ISSN 2616-5775 Vol. 6, Issue 8: 112-118, DOI: 10.25236/AJCIS.2023.060814

Improved algorithm of cartographer based on laser odometer

Bowen Du[1,*], Xiaochang Ni[1], Yaqing Wang[1], Jie Zhou[1], Jing Li[1]

[1]School of Electronic Engineering, Tianjin University of Technology and Education, Tianjin, China
[*]Corresponding author

Abstract: *In the front-end matching process of the Cartographer algorithm, the accuracy of matching between the point cloud and submap relies on the initial values provided by the pose fusion algorithm. However, the original algorithm's pose fusion algorithm has low accuracy. To address this issue, this paper proposes an improved Cartographer algorithm based on a laser odometer. The improved algorithm utilizes NDT registration to obtain the pose transformation between frames. Additionally, a pre-integration of the IMU between the front and back frames is performed for joint optimization, allowing for the acquisition of a more accurate pose. This enhanced accuracy contributes to improving the matching of high point clouds with the submap. To analyze the efficacy of the improved algorithm, comparisons were made with the original Cartographer algorithm by analyzing the map construction effect and conducting positioning accuracy tests using datasets. The experiments confirmed that the improved algorithm is both feasible and effective in enhancing the map construction effect and pose accuracy.*

Keywords: *Simultaneous Localization and Mapping, LIDAR, Multiple Sensor Fusion, Cartographer, Front-end matching, Driverless cars*

1. Introduction

SLAM (Simultaneous Localization and Mapping) was first introduced by Cheeseman in 1986 at the IEEE Conference on Robotics and Automation. This technology enables a robot to achieve localization and pose determination in an unknown environment using sensor data, and construct a map based on environmental data corresponding to its poses. There are two approaches to implementing LiDAR-based SLAM. The first approach is Bayesian filter-based LiDAR SLAM, which uses motion and observation equations to predict and update the state of observed data. For instance, Gmapping utilizes the Rao-Blackwellized Particle Filter (RBPF) algorithm to separate the localization process from the mapping process, resulting in accurate map construction in small-scale environments. However, the accuracy of map construction diminishes in large-scale environments due to the accumulation of instability over time, as the filter algorithm relies on estimates from previous time steps[1-3].

The second approach is graph-based LiDAR SLAM, which involves building and optimizing maps. The map is constructed using poses as nodes and constraints between nodes as edges. Map optimization aims to refine these constraints. Hector SLAM, for example, solves pose increments by establishing constraints between two adjacent frames and maps the poses to complete map construction.

Cartographer[4] is a graph-based LiDAR SLAM algorithm. The mapping process of Cartographer can be divided into two parts: front-end map construction and back-end loop detection and optimization. The implementation of Cartographer is illustrated in Figure 1 below.

ISSN 2616-5775 Vol. 6, Issue 8: 112-118, DOI: 10.25236/AJCIS.2023.060814

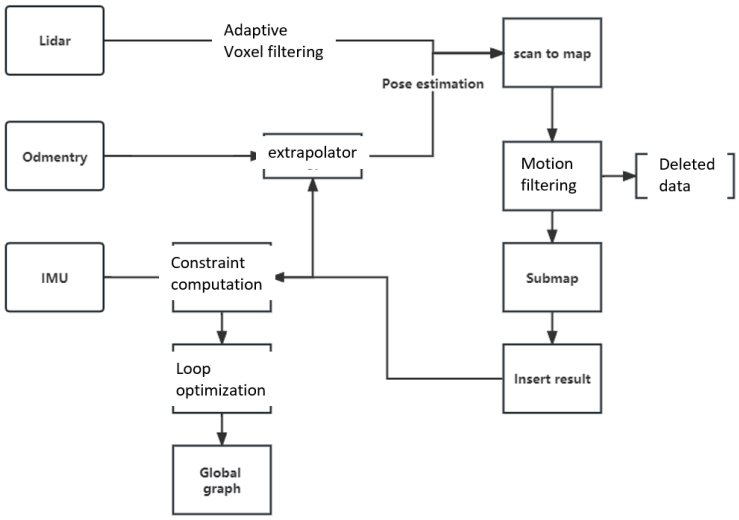


Figure 1: Flowchart of the cartographer algorithm implementation

The radar data in Cartographer is processed in the front-end using the Adaptive Speed-up Filter for scanning and matching. The pose prediction is provided by the pose extrapolator, which combines odometry data and inertial measurement unit data. In the absence of sensor data other than radar, a uniform motion model is used to infer the pose. Once the scanned and matched data is obtained, motion filtering is applied to refine the results, including obtaining the time stamp and pose information. Afterward, constraint calculations are performed, and the sparse map is efficiently optimized by incorporating global measurement data. Finally, the maps from different scans are stitched together to complete the entire SLAM process[5].

In the front-end module of Cartographer, the pose extrapolator is responsible for predictive pose matching between the point cloud and submap. However, the accuracy of the pose prediction using the uniform motion model is often insufficient for practical applications. The original version of Cartographer used the untraced Kalman algorithm to fuse radar data, IMU, and odometry data. Unfortunately, in real-world scenarios, the algorithm did not effectively align the timestamps of the multi-sensor data. Another approach, described in literature[6], involves caching sensor data in a double-ended queue and representing pose transformations using a tracker. This method enables fusion of radar matching data, IMU, and wheeled odometry data but requires significant computational resources for pose calculations. In another study[7-8], the pose fusion algorithm in the original algorithm was improved by incorporating velocity pre-integration, which helped improve pose accuracy. However, this method still relied on wheeled odometry data during the pose calculation process. To address the issue of low accuracy in the front-end pose algorithm of the Cartographer algorithm, this paper proposes a front-end map construction method based on laser odometry. The method utilizes NDT registration to determine the relative positions' pose within a specific time window. Additionally, IMU pre-integration is employed to correct radar motion distortion. Both NDT registration and IMU pre-integration are jointly optimized to obtain a precise initial pose value. This improvement aims to enhance the matching quality between frames and submaps.

2. Relate works

The current pose of a frame in the global coordinate system can be determined using laser rangefinder and wheel odometry data. However, laser data alone is sparse and vulnerable to motion interference, leading to reduced accuracy in laser odometry under degraded conditions. To improve the precision of laser odometry, data from an Inertial Measurement Unit (IMU) can be fused with laser odometry.

There are currently two methods for this fusion. The first method is loosely coupled, where the IMU and laser odometry are estimated separately. While this method may result in some data loss, it has a lower computational load. For instance, in the LOAM[9], high-frequency low-precision scan-scan matching is used for localization, while low-frequency high-precision scan-map matching is employed for map construction. In this approach, the odometry utilizes IMU data as a prior for orientation and estimates pose transformation with a constant velocity motion model, without directly fusing laser and

ISSN 2616-5775 Vol. 6, Issue 8: 112-118, DOI: 10.25236/AJCIS.2023.060814

IMU data. By using high-frequency localization and low-frequency mapping, the computational load is reduced while maintaining mapping accuracy.

Another method is known as the tightly coupled fusion method, which involves predicting the state using IMU data and then correcting the predictions using measured values. This fusion method is commonly referred to as Lidar Inertial Odometry (LIO). LIO_mapping[10] was the first to propose the tightly coupled method of fusing IMU data and laser odometry, optimizing both the IMU and Lidar data jointly. The main advantage of this method is that it can maintain good accuracy even in situations where the Lidar data is degraded.

In other implementations such as LIOM[11], laser inertial odometry fuses laser rangefinder data and IMU data using an error state Kalman filter (ESKF), resulting in low drift and robust pose estimation.MC2SLAM[12],on the other hand, optimizes point cloud distortion compensation and point cloud matching, improving accuracy by utilizing IMU preintegration for pose estimation. LIO-SAM[13] takes a unique approach by using factor graph optimization to treat laser odometry as one module in SLAM (Simultaneous Localization and Mapping) for pose estimation and mapping. By tightly coupling multiple sensors, the motion estimation results become more robust. The joint optimization of multiple sensors in a tightly coupled process improves mapping accuracy, making the system more resilient as it does not rely solely on a single sensor. Taking inspiration from references [13], this paper utilizes laser odometry to provide high-precision initial values and constructs a reliable local map to optimize the cartographer system.

3. Implementation and Mapping of Laser Odometry

3.1. Localized Submap-based Algorithm Framework

This paper focuses on optimizing the front-end of the algorithm. It achieves this by tightly coupling radar data with IMU data to provide an initial pose estimation. Subsequently, the point cloud is matched with a submap, resulting in the generation of a localized map. The implementation process is illustrated in Figure 2.

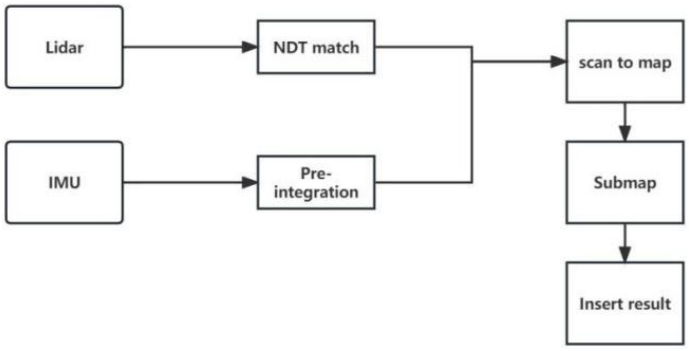


Figure 2: Local Submap Algorithm Framework

3.2. Implementation of Laser Odometry

In reference [13], a feature-based method is utilized for constructing maps, which involves the extraction and matching of lidar data features. However, this approach leads to lower efficiency. In contrast, this paper proposes a direct matching method for map construction that utilizes lidar data directly. This approach improves the efficiency of map construction by eliminating the need for feature matching between point clouds and overcoming associated problems. The paper achieves this by obtaining the relative pose of adjacent frames through NDT matching. This eliminates the necessity of feature matching between point clouds and mitigates challenges related to such matching. The local properties of point clouds are described using the normal distribution probability density function, enabling accurate and rapid computation since all derivatives can be calculated analytically. During the NDT matching process, adjacent point clouds are divided into two grids in order to facilitate efficient matching.Grid $X = \{x_1, x_2, ..., x_N\}$,grid $Y = \{y_1, y_2, ..., y_N\}$ .The matching process can be defined as follows.

ISSN 2616-5775 Vol. 6, Issue 8: 112-118, DOI: 10.25236/AJCIS.2023.060814

$$\mu = \frac{1}{N_i}\sum_{i=1}^{N_i} \mathrm{x}_i \tag{1}$$

$$\sigma = \frac{1}{N_i - 1}\sum_{i=1}^{N_i} (\mathrm{x}_i - \mu)(\mathrm{x}_i - \mu)^{\mathrm{T}} \tag{2}$$

N_i is the representation of the number of point clouds in a grid representation of the number of point clouds in a grid, μ represent of mean of grid X . σ is the representation of grid covariance. Of grid X , The Gaussian distribution modeling of point clouds can be achieved using formulas (1) and (2). The matching between two grids can be represented using the following formula:

$$y_i' = T(p,y) = Ry_i + t \tag{3}$$

y_i' is representation to project point clouds y onto point clouds x , p is a pose transformation relationship involved in the projection process. R is representation of the rotation relationship , t represents the translation relationship.

Frequency of radar data is lower than the frequency of IMU data. Therefore, between two consecutive radar data frames, there are multiple IMU data points. IMU preintegration is performed between adjacent radar data frames to correct the relative motion between point clouds. The motion model of the IMU can be represented as:

$$\begin{aligned} \widehat{a}_t &= a_t + b_{at} + n_a \\ \widehat{\omega}_t &= \omega_t + b_{\omega t} + n_w \end{aligned} \tag{4}$$

$\widehat{\omega}_{\mathrm{t}}$ represents the estimated value of the gyroscope. $\widehat{a}_{\mathrm{t}}$ represents the estimated value of the acceleration. $b_{\omega t}$ represents the actual measurement of the gyroscope. b_{at} represents the bias of the acceleration. n_a represents the noise of the acceleration, n_ω represents the noise of the gyroscope. Both the accelerometer noise and gyroscope noise follow Gaussian distribution. The accuracy of the IMU will be influenced by these factors, and they will be taken into account in the calculation process. The pose increment between two frames can be defined as:

$$\begin{aligned} \Delta R_{ij} &= R_i^T R_j \\ \Delta v_{ij} &= R_i^T (v_j - v_i - g\Delta t_{ij}) \\ \Delta p_{ij} &= R_i^T (p_j - p_i - v_i \Delta t_{ij} - \frac{1}{2} g\Delta t_{ij}^2) \end{aligned} \tag{5}$$

ΔR_{ij} represents the rotational change between the frames. Δv_{ij} represents the change in velocity between the frames. Δp_{ij} represents the change in position between the frames. g represents the gravitational constant. In reference [14], the transformation between visual frames is utilized to rectify the biases in the IMU, thereby achieving data synchronization. Building upon this approach, the correction of IMU biases can be defined based on the pose estimated through NDT matching.This process can be represented by the following equation:

$$\delta b_\omega^* = \arg\min_{\delta b_\omega} \sum_k \left\| \widehat{q}_{l0}^{b_{k+1}} \otimes \widehat{q}_{b_k}^{l0} \otimes \Delta \widehat{R}_{b_{k+1}}^{b_k} \right\|^2 \tag{6}$$

δb_ω^* is the minimum value of IMU bias, $\Delta \widehat{R}_{b_{k+1}}^{b_k}$ represents the that represents the rotational relationship between adjacent frames, $\otimes$ represents quaternions multiplication, $\widehat{q}_{l0}^{b_{k+1}}$ and $\widehat{q}_{b_k}^{l0}$

ISSN 2616-5775 Vol. 6, Issue 8: 112-118, DOI: 10.25236/AJCIS.2023.060814

represents the transformation of pose between NDT matches. By solving for the minimum IMU bias, it allows for the coupling between radar data and IMU to be achieved.

3.3. Construction of local maps

The goal of the front-end is to simultaneously construct high-precision local submaps, which can be achieved through the matching between point clouds and submaps. The error of the matching between point clouds and submaps can be represented as.

$$\delta\hat{\xi}^{*} = \arg\min_{\delta\xi} \sum_{i} \left[1 - M(T_i(\hat{\xi}))\right]^2 \tag{7}$$

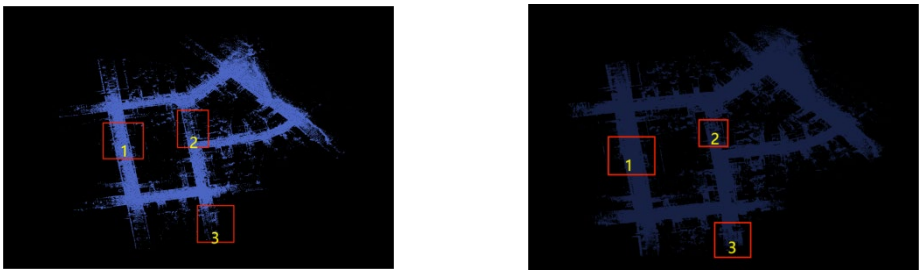
$M(\bullet)$ is the probability of matching with a voxel map, $T_i(\bullet)$ represents the pose transformation between the point cloud and the submap. The optimal matching effect can be achieved by finding the minimum value of $\hat{\xi}$

3.4. Loop detection

In local SLAM, the process of estimating poses often introduces errors due to factors such as map resolution and sensor noise. This is because pose estimation relies on the accumulation of translations and rotations, and point clouds are only matched with a subset of the map that includes the most recent scans. As the distance traveled increases, drift errors become more pronounced. To address this, Cartographer implements a loop detection method based on branch and bound. A loop is considered detected when the calculated pose of a point cloud yields a matching score above a predefined threshold. The relative pose associated with the loop is then added as a constraint to the nonlinear least squares

4. Experiment

The experiment utilized the KITTI dataset, which consists of data collected by a data collection platform equipped with two grayscale cameras, two color cameras, one Velodyne 64-line 3D LiDAR, four optical lenses, and one GPS navigation system. The experimental testing platform was based on the Ubuntu 16.04 operating system with Robot Operating System (ROS) installed.



a *b*

Figure 3: Illustrates a comparison of the mapping results between the improved algorithm and the original algorithm on the KITTI dataset. In this figure, (a) represents the point cloud map constructed by the improved algorithm, while (b) represents the map constructed by Cartographer.

Upon comparing Figure 3 (a) and Figure 3(b), it can be noted that in sections 1 and 2, the original algorithm exhibits a tilted road, while this problem has been addressed in the improved algorithm. Furthermore, in section 3, the map constructed by the original algorithm shows overlapping, however, this issue has been resolved in the improved algorithm.

The trajectory generated using Evo evaluation is presented in Figure 4. By examining the generated trajectory map, it becomes apparent that the improved algorithm produces trajectories that are more consistent with the ground truth when compared to the original algorithm. Additionally, the generated poses are more accurate as well.

ISSN 2616-5775 Vol. 6, Issue 8: 112-118, DOI: 10.25236/AJCIS.2023.060814

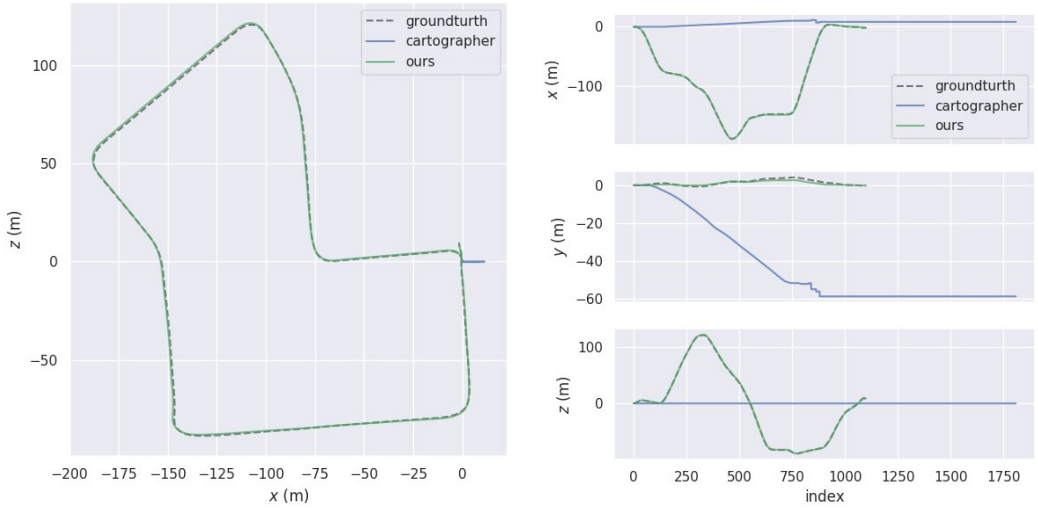


Figure 4: Trajectory map evaluated using Evo

5. Conclusions

This article aims to enhance the pose generation algorithm of the Cartographer algorithm by incorporating laser odometry, thereby improving the overall effectiveness and accuracy of the mapping process. The experimental results showcase the success of the optimized algorithm in addressing the issue of low accuracy in pose calculation, as well as rectifying motion distortion and map drift problems encountered in the original algorithm. This research introduces a promising new avenue for optimizing pose estimation within the Cartographer framework.

References

[1] Smith R, Self M, Cheeseman P. Estimating Uncertain Spatial Relationships in Robotics[J]. Machine Intelligence & Pattern Recognition, 1988, 5(5):435-461.
[2] Giorgio G, Cyrill S, Wolffram B. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters [C]//IEEE Transactions on Robotics, 2007:34-46.
[3] Kohlbrecher S, Stryk O V, Meyer J, et al. A flexible and scalable slam system with full 3d motion estimation [C]//2011 IEEE International Symposium on Safety, Security, and Rescue Robotics IEEE, 2011:155-160.
[4] Hess W, Kohler D, Rapp H, et al. Real-Time Loop Closure in 2D LIDAR SLAM[C]//2016 IEEE International Conference on Robotics and Automation (ICRA)IEEE, 2016:1271-1278.
[5] Julier S J, Uhlman J K. New extension of the Kalman filter to nonlinear systems [J]. Signal Processing, Sensor Fusion, and Target Recognition VI, 1997, 3068.
[6] Liang Z, Zhiyu L, Jingying C et al. Cartographer Algorithm and System Implementation of enhanced pose fusion for sweeping robot [J]. Journal of Software, 2020, 31(09):2678-2690.
[7] Xin S, Huasong M. Improved Cartographer algorithm based on velocity integral pose fusion [J]. Applied Laser, 2021, 41(05):1063-1069.
[8] Forster C, Carlone L, Dellaert F, et al. On-Manifold Preintegration for Real-Time Visual--Inertial Odometry [J]. IEEE transactions on robotics, 2017, 33(1):1-21.
[9] Zhang J, Sanjiv S. Loam: Lidar Odometry and Mapping in Real-time. Proceedings of Robotics [C]//Science and Systems Conference, 2014:109-111.
[10] H. Ye, Y. Chen, M. Liu. Tightly Coupled 3D LiDAR Inertial Odometry and Mapping[C]//IEEE International Conference on Robotics and Automation, 2019:3144-3150.
[11] Zhao S, Fang Z, Li H, Scherer S. A Robust Laser-Inertial Odometry and Mapping Method for Large-Scale Highway Environments [C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019:1285-1292.
[12] Neuhaus F, Ko T, Kohnen R, et al. MC2SLAM: Real-Time Inertial Lidar Odometry Using Two-Scan Motion Compensation[C]// German Conference on Pattern Recognition. Springer, Cham, 2018:60–72.
[13] Shan T, Englot B, Meyers D, et al. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via

ISSN 2616-5775 Vol. 6, Issue 8: 112-118, DOI: 10.25236/AJCIS.2023.060814

Smoothing and Mapping [C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020:5135-5142.
[14] Liguang J, Qiguang L I. Research on the calibration algorithm of disk odometer based on laser radar[J].Journal of Beijing Information Science & Technology University, 2019.