应的是室 1 和室 4 交界处。在真实场景所有墙面都是互相垂直的，因为点云畸变的存在图 14 的场景 A 中的墙面无法完全对齐，场景 B 中的墙面发生了严重倾斜，场景 C 中室 1 和室 4 之间的墙面已近完全丢失，这也就导致了机器人的轨迹出现了严重的飘移现象，对比图 15 和图 16 中的轨迹可以看出图 14 的场景 C 附近这部分的定位信息几乎完全丢失。基于匀速预测模型进行点云畸变矫正的 Cartographer 算法构建的地图如图 15 所示，可以看出同样的场景有了点云畸变矫正以后，建图效果得到了改善，但构建的地图与实际场景依然有较大的出入，图 15 A 中的墙面并未完全对齐，图 15 B 两墙面间距较大，图 15 C 中两墙面之间有大量的重影，这是因为当移动机器人处于变速运动时，匀速预测模型并不能准确的估计机器人的运动状态，所以点云畸变矫正的



**Figure 14.** Cartographer (no point cloud distortion correction)  
**图 14.** Cartographer (无点云畸变矫正)

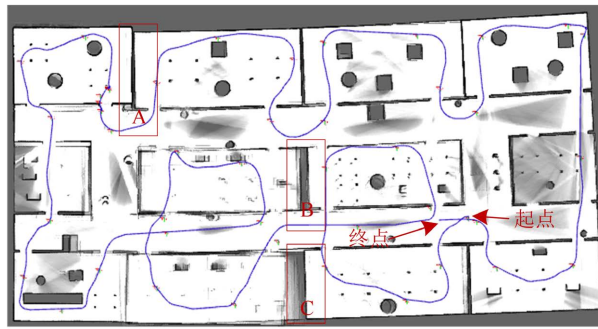


Figure 15. Cartographer (point cloud distortion correction based on uniform prediction model)

图 15. Cartographer (基于匀速预测模型的点云畸变矫正)

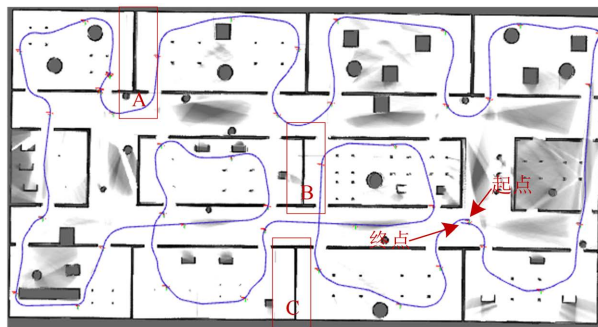


Figure 16. Cartographer (point cloud distortion correction for fusion wheel odometry)

图 16. Cartographer (融合轮式里程计的点云畸变矫正)

效果并不理想，这就导致构建的地图无法还原真实场景。对比图 14~图 16 可以看出融合轮式里程计进行点云畸变矫正的 Cartographer 算法构建的地图整体无明显畸变。

回环误差效果如图 14~图 16 所示。终点的真值与起点相同，从图 14 中可以看出实际定位位置和真值相差较远，这说明在没有点云畸变矫正的情况下 Cartographer 算法回环失败。从图 15 和图 16 可以看出点云畸变矫正后的 Cartographer 算法回环效果更好，从表 4 可以看出，融合轮式里程计进行点云畸变矫正的 Cartographer 算法的回环效果最好，误差在 0.5 m 以内，比基于匀速预测模型进行点云畸变矫正的 Cartographer 算法回环误差减小了 81%。从图 14~图 16 中可以看出除了重点圈出来的部分外，在长廊 1，长廊 3，长廊 4，室 2 内也存在不用程度的地图偏移，特别在图 14 中长廊 1 发生了明显变形，室 2 内物体的特征也有部分丢失，图 15、图 16 中虽然有所改变但也存在地图无法还原真实场景的情况。通过 SSIM 算法对比 Cartographer 算法结合不同点云畸变矫正方法后所建地图对真实场景的还原度如表 5 所示，从表 5 中可以看出文本提出的融合轮式里程计进行点云畸变矫正的方法对真实场景的还原度最高。

Table 4. Loopback error

表 4. 回环误差

算法	起点(x, y)	终点(x, y)	回环误差
Cartographer (无点云畸变矫正)	(0, 0)	(-5.6401, 1.9980)	5.9835
Cartographer (基于匀速预测模型的点云畸变矫正)	(0, 0)	(2.0273, 0.0918)	2.0294
Cartographer (融合轮式里程计的点云畸变矫正)	(0, 0)	(0.3429, 0.1767)	<b>0.3858</b>



**Table 5.** Scene restoration degree  
**表 5.** 场景还原度

算法	Cartographer (无点云畸变矫正)	Cartographer (基于匀速预测 模型的点云畸变矫正)	Cartographer (融合轮式里程计的 点云畸变矫正)
场景还原度	0.577	0.624	<b>0.775</b>

## 5. 结论

本文对退化场景和变速场景中点云配准效率低, 定位效果差, 点云易发生畸变的原因进行了分析, 将轮式里程计和激光雷达相融合用于改进上诉问题。将轮式里程计测量的移动机器人相对位姿变化信息作为 PL-ICP 算法配准的初值, 同时利用自适应体素滤波法对原始点云进行下采样, 在提升点云配准定位精度的同时降低计算量。实验结果表明本文提出的方法相比基于匀速预测模型的 PL-ICP 算法, 在移动机器人变速运动时, 在结构化场景中点云配准平均迭代次数减少了 30%、平均耗时减少 20.5%、定位误差减少了 51.3%, 在退化场景中的定位误差减少了 87.4%。针对点云畸变问题本文提出基于 EKF 融合轮式里程计和点云配准插值的方法进行点云畸变矫正。实验结果表明融合轮式里程计进行点云畸变矫正的 Cartographer 算法在变速场景中回环误差减少了 81%, 对场景的还原度提升了 24.2%, 本文提出的点云畸变矫正方案除了移植到 Cartographer 算法中同样可以应用到其他算法中。

## 参考文献

- [1] 吴建清, 宋修广. 同步定位与建图技术发展综述[J]. 山东大学学报(工学版), 2021, 51(5): 16-31.
- [2] 李延真, 石立国, 徐志根, 程超, 夏清泉. 移动机器人视觉 SLAM 研究综述[J]. 智能计算机与应用, 2022, 12(7): 40-45.
- [3] Macario Barros, A., Michel, M., Moline, Y., *et al.* (2022) A Comprehensive Survey of Visual Slam Algorithms. *Robotics*, **11**, Article No. 24. <https://doi.org/10.3390/robotics11010024>
- [4] Huang, L. (2021) Review on LiDAR-Based SLAM Techniques. 2021 *International Conference on Signal Processing and Machine Learning (CONF-SPML) IEEE*, Stanford, 14 November 2021, 163-168. <https://doi.org/10.1109/CONF-SPML54095.2021.00040>
- [5] Tee, Y.K. and Han, Y.C. (2021) Lidar-Based 2D SLAM for Mobile Robot in an Indoor Environment: A Review. 2021 *International Conference on Green Energy, Computing and Sustainable Technology (GECOST) IEEE*, Miri, 7-9 July 2021, 1-7. <https://doi.org/10.1109/GECOST52368.2021.9538731>
- [6] Yeong, D.J., Velasco-Hernandez, G., Barry, J., *et al.* (2021) Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. *Sensors*, **21**, Article No. 2140. <https://doi.org/10.3390/s21062140>
- [7] Qu, Y., Yang, M., Zhang, J., *et al.* (2021) An Outline of Multi-Sensor Fusion Methods for Mobile Agents Indoor Navigation. *Sensors*, **21**, Article No. 1605. <https://doi.org/10.3390/s21051605>
- [8] Chen, S., Zhou, B., Jiang, C., *et al.* (2021) A LiDAR/Visual SLAM Backend with Loop Closure Detection and Graph Optimization. *Remote Sensing*, **13**, Article No. 2720. <https://doi.org/10.3390/rs13142720>
- [9] 章弘凯, 陈年生, 代作晓, 范光宇. 一种多层次数据融合的 SLAM 定位算法[J]. 机器人, 2021, 43(6): 641-652.
- [10] Hess, W., Kohler, D., Rapp, H., *et al.* (2016) Real-Time Loop Closure in 2D LIDAR SLAM. 2016 *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, 16-21 May 2016, 1271-1278. <https://doi.org/10.1109/ICRA.2016.7487258>
- [11] Roh, M., Na, K.-I., Seo, B.-S. and Byun, J. (2015) Drivable Road Detection with 3d Point Clouds Based on the MRF for Intelligent Vehicle. In: Mejias, L., Corke, P. and Roberts, J., Eds., *Field and Service Robotics*, Springer, Berlin, 49-60. [https://doi.org/10.1007/978-3-319-07488-7\\_4](https://doi.org/10.1007/978-3-319-07488-7_4)
- [12] Bai, W., Li, G. and Han, L. (2017) Correction Algorithm of LIDAR Data for Mobile Robots. *International Conference on Intelligent Robotics and Applications*, Wuhan, 16-18 August 2017, 101-110. [https://doi.org/10.1007/978-3-319-65292-4\\_10](https://doi.org/10.1007/978-3-319-65292-4_10)
- [13] Moosmann, F. and Stiller, C. (2011) Velodyne Slam. *Intelligent Vehicles Symposium*, **4**, 393-398. <https://doi.org/10.1109/IVS.2011.5940396>

- 
- [14] Zhang, J. and Singh, S. (2014) LOAM: Lidar Odometry and Mapping in Real-Time. *Robotics: Science and Systems*, **2**, 1-9. <https://doi.org/10.15607/RSS.2014.X.007>
  - [15] Yang, W., Gong, Z., Huang, B., *et al.* (2022) Lidar with Velocity: Correcting Moving Objects Point Cloud Distortion from Oscillating Scanning Lidars by Fusion with Camera. *IEEE Robotics and Automation Letters*, **7**, 8241-8248. <https://doi.org/10.1109/LRA.2022.3187506>
  - [16] Zhang, B., Zhang, X., Wei, B., *et al.* (2019) A Point Cloud Distortion Removing and Mapping Algorithm Based on Lidar and IMU UKF Fusion. 2019 *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Hong Kong, 8-12 July 2019, 966-971. <https://doi.org/10.1109/AIM.2019.8868647>
  - [17] Rusinkiewicz, S. and Levoy, M. (2001) Efficient Variants of the ICP Algorithm. *IEEE Proceedings 3rd International Conference on 3-D Digital Imaging and Modeling*, Quebec City, 28 May-1 June 2001, 145-152.
  - [18] Censi, A. (2008) An ICP Variant Using a Point-to-Line Metric. 2008 *IEEE International Conference on Robotics and Automation*, Pasadena, 19-23 May 2008, 19-25. <https://doi.org/10.1109/ROBOT.2008.4543181>
  - [19] Kalman, R.E. (1960) A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, **82**, 35-45. <https://doi.org/10.1115/1.3662552>
  - [20] Smith, R.C. and Cheeseman, P. (1986) On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*, **5**, 56-68. <https://doi.org/10.1177/027836498600500404>
  - [21] Wang, Z., Bovik, A.C., Sheikh, H.R., *et al.* (2004) Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, **13**, 600-612. <https://doi.org/10.1109/TIP.2003.819861>