

ROLO-SLAM: Rotation-Optimized LiDAR-Only SLAM in Uneven Terrain with Ground Vehicle

Yinchuan Wang¹ | Bin Ren¹ | Xiang Zhang¹ | Pengyu Wang² | Chaoqun Wang¹ | Rui Song¹ | Yibin Li¹ | Max Q.-H. Meng²

¹School of Control Science and Engineering,
Shandong University, Jinan, China

²Department of Electronic and Electrical
Engineering, Southern University of Science and
Technology, Shenzhen, China

Correspondence

Chaoqun Wang, School of Control Science and
Engineering, Shandong University, Jinan 250061,
China.

Email: chaoqunwang@sdu.edu.cn

Funding Information

This research was supported by the Text

Abstract

LiDAR-based SLAM is recognized as one effective method to offer localization guidance in rough environments. However, off-the-shelf LiDAR-based SLAM methods suffer from significant pose estimation drifts, particularly components relevant to the vertical direction, when passing to uneven terrains. This deficiency typically leads to a conspicuously distorted global map. In this article, a LiDAR-based SLAM method is presented to improve the accuracy of pose estimations for ground vehicles in rough terrains, which is termed Rotation-Optimized LiDAR-Only (ROLO) SLAM. The method exploits a state front propagation to separately and accurately determine the location and orientation at the front-end. Furthermore, we adopt a parallel-capable spatial voxel map to achieve correspondence-matching, and a spherical alignment-guided rotation registration within each voxel for precise attitude estimation. Incorporating geometric alignment, we introduce the continuous-trajectory constraint into the optimization formulation to enhance the rapid and effective estimation of LiDAR's translation. Subsequently, we extract several keyframes to construct the submap and exploit an alignment from the current scan to the submap (scan-to-submap) for precise pose estimation. Meanwhile, a global-scale factor graph is established to aid in the reduction of cumulative errors. In various scenes, diverse experiments have been conducted to evaluate our method. The results demonstrate that ROLO-SLAM excels in pose estimation of ground vehicles and outperforms existing state-of-the-art LiDAR SLAM frameworks.

KEY WORDS

LiDAR odometry, scan matching, simultaneous localization and mapping (SLAM), uneven terrain, ground vehicle.

1 | INTRODUCTION

Localization is of utmost significance in the context of autonomous driving. It serves as the fundamental building block for safe and efficient navigation, enabling vehicles to precisely determine their position within their environment. For navigation in uneven terrain, the movement of the vehicle is inevitably subject to fluctuations as it negotiates uneven terrain. The sensors rigidly mounted on the ground vehicle are particularly vulnerable to these movements, rendering off-terrain localization a particularly intricate and challenging endeavor.

Simultaneous localization and mapping (SLAM) technique allows one to localize ego-pose of sensors and meantime offer an environmental map, offering . Vision-based methods generally exploit monocular or stereo cameras as the sensor and match successive image features to estimate the camera's ego-motion. Nevertheless, the sensitivity to illumination and sensing range constrains the robustness of the vision-based method. While LiDAR-based methods typically utilize consecutive LiDAR scans for point cloud registration in the local and global scale, allowing for the estimation of precise sensor ego-motion. These methods enjoy the benefits of insensitivity to environmental conditions, long sensing range and low measurement noise, particularly in outdoor scenes. Thus, LiDAR-based methods have increased in popularity among SLAM researches.

Abbreviations: ANA, anti-nuclear antibodies; APC, antigen-presenting cells; IRF, interferon regulatory factor.

The off-the-shelf LiDAR-based SLAM frameworks generally use the geometric feature of point clouds to restore LiDAR's locomotion. They typically incorporate two modules: the front-end LiDAR odometry and back-end mapping optimization (Latif et al., 2014). The front-end provides initial pose estimation through frame-to-frame registration, while the back-end employs alignment and optimization methods on a global scale to refine the pose estimation and reconstruct the surroundings. This approach enables the framework to achieve coarse-to-fine localization. It's universal to solve the normal scenes, such as cities-like or flat roads. Yet, in the uneven cases, one deployed on the ground vehicles produce non-negligible drifts of localization. The aroused intuitive phenomenon is that the reconstructed map suffers from distortion or oblique. The substantial reason is that the pose components in the vertical direction undergo significant changes due to the vehicle's body shaking in response to the terrain surface. These changes directly lead to a reduction in the consensus set during frame-to-frame matching, resulting in incorrect initial pose estimation by the front-end. Although the previous efforts (Shan and Englot, 2018) attempt to solve it and have achieved notable improvements, the issue remains challenging and needs further enhanced solutions.

To address the aforementioned problem, we propose ROLO-SLAM: A **R**otation-**O**ptimized **L**iDAR-**O**nly SLAM framework to estimate the ground vehicle's pose more precisely upon uneven terrains. In detail, we initiate our analysis by examining the positional change rule throughout the motion. Then, we establish a state front propagation attaining a soft decoupling of translation and rotation between consecutive data frames. After that, we follow the two-step method to get the initial pose estimation from the front-end: (a) *Rotation Estimation*, we formulate a rotation registration method using a degenerative spherical Gaussian process and multi-layer spherical alignment to independently predict the rotation of LiDAR locomotion. (b) *Translation Estimation*, after the *Rotation Estimation* step, we formulate the translation estimation problem as a unified optimization task that combines geometric alignment and continuous trajectory. We will find a suitable translation by achieving the convergence of translation error. Subsequently, the initial pose estimation is offered to the back-end module. We extract specific quantity keyframes to build a submap and finally apply the scan-to-submap optimization to expect a more accurate pose estimation. In the meantime, a global factor graph is incrementally constructed for global pose optimization. Overall, our main contributions lie in the following aspects:

- We leverage a state front propagation to achieve a soft decouple between rotation estimation and translation estimation, which allows us to estimate the rotation and translation independently.

- Towards alignment of the two consecutive data frames, a rotation registration method is first developed to optimize rotation by using a spherical alignment model. Followed by the continuous-trajectory-based translation estimation is optimized by considering the geometry and locomotion alignment.
- We design a robust LiDAR SLAM framework upon uneven terrain with ground vehicles. Our framework applies the optimized frame-to-frame registration as front-end while leveraging the scan-to-submap optimization and global optimization as back-end. Finally, the proposed framework is able to achieve more precise pose estimation and mapping tasks.

Different experiments are conducted to validate the efficacy of the proposed method and the results showcase that our method has the best unified performance compared with the state-of-the-art SLAM framework.

This article is organized as follows. Sec. 2 discusses the related research. Sec. 3 addresses problem formulation and potential causes, followed by the SLAM system pipeline and detailed mechanism in Sec. 4. Subsequently, the conducted experimental evaluation is presented in Sec. 5 and the corresponding results analysis is illustrated in Sec. 6. Finally, Sec. 7 summarizes our conclusions and discusses future research directions.

2 | RELATED WORK

Although the field of LiDAR-based localization with ground vehicles has incubated many application samples, such as search and rescue, autonomous driving and subterranean exploration (Jian et al., 2022; Ebadi et al., 2020; Xue et al., 2023), the vehicle's localization performance is worried under harsh environments. Regarding the first LiDAR-based method (Zhang and Singh, 2014), the large drifts in the vertical direction are produced when the vehicle passes uneven roads, which results in the skewed and overlapped point cloud map. To date, many studies attempt to solve the issue (Wang et al., 2021; Lin and Zhang, 2020; Chen et al., 2022a). These methods enhance the performance of accuracy and real-time, however, the reduction of vertical errors is limited when deployed under harsh environments.

One anticipated method to reduce the aforementioned problem is promoting the accuracy of point registration. Iterative closest point (ICP) algorithm (Pomerleau et al., 2013) and its variant are generally used for achieving the alignment tasks in the LiDAR-based SLAM. (Li et al., 2022) propose a refined LiDAR SLAM framework using generalized ICP (GICP) algorithm, which exploits the spatial voxelization to

enhance the point-matching process. To improve the robustness towards LiDAR point cloud, (Chen et al., 2021) leverages the normal distribution matching to replace the point-to-point matching by pursuing the maximum of joint probability. (Delkenbach et al., 2022) presents a CT-ICP method by adding a continuous time constraint to the optimization function to pursue smoother pose transformation. This method is significantly efficient towards variant terrain.

Furthermore, feature-based methods (Choi et al., 2023; Guo et al., 2023; Guo et al., 2022) upgrade the efficiency and efficacy of registration compared with the original ones by aligning representative features, such as edge features and planar features in the environments. (Wang et al., 2022a) develop FEVO-LOAM framework towards solving vertical drift problem, which enhances the feature extraction to capture the valid line, planar and ground feature points. (Chen et al., 2022c) formulate a slope feature extraction method and incorporate it into a factor graph model to achieve optimized pose estimation in uneven terrain. Nevertheless, these methods all simultaneously estimate translation and rotation, leading to a vast solution space for optimization problems and hindering rapid convergence. (Shan and Englot, 2018) propose the LeGO-LOAM method towards accurately estimating the pose in variant terrain. This method divides the estimation of the pose into steps and optimizes the results using different features, which enable to shrinking of the solution space to guarantee the convergence quality of the solution. (Yang et al., 2020) use the translation-invariant principle to achieve the rotation and translation registration independently. This method has been verified to decrease the optimization solution space and achieve improved results through numerous experiments.

Another method to address the vertical drift problem is fusing multi-sensor information. This method typically leverages multi-scale features of environments to compensate for the absence of vertical observation using multiple sensors such as LiDARs, cameras, and IMUs (Chen et al., 2022b; Wang et al., 2022b). (Shan et al., 2020) propose a tightly-coupled LiDAR-inertial SLAM method, which uses the IMU and LiDAR to tightly follow the transformation of pose and adopts a factor graph to fuse these observations. SDV-LOAM (Yuan et al., 2023) leverages the abundant camera's observation to achieve a semi-direct odometry in the front-end while the LiDAR sensor is used for executing sweep-to-map optimization in the back-end. However, in multi-sensor fusion scenarios, the integration of data from various sensors introduces additional errors unless precise calibration is achieved.

With the soaring in popularity of deep learning (DL) techniques, more researchers are recognizing the potentiality of DL methods to address challenges in environmental sensing and data association. The general methods (Pais et al., 2020; Li et al., 2019; Chen et al., 2020) leverage the neural network

to accomplish the complex matching and recognition tasks, which are under sparse sensor observation. (Chen et al., 2019) present SuMa++ method to achieve high-quality mapping and odometry measurement tasks, extracting the semantic information by network and combining it with the surfel feature to construct a continuous semantic map. (Deng et al., 2023) and (Ruan et al., 2023) exploit neural networks to render point clouds into the mesh map by state-of-the-art 3D reconstruction technique. While DL-based approaches offer powerful solutions for complex registration challenges, guaranteeing their processing speed and solution stability remains a challenge.

3 | PROBLEM DEFINITION

3.1 | Notation

In the world frame $\mathcal{W} \in \mathbb{R}^3$, we denote \mathcal{B} and \mathcal{L} as the vehicle and LiDAR frame respectively. one's pose can be written as

$$\mathbf{x} = [\mathbf{R}^\top, \mathbf{t}^\top]^\top, \quad (1)$$

where $\mathbf{R} \in SO(3)$ represents the rotation matrix. $\mathbf{t} \in \mathbb{R}^3$ is translation vector. We define the Euclidean transformation $\mathbf{T} \in SE(3)$ as

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}. \quad (2)$$

Concerning a vehicle-LiDAR rigid connection system, the vehicle's pose ${}^{\mathcal{B}}\mathbf{x}$ can be derived from LiDAR's pose ${}^{\mathcal{L}}\mathbf{x}$, which is computed by:

$${}^{\mathcal{B}}\mathbf{x} = [{}^{l_1}T \dots {}^{l_{n-1}}T] {}^{\mathcal{L}}\mathbf{x}, \quad (3)$$

where $l_i, i \in 1, 2, \dots, n$ represent the links connecting vehicle and LiDAR. In addition, each scan \mathcal{P} from LiDAR is composed of point set $\{p_i \in \mathcal{P}\}$. $\mathbf{p}_i \in \mathbb{R}^3$ is the position vector of the point p_i . \mathcal{M} represents the point cloud map.

3.2 | Problem Description

Facing a ground vehicle localization case, we assume that the rigid connection is always maintained between the vehicle and LiDAR. Consequently, our primary objective is to ascertain LiDAR's pose ${}^{\mathcal{L}}\mathbf{x}$. Subsequently, we can derive the vehicle's pose ${}^{\mathcal{B}}\mathbf{x}$ using Eq. 3. However, when the vehicle moves in uneven terrains, most LiDAR SLAM methods produce visible pose drifts in the vertical direction when deployed on the ground vehicle. our observation of the issue focuses on the following aspects:

On the one hand, in contrast to the flat case, non-leveled terrain surface forces the vehicle's pose to produce change, especially in roll and pitch ones. In such scenarios, LiDAR

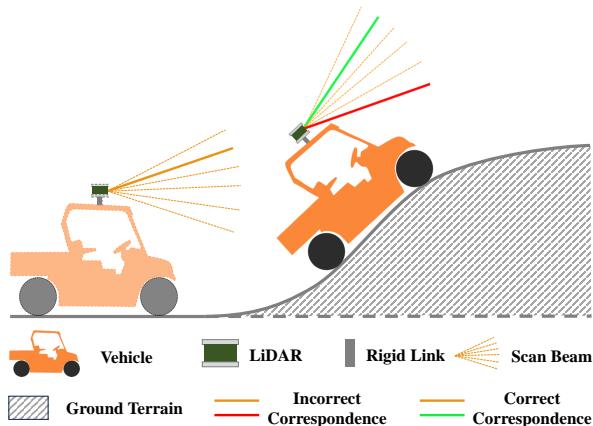


FIGURE 1 A simple case of easily suffering corresponding problems.

sensors exhibit large angular displacements in the vertical direction as the vehicle navigates uneven terrain. However, LiDAR’s sensitivity in the vertical direction is limited, leading to the gradual accumulation of pose errors.

On the other hand, most point cloud registration methods used in LiDAR SLAM exploit iterative optimization to approximate a solution. Fig. 1 shows a case of driving on uneven terrain. The dark orange vehicle occupies the position at the last moment, while the light orange one represents the current moment, denoted as \mathcal{B}_{t-1} and \mathcal{B}_t , respectively. In addition, the point cloud produced by LiDAR \mathcal{L}_{t-1} and \mathcal{L}_t are denoted as \mathcal{P}_{t-1} and \mathcal{P}_t respectively. In such case, the transformation \mathbf{T}_{t-1}^t of the scans can be calculated by

$$\mathbf{T}_{t-1}^t = \underset{\mathbf{T}}{\operatorname{argmin}} (\mathcal{P}_t - \mathbf{T}\mathcal{P}_{t-1}) = \underset{\mathbf{T}}{\operatorname{argmin}} \sum p_t^i - \mathbf{T}p_{t-1}^j, \quad (4)$$

where p_t^i and p_{t-1}^j are called a correspondence, while these correspondences compose consensus sets. The \mathbf{T}_{t-1}^t is determined by the maximum consensus set. However, in uneven cases, since the intense shaking from the vehicle, the points of two scans are easier to and produce incorrect correspondence. Furthermore, the optimization variable \mathbf{T} needs to account for changes in 6D pose, whereas in flat scenes, only changes in 3D pose are necessary. These factors lead to a reduction in the size of the maximum consensus set, thereby resulting in an inferior solution.

Problem: In this study, in the case of vehicle localization in uneven terrain, our target problem focuses on how to improve vehicle localization accuracy by using a single LiDAR sensor. To this end, we reconstruct the entire front-end and divide it into several modules to refine the pose estimation. Subsequently, we introduce scan-to-submap alignment and factor graph to optimize the vehicle’s pose within the map.

4 | ROLO-SLAM SYSTEM ARCHITECTURE

4.1 | System Pipeline of ROLO-SLAM

The architecture of our system is illustrated in Fig. 2. ROLO-SLAM is composed of two principal components: the LiDAR odometry module (front-end) and the Mapping module (back-end). Initially, the LiDAR scan undergoes rectification to correct motion distortion, utilizing odometry data sourced from the back-end. This rectification facilitates the front-end in approximating the LiDAR’s coarse odometry. In the front-end, geometric features are discerned based on their edge and planar characteristics, identified through a metric of smoothness (?). Subsequently, forward location adjustment is employed for a swift estimation of the LiDAR translation, which facilitates the loose decoupling of rotation and translation. This process is elaborated in Sec. 4.2. Correspondences are efficiently determined using a voxelization approach. The rotation and translation are independently ascertained, where the rotation is registered by spherical alignment model while the translation is calculated based on continuous trajectory optimization, detailed in Sec. 4.3 and Sec. 4.4. Advancing further, the back-end initiates by aggregating keyframes to construct submaps. These are leveraged to a scan-to-submap alignment, and further optimize the LiDAR’s global pose and point cloud map using a factor graph, as detailed in Sec. 4.5.

4.2 | Forward Location Adjustment

In the front-end processing, we aim to independently estimate the rotation and translation between consecutive scans by establishing a decoupling relationship between these two parameters. This is achieved by eliminating the translation differences through a forward location adjustment, thereby satisfying the required decoupling. Given the vehicle’s operation on uneven terrain, we assume that it remains on the ground surface, with its wheels consistently in contact with the ground. Additionally, since the observation frequency of most LiDARs exceeds 10 Hz, we can assume that the vehicle maintains uniform motion during these short scan intervals. Note that this uniform motion affects the location, while the attitude depends on the terrain. Fig. 3 displays snapshots of the vehicle during the same scan interval on the XOZ plane. As the vehicle travels across uneven terrain at a constant linear speed v , and experiences a position change δd , we can decompose δd as

$$\delta d = \sqrt{(\delta d_x)^2 + (\delta d_z)^2}, \quad (5)$$

where δd_x and δd_z are the translation components in x -axis and z -axis respectively. When the scan elapsed time $\delta \tau$ is small,

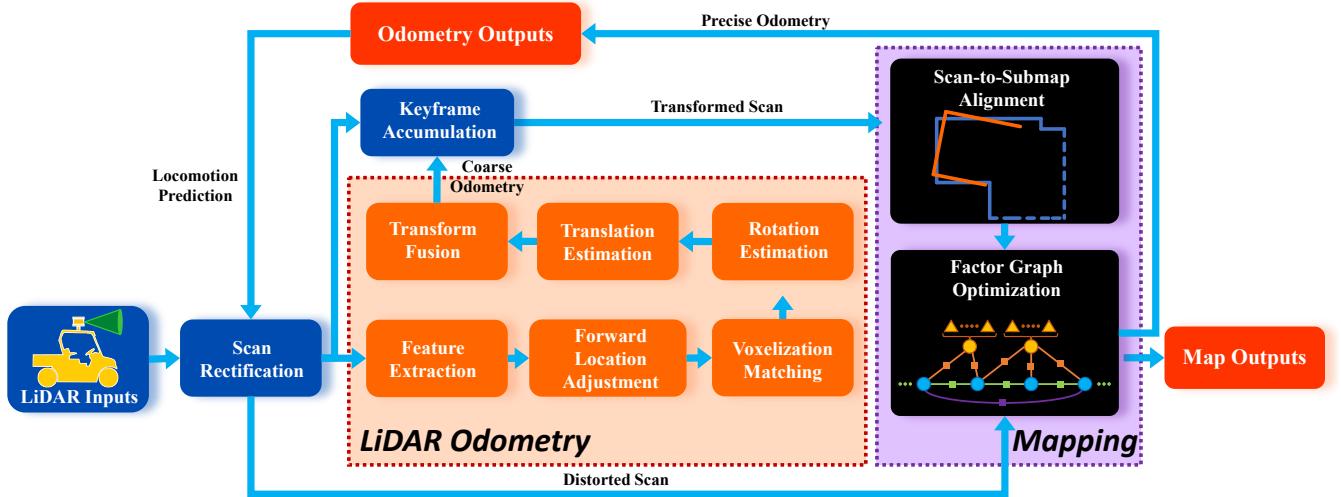


FIGURE 2 ROLO-SLAM system pipeline.

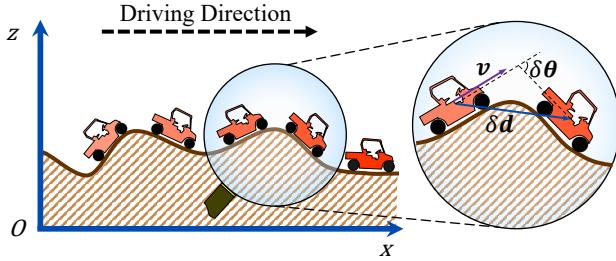


FIGURE 3 Snapshots of vehicle driving in uneven terrain. Where any two vehicle's pose is recorded in the same time intervals.

δd could be written as

$$\delta d = \lim_{\delta\tau \rightarrow \epsilon} \sqrt{(\mathbf{v}_x \delta\tau)^2 + (\mathbf{v}_z \delta\tau)^2} \quad (6)$$

$$= \lim_{\delta\tau \rightarrow \epsilon} \delta\tau \sqrt{(\mathbf{v}_x)^2 + (\mathbf{v}_z)^2} \quad (7)$$

$$= \lim_{\delta\tau \rightarrow \epsilon} \delta\tau \|\mathbf{v}\|, \quad (8)$$

Where ϵ is an infinitesimal, and \mathbf{v}_x and \mathbf{v}_z are the velocity components on the x -axis and z -axis, respectively. Eq. 6 illustrates that δd remains constant when the scan interval $\delta\tau$ is very small. In contrast to the rotation domain, the pitch angle of the vehicle θ_i is constrained by the ground surface, which has

$$\theta_i \propto \Gamma(t_i, S), \quad (9)$$

where $\Gamma(\cdot)$ represents the curvature of the ground, and S denotes the ground surface equation. In SLAM applications on uneven terrains, $\Gamma(\cdot)$ is typically unknown and nonlinear; therefore, the attitude change $\delta\theta$ does not remain constant during vehicle operation.

Denote the vehicle's k -th location as the t_k after undergoing k LiDAR scan $\mathcal{P}_{0:k}$. Before receiving the \mathcal{P}_{k+1} , the vehicle could preemptively estimate the location t_{k+1} by

$$t_{k+1} = t_k + \frac{\tau_{k+1} - \tau_k}{\tau_k - \tau_{k-1}}(t_k - t_{k-1}), \quad (10)$$

where the τ_i represents the timestamp of i -th scan. Eq. 10 utilizes the assumption that the vehicle maintains uniform motion across uneven terrain. Prior to the arrival of the scan \mathcal{P}_{k+1} , the vehicle's position at time τ_{k+1} is pre-estimated, thus effectively reducing the translation differences between the scans \mathcal{P}_k and \mathcal{P}_{k+1} . This procedure paves the way for independent subsequent estimations of rotation and translation for \mathcal{P}_k and \mathcal{P}_{k+1} .

4.3 | Voxelization Match and Rotation Registration Algorithm

In front-end LiDAR odometry estimation task, the rapid scan registration algorithm is maintained to obtain the transformation between two frames of data. The conventional method is majority to model the registration question as the Least-Square-Problem (LSP) solved by using non-linear optimization tools to align two frames of point cloud. By the way, the process of alignment is accompanied by simultaneous rotation and translation estimation. However, this simultaneity of the algorithm cannot focus on the accuracy of the outcome. In this paper, we attempt to obtain the continuous transformation by independent rotation and translation registration respectively. Through the state front propagation indicated at Sec. 4.2, we could consider that the ray centers of two data frames are approximately aligned. In other words, as for the adjacent point clouds, it

is assumed that only exists rotation angles' differences in the points.

In consideration of sparse data and fast matching, VGICP (Voxelized Generalized Iterative Close Point) algorithm (Koide et al., 2021) is validated for fast data association towards sparse point clouds, which is attributed to the efficient voxelized parallel operations. In this paper, we formulate a fast and robust data association method based on the sparse voxel maps and VGICP algorithm. Mathematically, gaussian distribution is generally used for describing noise of sensor's observation and continuing operation. Without loss of generality, the location \mathbf{p}_i of a 3D spatial point is assumed to suffer the white Gaussian noises, which is written as:

$$\mathbf{p}_i \sim \mathcal{N}(\hat{\mathbf{p}}_i, \Omega_i), \quad (11)$$

where, Ω_i represents the covariance matrix of Gaussian noise. Since the vertical discreteness of LiDAR observation, we exploit the invariety of linear operation between independent Gaussian distributions and develop a spacial sparse voxel map, whose each voxel records the joint Gaussian distribution of points lied in voxel. The detailed description of space voxelization and data association process is illustrated in Alg. 1. For each voxel m_k within spacial voxel map \mathcal{V} , it is established by storing distribution features including point number N_k , spatial centroid $\bar{\mathbf{p}}_k$ and distribution covariance $\bar{\Omega}_k$, which is given by

$$m_k \sim \mathcal{N}(\bar{\mathbf{p}}_k, \hat{\Omega}_k), \quad (12)$$

$$\bar{\mathbf{p}}_k = \frac{\sum_i \hat{\mathbf{p}}_i}{N_k}, \quad \bar{\Omega}_k = \frac{\sum_i \Omega_i}{N_k}. \quad (13)$$

In this article, the point cloud matching process is achieved by single index query operation. In detail, as for the two unregistered point clouds, the target point cloud \mathcal{P}_t suffers the spacial voxelization to formulate \mathcal{V} . When the source point cloud voxel index is lied in the index set $\text{index}(\mathcal{V})$, it means that the point pair is matched and the point corresponding set \mathcal{C} will be updated. It's worth mentioning that the spacial voxelization and matching operation is independent towards each voxel, thus they allow efficient and convenient parallel processing for fast voxel construction and matching.

Through the aforementioned state front propagation and data association, center-aligned adjacent data frames could singly estimate attitude transformation, which means that the LiDAR attitude and position estimation could be approximately decoupled. Within this context, the LiDAR rotation registration problem could be approximated as an alignment problem towards multi-layer spherical points, which is shown in Fig. 4. Two pairs of source points and target points lie on the different sphere layers respectively, where the target point is the expect of the matched voxel. Intuitively, the rotation alignment can

Algorithm 1: Voxelization and Matching

```

Input: Source point cloud:  $\mathcal{P}_s = \{\mathbf{p}_i^s \sim \mathcal{N}(\hat{\mathbf{p}}_i^s, \Omega_i^s)\}$ ,
        Target points cloud:  $\mathcal{P}_t = \{\mathbf{p}_i^t \sim \mathcal{N}(\hat{\mathbf{p}}_i^t, \Omega_i^t)\}$ ,
        Point corresponding  $\mathcal{C}$ 

1 Function Matching( $\mathcal{P}_s, \mathcal{P}_t$ ):
2    $\mathcal{C} \leftarrow \emptyset$ ;
3    $\mathcal{V} \leftarrow \text{Voxelization}(\mathcal{P}_t)$ ;
4   for  $\mathbf{p}_i^t \in \mathcal{P}_t$  do
5      $\text{voxel\_index} \leftarrow \text{int}(\mathbf{p}_i^t/\text{voxel\_resolution})$ ;
6     if  $\text{voxel\_index} \in \text{index}(\mathcal{V})$  and  $\mathcal{V}.N \geq N^+$ 
7       then
8          $\mathcal{C} \leftarrow \mathcal{C} \cup \text{make\_pair}(\mathbf{p}_i^t, \mathcal{V}.[\text{voxel\_index}])$ ;
9       end
10    end
11   return  $\mathcal{C}$ 

11 Function Voxelization( $\mathcal{P}$ ):
12    $\mathcal{V} \leftarrow \emptyset$ ;
13   for  $\mathbf{p}_i \in \mathcal{P}$  do
14      $\text{voxel\_index} \leftarrow \text{int}(\mathbf{p}_i/\text{voxel\_resolution})$ ;
15     if  $\text{voxel\_index} \notin \text{index}(\mathcal{V})$  then
16        $\text{index}(\mathcal{V}) \leftarrow \text{index}(\mathcal{V}) \cup \text{voxel\_index}$ ;
17     end
18      $\mathcal{V}[\text{voxel\_index}].N \leftarrow [\text{voxel\_index}].N + 1$ ;
19      $\mathcal{V}[\text{voxel\_index}].\bar{\mathbf{p}} \leftarrow [\text{voxel\_index}].\bar{\mathbf{p}} + \mathbf{p}_i$ ;
20      $\mathcal{V}[\text{voxel\_index}].\bar{\Omega} \leftarrow [\text{voxel\_index}].\Omega + \Omega_i$ ;
21   end
22    $[\mathcal{V}.\bar{\mathbf{p}}, \mathcal{V}.\bar{\Omega}] / = \mathcal{V}.N$ ;
23   return  $\mathcal{V}$ 

```

be seen as the process of the source point sliding and aligning to the target point between the multi-layer spherical layers.

Regarding a pair of source point and target point ($\mathbf{p}_s, \mathbf{p}_t$), they can be approximately seen lying on the same sphere surface, having the same sphere center \mathbf{O} . We aim to accomplish rotation registration by minimizing the sum of spherical angular discrepancy of all point pairs through a designated rotation, which could be written as

$$\mathbf{R} = \underset{\mathbf{R}}{\operatorname{argmin}} \sum \triangle(\mathbf{p}_t, \mathbf{R}\mathbf{p}_s), \quad (14)$$

where $\mathbf{R} \in SO(3)$ is the rotation matrix. $\triangle(\cdot)$ represents the spherical angle between corresponding points. The target function could be constructed by spherical Gaussian distribution (Banerjee et al., 2005). To streamline computations, we develop a registration method using a degenerated spherical Gaussian distribution and distance metrics.

As illustration in Fig. 5, we assume \mathbf{p}_t and \mathbf{p}_s lie on the same registered sphere while \mathcal{S} is the S is the spherical tangent plane passing through $\mathbf{R}\mathbf{p}_s$ with \mathbf{p}_t as its normal vector. \mathbf{p}'_t is

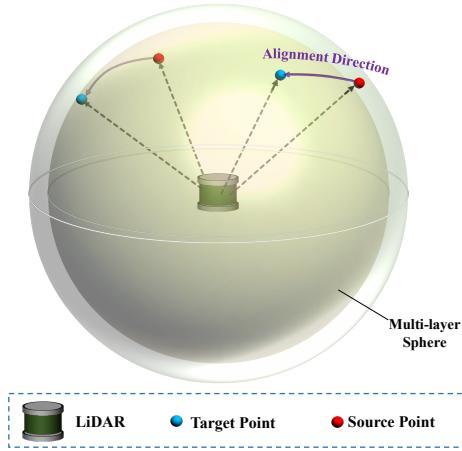


FIGURE 4 Rotation registration model

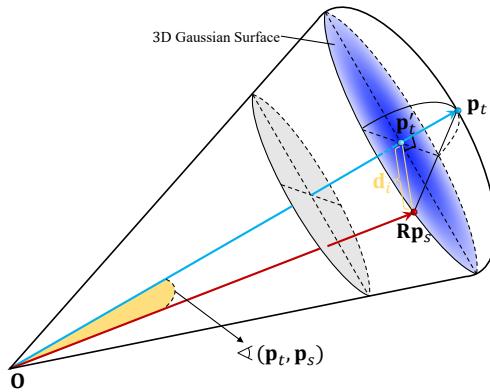


FIGURE 5 Corresponding point alignment

the projection from \mathbf{p}_t to \mathcal{S} . we have

$$\angle(\mathbf{p}_t, \mathbf{R}\mathbf{p}_s) = \arcsin \frac{\|\mathbf{d}_i\|}{\|\mathbf{p}'_t\|}, \quad (15)$$

$$\angle(\mathbf{p}_t, \mathbf{R}\mathbf{p}_s) \propto \|\mathbf{d}_i\|, \text{s.t., } \angle(\mathbf{p}'_t, \mathbf{R}\mathbf{p}_s) \leq \frac{\pi}{2}, \quad (16)$$

where $\mathbf{d}_i = \mathbf{p}'_t - \mathbf{R}\mathbf{p}_s$. Subsequently, we exploit the general point-plane relationship to calculate \mathbf{p}'_t by

$$\mathbf{p}'_t = (\|\mathbf{p}_t\| - \frac{(\mathbf{p}_t - \mathbf{R}\mathbf{p}_s) \cdot \mathbf{p}_t}{\|\mathbf{p}_t\|}) \mathbf{n}_t, \quad (17)$$

where \mathbf{n}_t represents the direction vector of \mathbf{p}_t . Without loss of generality, we adopt the mahalanobis distance form to construct object function by

$$\mathbf{R} = \operatorname{argmin}_{\mathbf{R}} \sum \|\mathbf{d}_i\|_{\Omega^{-1}}, \quad (18)$$

where $\|\cdot\|_{\Omega^{-1}}$ represents the mahalanobis distance while $\Omega_{\mathbf{R}}$ is the covariance matrix. Furthermore, a surfel approximation method is leveraged to approximate the spacial manifold of

spherical Gaussian distribution. Since the \mathbf{p}_t and $\mathbf{R}\mathbf{p}_t$ associate spacial Gaussian distribution respectively, we obtain the covariance of \mathbf{d}_i by

$$\Omega'_{\mathbf{R}} = \Omega_t + \mathbf{R}^T \Omega_s \mathbf{R}, \quad (19)$$

where Ω_t and Ω_s are covariance matrix of \mathbf{p}_t and \mathbf{p}_s respectively. We then perform singular value decomposition (SVD) on $\Omega'_{\mathbf{R}}$ and reconstruct the covariance $\Omega_{\mathbf{R}}$ as

$$\Omega_{\mathbf{R}} = \mathbf{U} \begin{bmatrix} \lambda_{max} & & \\ & \lambda_{max} & \\ & & \dots \end{bmatrix} \mathbf{V}. \quad (20)$$

λ_{max} is the maximum of eigen values while \mathbf{U} and \mathbf{V} are obtained by SVD operation. From the perspective of manifold, this operation results in the spherical Gaussian distribution being approximated into a circle surfel.

4.4 | Continuous Trajectory Translation Optimization

The state front propagation process relies on the last translation ($\mathbf{t}_c - \mathbf{t}_l$) to offer an initial guess for center alignment and rotation estimation towards adjacent frames. As for ground vehicles, one driving over a rough terrain is assumed to maintain continuous and normal locomotion during the observation interval of adjacent sensor data. Furthermore, the two adjacent point clouds haven't obvious attitude differences through the rotation registration. We leverage the principle of continuous and uniform locomotion, developing a continuous trajectory translation estimation method for sustaining state front propagation.

For the sake of brevity, we use the symbols ${}^o\mathbf{T}$ and ${}^n\mathbf{T}$ to represent the transformation matrix without translation and rotation respectively. With respect to the state front propagation and rotation estimation, we could obtain an initial translation guess ${}^n\hat{\mathbf{T}}_{sfp}^n$, rotation transformation ${}^o\mathbf{T}^n$. In this part, the translation estimation ${}^n\hat{\mathbf{T}}^n$ is further optimized by developed continuous trajectory method. The overall translation estimation is modeled as the following optimization process:

$${}^n\hat{\mathbf{T}}_{opt}^n = \operatorname{argmin}_{\mathbf{T}} \sum_i (F_{ICP}[\mathbf{T}] + \lambda F_{CT}[\mathbf{T}]), \quad (21)$$

where objective function incorporates two components: $F_{ICP}[\cdot]$ is the loss function of point-to-distribution while $F_{CT}[\cdot]$ is the loss function of continuous trajectory constrain. The *lambda* is adjustable weight for $F_{CT}[\cdot]$. To facilitate iterative solution, our method adopts mahalanobis distance $\|\cdot\|_{\Omega^{-1}}$ to represent the

loss function. For each corresponding $[\mathbf{p}_i^{n-1}, m_j^n] \in \mathcal{C}$, we have

$$F_{ICP}[\neg \mathbf{T}] = \|N_j(\bar{\mathbf{p}}_j^n - \neg \mathbf{T} \mathbf{p}_i^{n-1})\|_{\Omega_{ICP}^{-1}}, \quad (22)$$

$$\Omega_{ICP} = \bar{\Omega}_j^n + \neg \mathbf{T}^\top \Omega_i^{n-1} \neg \mathbf{T}, \quad (23)$$

where N_j is the number of internal points of voxel m_j^n . Since vehicle driving on the terrain has the velocity invariety in short time, the $F_{CT}[\cdot]$ towards each point \mathbf{p}^n can be written as

$$F_{CT}[\neg \mathbf{T}] = \|\mathbf{t}^n - \mathbf{t}^{n-1}\|_{\Omega_{CT}^{-1}}, \quad (24)$$

$$\mathbf{t}^n = (\neg \mathbf{T} \mathbf{p}_i - [\neg \hat{\mathbf{T}}_{sf}^{-1}] \mathbf{p}_i), \quad (25)$$

$$\mathbf{t}^{n-1} = [\neg \mathbf{T}^{n-1}]^{-1} \mathbf{p}_0, \quad (26)$$

$$\Omega_{CT} = \mathbf{t}^n \otimes \mathbf{t}^{n-1}, \quad (27)$$

where \mathbf{p}_0 is a zero vector to extract the last-moment translation. In addition, the \mathbf{t}^n and \mathbf{t}^{n-1} are two adjacent translations, whose right superscript denotes the time sequence. The covariance Ω_{CT} is the tensor product of \mathbf{t}^n which means the corresponding point with large translation suffers bigger penalty. $F_{ICP}(\cdot)$ is leveraged to solve the geometric alignment of sensor data while $F_{CT}(\cdot)$ forces the vehicle to maintain continuous uniform locomotion as possible. The final translation transformation $\neg \mathbf{T}^n$ of two sensor frames is calculated by

$$\neg \mathbf{T}^n = \neg \mathbf{T}_{opt}^n \neg \mathbf{T}_{sf}^n. \quad (28)$$

Of note is that solving the problem in Eq. 21 no longer executes the matching process formulated in Sec. 4.3, we directly inherit the corresponding set \mathcal{C} from rotation registration since we find the propagated translation is almost accurate while Eq. 21 is to further refine translation. It's beneficial to prompt the process speed of the front-end.

4.5 | Back Mapping and Loop Closure

Building on the relatively precise transformation estimation output from the front-end, the back-end mapping module becomes more efficient in refining these estimations, thereby enabling the production of high-quality global poses and environmental maps. The back mapping module consists of two steps of optimization: scan-to-submap optimization and global pose optimization. Among which, scan-to-submap optimization exploits a more meticulous registration method to align the latest data frame to the accumulated local submap for obtaining more precise LiDAR odometry. On the global scale, a global factor graph is incrementally constructed from the accumulated keyframes and delicately adjusts the pose of each history keyframe to minimize the overall historical errors.

4.5.1 | Scan-to-Submap Optimization

To achieve more precise LiDAR odometry, the global map is comprised of edge map and planar map accumulated by extracted feature points from origin data frames. After these feature points \mathbf{F} are extracted, they are transformed into world coordinates \mathcal{W} , where ${}^W\mathbf{F} = \{{}^W\mathcal{F}_e, {}^W\mathcal{F}_p\}$. Subsequently, our method selects specific data frames as keyframes \mathbb{F} to insert the global map. The keyframe selection is based on predetermined thresholds for distance, angle deviation, and time interval. We extract a fixed numbers k of most neighbor ${}^W\mathbf{F}$ to form a submap \mathcal{M}^i , we have

$$\mathcal{M} = \{\mathcal{M}_e, \mathcal{M}_p\}, \quad (29)$$

$$\mathcal{M}_e = \{{}^W\mathcal{F}_e^i, {}^W\mathcal{F}_e^{i-1}, {}^W\mathcal{F}_e^{i-2}, \dots, {}^W\mathcal{F}_e^{i-k+1}\}, \quad (30)$$

$$\mathcal{M}_p = \{{}^W\mathcal{F}_p^i, {}^W\mathcal{F}_p^{i-1}, {}^W\mathcal{F}_p^{i-2}, \dots, {}^W\mathcal{F}_p^{i-k+1}\}, \quad (31)$$

where \mathcal{M}_e and \mathcal{M}_p are corresponding edge submap and planar submap respectively. Scan-to-submap optimization leverages point-to-line and point-to-plane constraints to achieve the accurate registration from the current scan to submap, which can be written as

$$\mathbf{T}_b = \underset{\mathbf{T}_b}{\operatorname{argmin}} \sum_j (F_e[\mathbf{T}_b \mathbf{p}_e] + F_p[\mathbf{T}_b \mathbf{p}_p]), \quad (32)$$

where $F_e[\cdot]$ and $F_p[\cdot]$ are the loss function of edge features and planar features respectively. \mathbf{p}_e and \mathbf{p}_p are specific points in ${}^W\mathcal{F}_e$ and ${}^W\mathcal{F}_p$. The objective variety is the transformation matrix \mathbf{T}_b between the current scan to submap in the world coordinate. In our approach, $F_e[\cdot]$ characterizes the distance between the current edge features and the global edges within \mathcal{M}_e , whereas $F_p[\cdot]$ quantifies the distance between the current planar features and the global planar features within \mathcal{M}_p . They can be calculated by

$$F_e[\mathbf{T}_b \mathbf{p}_e] = W_e \frac{\|(\mathbf{p}_e^{\mathcal{M}} - \mathbf{T}_b \mathbf{p}_e) \times (\mathbf{p}_e^{\mathcal{M}} - \mathbf{T}_b \mathbf{p}_e - \mathbf{n}_e)\|}{\|\mathbf{n}_e\|}, \quad (33)$$

$$F_p[\mathbf{T}_b \mathbf{p}_p] = W_p \frac{(\mathbf{T}_b \mathbf{p}_p - \mathbf{p}_p^{\mathcal{M}}) \cdot \mathbf{n}_p}{\|\mathbf{n}_p\|}, \quad (34)$$

$$W_p = \frac{1}{|\sum \mathbf{n}_p^\top \mathbf{p}_p^{\mathcal{M}} + 1|}, \quad (35)$$

$$W_e = \sum \frac{\|\mathbf{n}_e\|}{\|(\mathbf{p}_e^{\mathcal{M}} - \bar{\mathbf{p}}_e^{\mathcal{M}}) \times \mathbf{n}_e\|}, \quad (36)$$

where $|\cdot|$ and $\|\cdot\|$ represent absolute value and L2 norm. $\mathbf{p}_e^{\mathcal{M}}$ and $\mathbf{p}_p^{\mathcal{M}}$ are feature points lied in \mathcal{M}_e and \mathcal{M}_p respectively. \mathbf{n}_e is the direction vector of line features while \mathbf{n}_p is the normal vector of planar features. In addition, W_e and W_p are the weight parameters of specific residuals, which focus on the features having a smaller distance to the corresponding edge or planar.

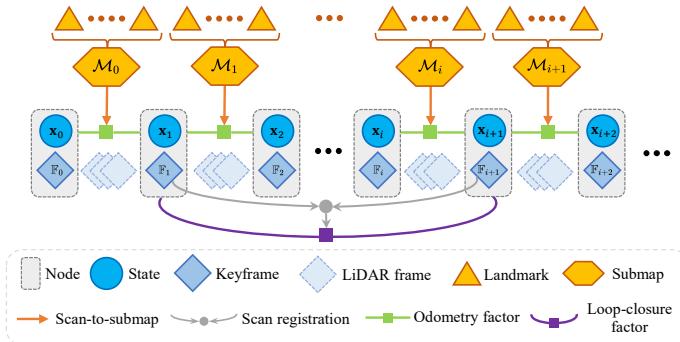


FIGURE 6 Factor graph model

This scan-to-submap process has low execution frequency for the better trade-off between real-time and accuracy.

4.5.2 | Global Optimization and Loop Closure

As we know, The whole optimization on global scale could be modeled as a Maximum a Posterior Problem (MAP). The factor graph (FG) model is generally leveraged to represent the nonlinear optimization structure of SLAM and conveniently solved by incremental smoothing and mapping (iSAM) algorithm. The whole FG is composed of nodes and edges with factors. The each node stores a LiDAR's state at any moment, we here define the LiDAR's state \mathbf{X}_i as

$$\mathbf{X}_i = [\mathbf{t}_i, \mathbf{R}_i], \quad (37)$$

where \mathbf{t}_i and \mathbf{R}_i are translation vector and rotation matrix, implying the position and attitude relative to the world coordinate origin. Furthermore, \mathbf{X}_i is associated with a keyframe \mathbb{F}_i . In this paper, we define two basic factors: odometry factor and loop-closure factor. Among them, the odometry factor applies to nodes that are temporally sequential and proximate in physical location. The loop-closure factor is utilized for cases where temporal discontinuity exists, yet spatial positions exhibit similarities. The overall factor graph model is illustrated by Fig. 6.

The odometry factor constrains the transform relation between the adjacent states and keyframes, like a Markov chain. It could reject abnormal states estimated by the scan-to-submap process and smooth the whole locomotion trajectory. The transformation linked with two adjacent nodes is given by the combined transformation from both the front end and the scan-to-submap process. It is unignorable for accumulated error in long-term and large-scale scenarios. Yet this issue could be efficiently addressed by the loop closure technique. The ROLO-SLAM exploits an efficient loop closure method based on the Euclidean Distance search. In detail, we set a search box

B with a stable size, which is centered on the current LiDAR state. There are states and keyframes surrounded by B , denoted as $B = \{\mathbb{F}_0, \mathbb{F}_1, \dots, \mathbb{F}_i, \dots\}$. These keyframes \mathbb{F}_i constitute a search queue, indexed according to the insertion times. During the locomotion of LiDAR, a loop closure thread is executed to check the similarity between every \mathbb{F}_i and the current keyframe \mathbb{F}_c . Once find similar \mathbb{F}_i and \mathbb{F}_c , a fast registration method, like the GICP algorithm, is exploited to compute transformation $\widehat{\mathbf{T}}_{\mathbb{F}_c, \mathbb{F}_i}$ within them. Meantime, we establish a loop-closure pair $\mathcal{L}_{c,i}$, denoted as:

$$\mathcal{L}_{c,i} = \{\mathbf{X}_c, \mathbf{X}_i, \widehat{\mathbf{T}}_{\mathbb{F}_c, \mathbb{F}_i}\}. \quad (38)$$

This loop-closure pair will be converted to a loop-closure factor inserting FG to establish a constraint between nodes inserted at different time instances. By employing FG for global optimization, LiDAR's trajectory can be refined, effectively eliminating accumulated errors and facilitating adaptive adjustments of all nodes in pursuing minimizing global errors.

5 | EXPERIMENTAL EVALUATION

5.1 | Platform and Experiments Setting

In this article, we present the ROLO-SLAM framework for localization and mapping in uneven terrain with ground vehicles. The expanded experiments are conducted to evaluate the performance of ROLO-SLAM. The comprehensive experiments focusing on pose estimation accuracy, robustness, computational efficiency, and mapping are performed using both public datasets and our meticulously prepared dataset. The detailed illustration of dataset is as follows:

- **KITTI-odometry (sequence 00, 05 and 08).** KITTI (Geiger et al., 2012) is a benchmark dataset for several odometry and perception tasks, which is captured in urban, rural, and highway environments, where the odometry benchmark could offer multi-modal data including LiDAR point clouds camera images (grayscale and color), etc. In addition, KITTI maintains the calibration parameters and benchmark results. In this paper, we mainly use KITTI-odometry dataset to evaluate the localization accuracy on the XY plane.
- **SDU Campus scene (Qianfoshan campus and Xinglongshan campus).** We formulate the Shandong University datasets in Qianfoshan campus and Xinglongshan campus, which both incorporate variant terrains, like slopes and uneven floors. The benchmark of localization is offered by the fusion of LiDARs, IMU and GPS sensors. This dataset is leveraged to evaluate the overall performance of our method.

TABLE 1 Dataset parameter setting

Dataset	KITTI_raw seq. 00	KITTI_raw seq. 05	KITTI_raw seq. 08	SDU Campus Qianfoshan	SDU Campus Xinglongshan	Offroad1	Offroad2	Offroad3
Have Loop	✓	✓	✓	✓				✓
Data Frames	4541	2761	4071	5370	5201	1882	2339	4336
Max Ground Height (m)	3.225	6.573	5.051	33.389	34.285	0.797	0.050	23.705
Min Ground Height (m)	-22.295	-12.092	-40.681	-0.105	-0.301	-29.308	-31.958	-2.986
Trajectory Length (m)	3387.473	1997.451	2879.163	2578.197	3580.265	927.516	1402.278	1140.229
Time Duration (s)	470	287	422	541	521	188	234	434

**FIGURE 7** Ground vehicle platform

- **Real Off-Road scene (Offroad1, Offroad2, Offroad3).**

These datasets are collected in off-road settings near Fohui Mountain, Jinan. They include various scenarios, like steep slopes, muddy roads, grasslands, and hard-surfaced roads. Furthermore, the dataset offers multiple sensor data and believable localization benchmark, which is suitable for evaluation in 6D pose estimation.

The Tab. 1 lists the detailed parameters of each dataset for reference. As for the ground vehicle platform, our elaborate datasets are recorded by a ground vehicle platform as shown in Fig. 7. The vehicle platform comes outfitted with a VLP-32C Puck 3D LiDAR, a ZED2i stereo camera, a GW-GNS100 GPS, and a Mti-300 9-axis IMU. It's worth mentioning that our approach solely relies on the VLP-16 Puck 3D LiDAR for accomplishing the localization task, although the platform also provides accurate calibration parameters and localization references through the fusion of multiple sensors.

To better indicate the performance of ROLO, we conduct abundant comparison experiments with state-of-the-art methods. These methods are as follows:

- **LOAM**, presented by (Zhang and Singh, 2014), first leverages the simple but efficient registration to achieve precise LiDAR odometry tasks and indicates high performance in durational localization accuracy.
- **LeGO-LOAM** (Shan and Englot, 2018), towards vertical accumulated drift problem, segments the ground observation and divides the 6-DOF pose estimation into sections. It has excellent accuracy in uneven terrains with ground vehicles.

- **CT-ICP** (Dellenbach et al., 2022) presents a continuous time ICP method designed to achieve smoother motion trajectories by incorporating time and velocity constraints. This approach demonstrates enhanced performance, particularly in the context of continuous motion scenarios.
- **HDL-SLAM** (Koide et al., 2019) integrates a variety of cutting-edge registration and loop-closure techniques to achieve thorough localization. Furthermore, it incorporates people behavior prediction and ground perception capabilities, thereby showcasing superior performance in long-term localization tasks.

Our implementation for ROLO-SLAM system is in C++ and Robot Operation System (ROS) Melodic, which is open-sourced in the repository [‡]. All evaluated algorithm is executed on a Levnovo Y9000P Laptop with an Intel Core i7th CPU, an Nvidia RTX2070 GPU and 32 GB RAM.

5.2 | Accuracy Evaluation

In this section, we evaluate the localization accuracy of our method with other state-of-the-art frameworks. In the beginning, we perform the evaluation in the public KITTI-odometry datasets (sequence 00, 05 and 08). The results are shown in Fig. 8.

Regarding the trajectories of odometry, the estimated trajectory of ROLO (ours) aligns closely with the ground truth (GT), while the efficacy is enhanced by the loop closure technique (ROLO+LC). Instead, the trajectories of other methods display different biases of varying magnitudes. The **LOAM** method has the largest drift from the perspective of trajectory contour, which is caused by the coarse feature extraction and front-end estimation. Since the locomotion pattern is relatively fixed, the **CT-ICP** method demonstrates a notably high degree of alignment with the Ground Truth in KITTI datasets. Additionally, the **LeGO-LOAM** and **HDL-SLAM** methods, when applied to the KITTI datasets, exhibit similar performance and approximately trace correct trajectories using their respective localization techniques. In a word, compared to alternative approaches, **ROLO** exhibits the greatest degree of

[‡] Code: <https://github.com/sdwyc/history-aware-planner>

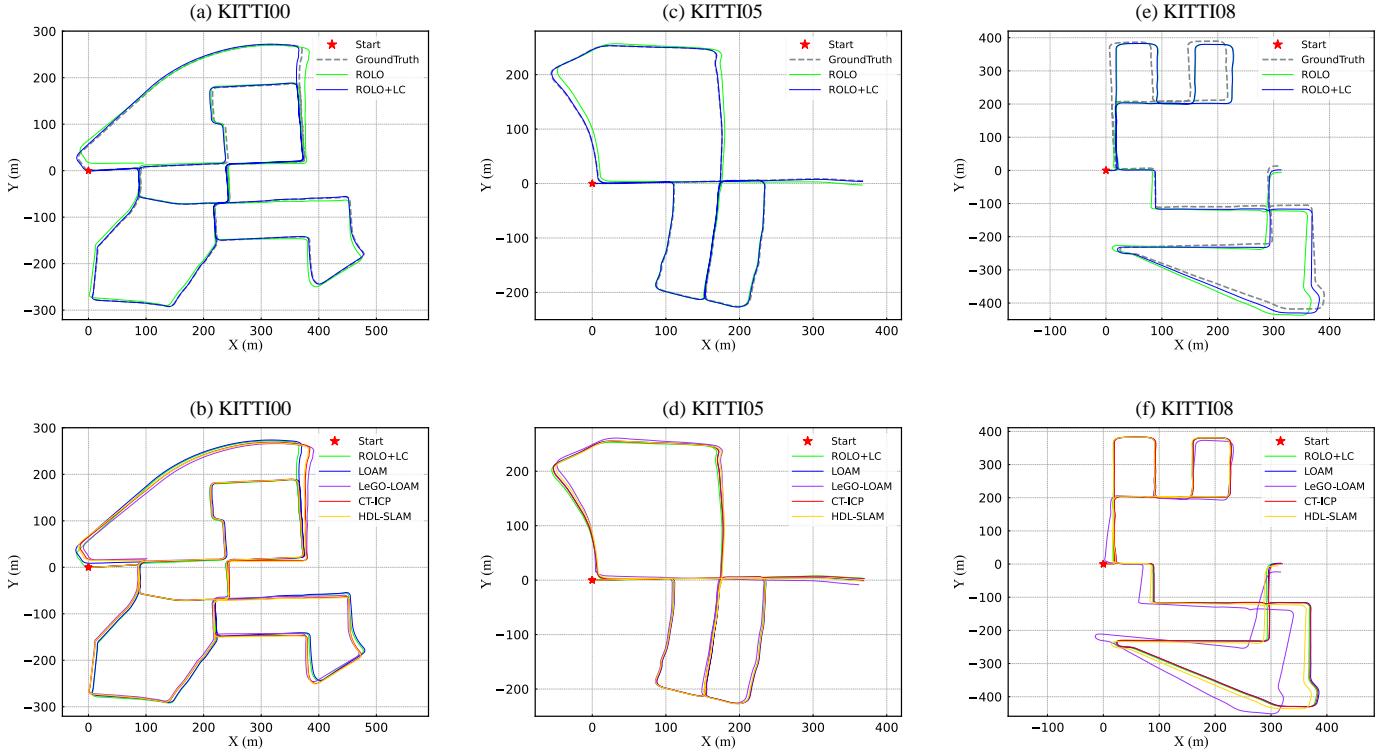


FIGURE 8 The trajectory estimation of ROLO, our method with loop closure correction (ROLO+LC) and other odometry frameworks for the KITTI-odometry sequence 00, 05 and 08.

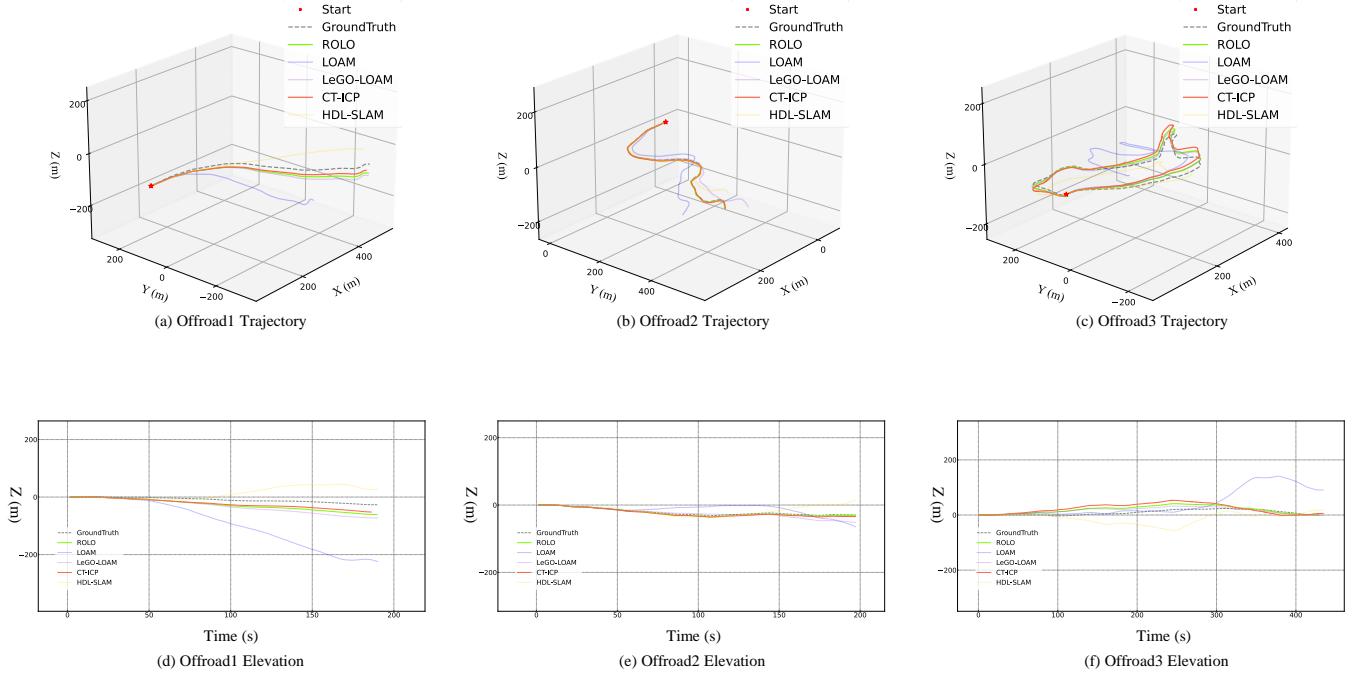


FIGURE 9 The trajectory estimation of our method and alternative methods for the off-road datasets (Offroad1, Offroad2, Offroad3).

alignment with the benchmark. These results highlight ROLO's exceptional performance in XY plane localization.

To evaluate the efficacy of z-axis localization, we leverage **Offroad1**, **Offroad2** and **Offroad3** datasets to test the

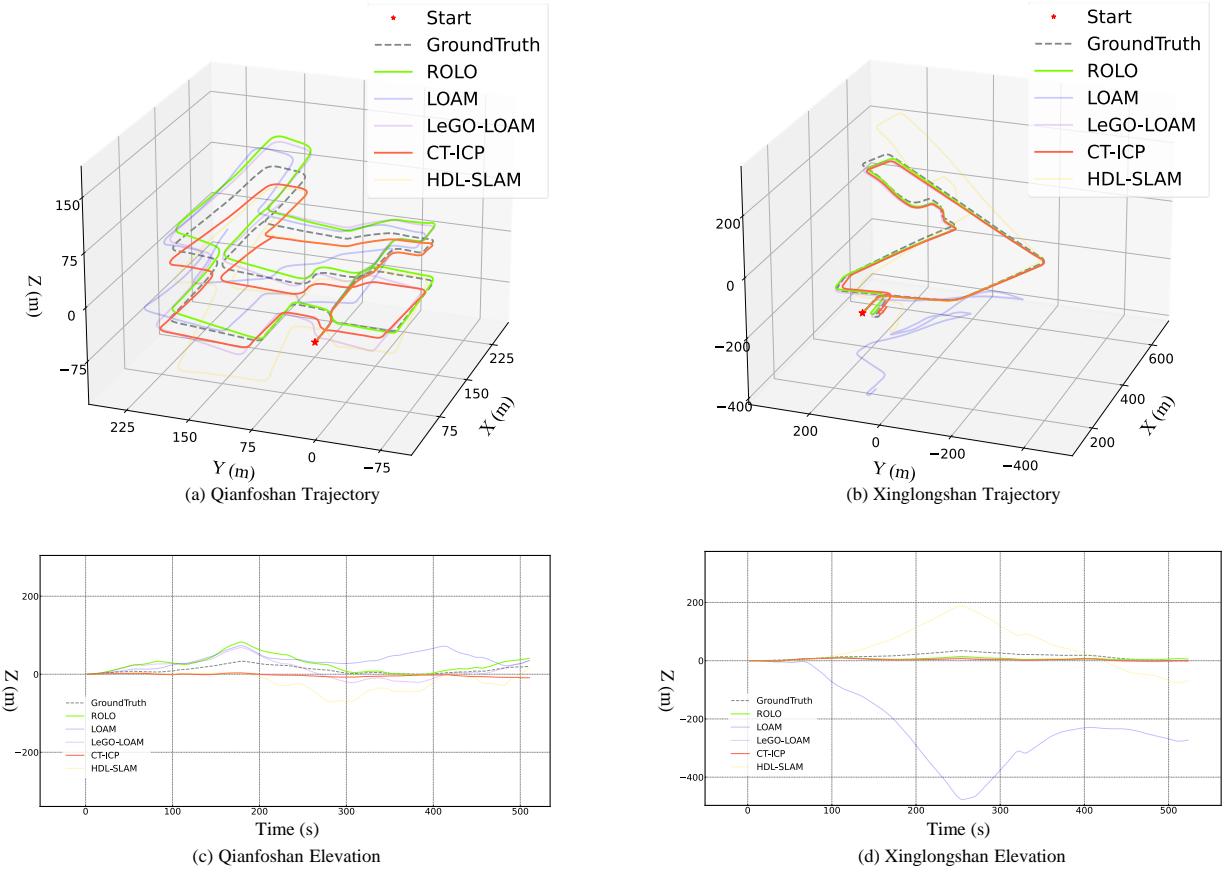


FIGURE 10 The trajectory estimation of our method and alternative methods for the SDU campus datasets (**Qianfoshan**, **Xinglongshan**).

proposed method. Note that these datasets are captured in real off-road and unstructured environments, characterized by numerous steep slopes and uneven road surfaces. These environmental features lead to intensive changes in the position and orientation of the mobile body. The evaluated trajectories are depicted in Fig. 9.

The **Offroad1** and **Offroad2** datasets are captured in the downhill scenes, one can see the descent at z-axis coordinates. Compared with other approaches, our method thanks to the rotation optimization technique achieves more precise z-axis changes and attitude estimation. In contrast, other methods utilize conventional point-to-plane and point-to-line registration techniques, which prove to be less effective in scenarios with sparse features and drastic changes in surface geometry.

Subsequently, the performance of pose estimation with these approaches is further validated using the SDU campus datasets (**Qianfoshan**, **Xinglongshan**). Each of these campus datasets features distinct terrain characteristics: **Qianfoshan** is characterized by steeper slopes, whereas **Xinglongshan** encompasses a broader regional expanse and features winding roads. The results is shown in Fig. 10.

Intuitively, the trajectory change trend of the proposed method accurately reflects the ground variations across different terrains, thanks to the sophisticated registration approach employed in the front-end. The **CT-ICP** method leverages excellent continuous motion constraints, and its lightweight registration process also yields impressive results. In contrast, other methods exhibit significant pose drift, especially in terms of z-axis positioning and roll, pitch attitude angles. These outcomes demonstrate our method's substantial advantage in 6-DOF pose estimation, as evident from the drawn trajectory. Furthermore, Tab. 2 lists the root mean square error (RMSE) of the translation and rotation estimation for all methods.

From the perspective of data, **ROLO** method has the lowest RMSE for the translation and rotation in the whole, while the effort is more obvious by the enhancement of loop-closure technique. The **CT-ICP** method also showcases excellent performance, benefiting from the optimal conditions for continuous movement. However, other approaches exhibit varying errors across different scenarios. Specifically, the **LOAM** and **HDL-SLAM** methods achieve precise localization efforts but suffer from significant drift on uneven roads. Conversely, the **LeGO-LOAM** method experiences noticeable drifts in large-scale and flat environments, despite its substantial localization

TABLE 2 RMSE (translation (m)\rotation (radian)) of methods with ground truth.

	KITTI seq.00	KITTI seq.05	KITTI seq.08	SDU camp (Qianfoshan)	SDU camp (Xinglongshan)	Offroad1	Offroad2	Offroad3
ROLO (ours)	0.248\0.471	0.194\0.617	0.355\0.310	0.450\0.149	0.188\0.049	0.806\0.178	0.153\0.068	0.358\0.003
ROLO+LC (ours)	0.200\0.457	0.061\0.510	0.291\0.311	0.596\0.299	0.222\0.052	0.448\0.171	0.089\0.027	0.344\0.003
LOAM	1.103\0.791	0.926\0.885	1.170\0.621	1.271\0.357	6.924\0.854	7.335\2.041	3.845\0.446	2.775\0.018
LeGO-LOAM	0.662\0.518	0.320\0.708	0.977\0.330	0.594\0.311	0.324\0.044	1.256\0.233	1.810\0.373	0.389\0.004
CT-ICP	0.246\0.456	0.093\0.516	0.306\0.365	0.541\0.175	0.396\0.052	0.686\0.178	0.133\0.037	0.549\0.005
HDL-SLAM	0.229\0.462	0.134\0.490	0.448\0.311	1.086\0.580	1.817\0.316	2.042\1.262	1.577\0.458	0.289\0.008

TABLE 3 Absolute Error for inter-frame (*mean* /*max.* /*std.*) towards SDU campus datasets

Datatsets	Methods	E^t (m)			E^R (rad)		
		x	y	z	roll	pitch	yaw
Qianfoshan	ROLO	1.445/2.390/0.414	0.731/1.908/0.636	0.081/0.225/0.046	0.002/0.031/0.003	0.002/0.021/0.002	0.001/0.026/0.002
	ROLO+LC	1.418/2.284/0.410	0.711/1.813/0.541	0.081/0.224/0.046	0.003/0.020/0.004	0.003/0.020/0.004	0.001/0.014/0.002
	LOAM	1.370/2.212/0.390	0.800/2.416/0.670	0.081/0.222/0.046	0.005/0.051/0.006	0.004/0.044/0.005	0.002/0.015/0.001
	LeGO-LOAM	1.469/2.347/0.419	0.750/1.822/0.522	0.081/0.414/0.048	0.005/0.078/0.006	0.005/0.053/0.007	0.004/0.050/0.004
	CT-ICP	1.442/2.274/0.409	0.723/2.000/0.675	0.081/0.226/0.046	0.002/0.023/0.003	0.002/0.025/0.002	0.001/0.013/0.001
	HDL-SLAM	1.402/2.244/0.398	0.862/2.181/0.606	0.082/0.572/0.050	0.005/0.126/0.012	0.007/0.097/0.013	0.002/0.030/0.003
Xinglongshan	ROLO	1.110/2.799/0.675	0.907/2.015/0.346	0.037/0.135/0.021	0.001/0.022/0.001	0.001/0.018/0.002	0.001/0.020/0.002
	ROLO+LC	1.093/2.784/0.719	0.929/2.027/0.354	0.037/0.140/0.021	0.001/0.017/0.001	0.001/0.018/0.002	0.001/0.018/0.002
	LOAM	0.918/3.092/0.853	1.169/2.872/0.626	0.037/0.140/0.022	0.005/0.119/0.010	0.005/0.114/0.010	0.003/0.053/0.005
	LeGO-LOAM	1.040/3.239/0.726	0.930/2.038/0.358	0.037/0.134/0.022	0.002/0.088/0.004	0.002/0.032/0.003	0.003/0.058/0.004
	CT-ICP	1.031/2.828/0.724	0.933/2.035/0.357	0.037/0.138/0.022	0.001/0.009/0.001	0.001/0.009/0.001	0.001/0.015/0.001
	HDL-SLAM	1.005/2.812/0.745	0.818/1.941/0.338	0.038/0.376/0.027	0.002/0.048/0.005	0.002/0.063/0.005	0.001/0.013/0.001

accuracy on unlevel terrains, which thanks to its independent ground optimization.

5.3 | Robustness Evaluation

The localization robustness substantially reflects the capacity to sustain self-location stability. It is also one of the crucial factors in addressing localization drift in rough environments. We design experiments to evaluate the performance of our method in the localization robustness. We first define an absolute error for inter-frame transformation as:

$$E_i^R = \text{abs}[\text{euler}(\hat{T}_{i,i+1}) - \text{euler}(T_{i,i+1}^{gt})], \quad (39)$$

$$E_i^t = \text{abs}[\hat{t}_{i,i+1} - t_{i,i+1}^{gt}], \quad (40)$$

where, for the consecutive scan \mathcal{P}_i and \mathcal{P}_{i+1} , $\hat{T}_{i,i+1}$ is the transformation matrix between above two scans, while $\hat{t}_{i,i+1}$ is translation in $\hat{T}_{i,i+1}$. $T_{i,i+1}^{gt}$ and $t_{i,i+1}^{gt}$ are corresponding ground truth transformation. In addition, $\text{euler}(\cdot)$ denotes the Euler angles calculation. In effect, E_i^R and E_i^t both are 3D vectors. $\text{abs}[\cdot]$ represents the absolute value operation for each dimension. They show the instantaneous accuracy of each inter-frames transformation. Subsequently, the proposed method is executed 5 times in datasets. Fig. 11 shows the change curve of E^R and E^t . From a macro perspective, our method has small

amplitude and mean for each datasets, which is particularly obvious in the off-road cases. These results indicate that our method maintains the stability of localization along with high instantaneous accuracy. The reason is that ROLO achieves precise attitude estimation by independently estimating rotation and translation at the front-end. This further leads to final pose estimation with excellent accuracy.

In order to compare the localization robustness among methods, each method is executed 3 times in SDU campus datasets respectively. Tab. 3 shows the *mean*, maximum (*max*) and standard covariance (*std.*) of each dimension in E^R and E^t . Compared with other methods, ROLO has relatively small means, which guarantees the estimation accuracy for each scan. Of note is that ROLO has the smallest std. value. It demonstrates the estimation accuracy of our method is significantly stable in execution.

5.4 | Computation Efficiency Evaluation

The off-the-shelf methods typically adopt estimate translation and rotation at mean time. Differently, ROLO divides estimation into independent translation and rotation estimation at the front-end. This operation is not only bring precise estimation guesses for back-end, but also maintains the real-time process. To demonstrate the processing efficiency of our method, all

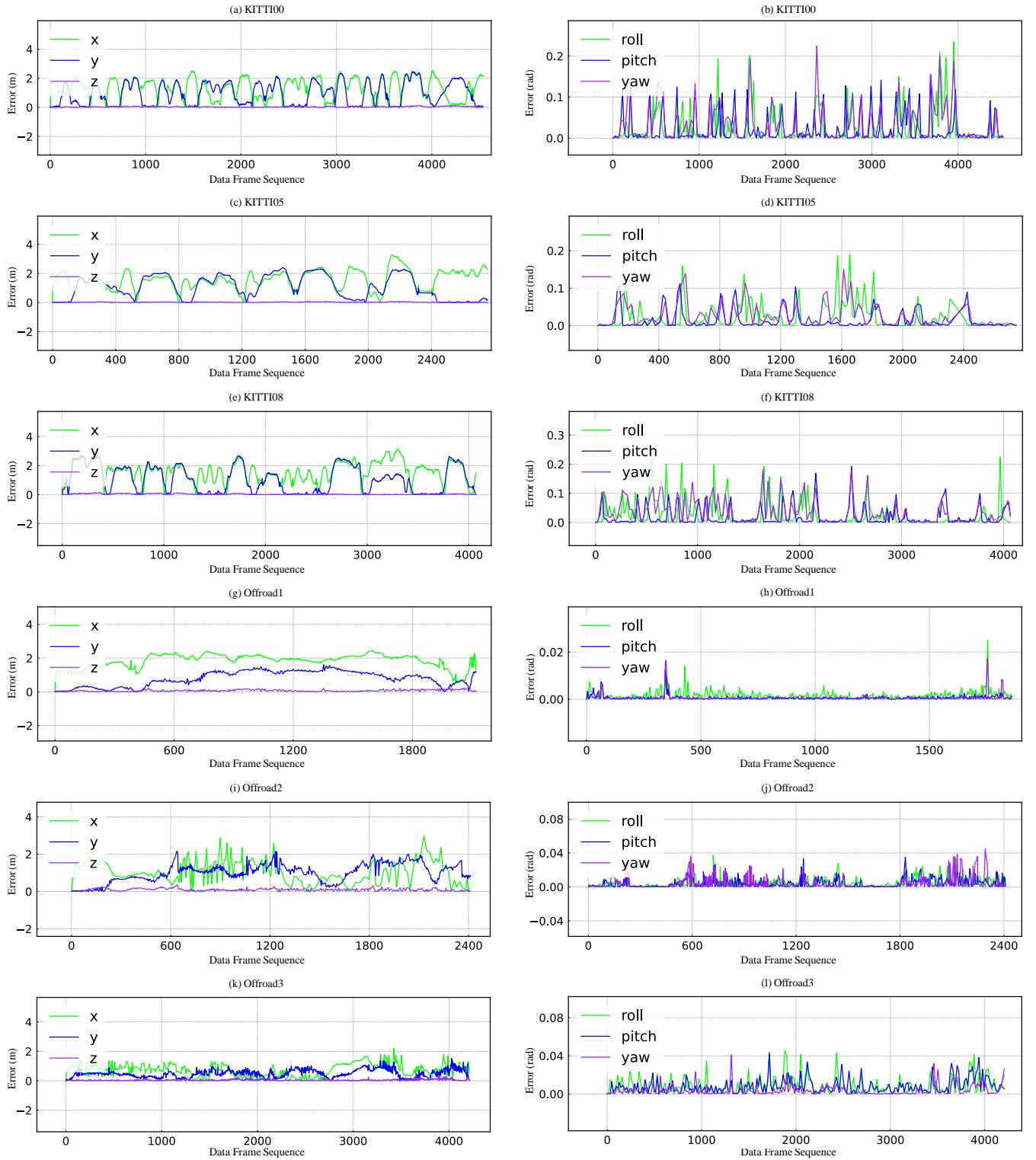
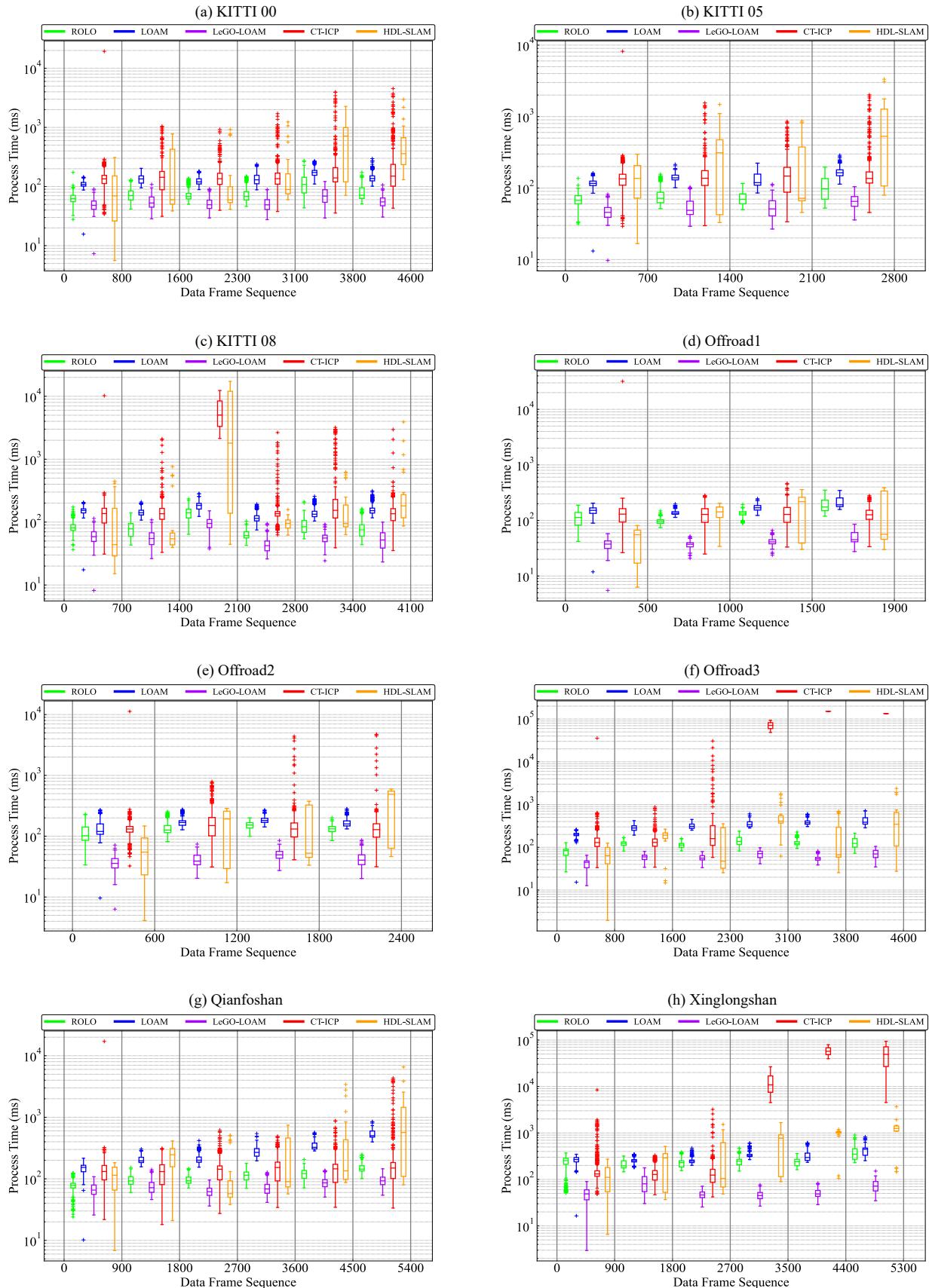


FIGURE 11 The absolute errors for all datasets.

data frames are timed for processing. Since the processing time is relevant to the saved global map size, we divide several pieces of datasets. Fig. 12 showcases box plots of process times

per scan, whose data is from the sum time of methods' front-end and back-end. Note that **CT-ICP** is recorded by the whole processing times since it is without back-end. The length of boxes represents the fluctuation range of processing time. We

**FIGURE 12** The absolute errors for all datasets.

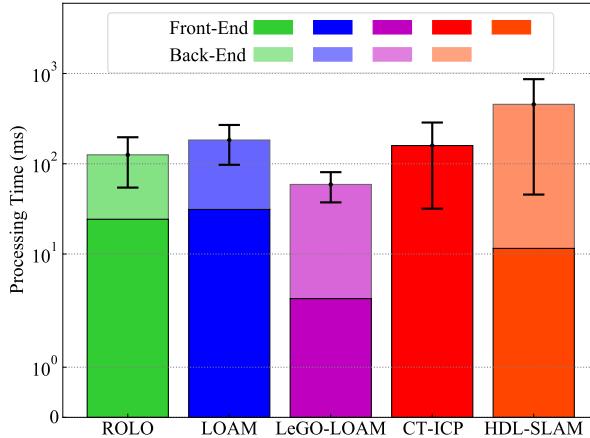


FIGURE 13 The overall processing time per scan for methods.

can see our method has relatively short boxes for all datasets. It means **ROLO** has robust processing capacity, whose processing time is basically stable at around 100 ms. The whole time boxes of **CT-ICP** are located at high position and has many high outliers. It indicates that the added motion constraints bring more processing delay though the localization accuracy obtains improvement. In contrast, **LeGO-LOAM** has excellent performance in processing efficiency but is worrisome in aspect of localization accuracy. In summary, **ROLO** has highly overall performance in spite of processing robustness and efficiency.

In addition, we also integrate all processing times and visualizing them in Fig. 13. Fig. shows the average processing times per scan of methods in all datasets which is equal with sum of front-end times and back-end times, and the error bar represents the fluctuation range of total processing time. The results show that the front-end of **ROLO** maintains fast processing speed, although it is divided into three successive modules. The total processing time is around 100 ms per scan, which could satisfy the requirements for real-time. Moreover, **CT-ICP** and **HDL-SLAM** have longer error bars, which means their processing efficiency is erratic.

5.5 | Mapping Results

In this evaluation, we focus on the mapping quality of the compared methods, and macroscopically observe localization accuracy on the global scale. At the beginning, Fig. 14 shows the overall mapping result under **Xinglongshan** dataset. Where Fig. 14(c) is a snapshot for a curved road with slope. The point cloud and trajectory both showcase high localization accuracy and mapping quality. To demonstrate the performance of localization in the non-level scenes, we compare the **HDL-SLAM**, **LeGO-LOAM** and **ROLO** on the **Qianfoshan** dataset, whose results are shown in Fig. 15. At a global scale, the

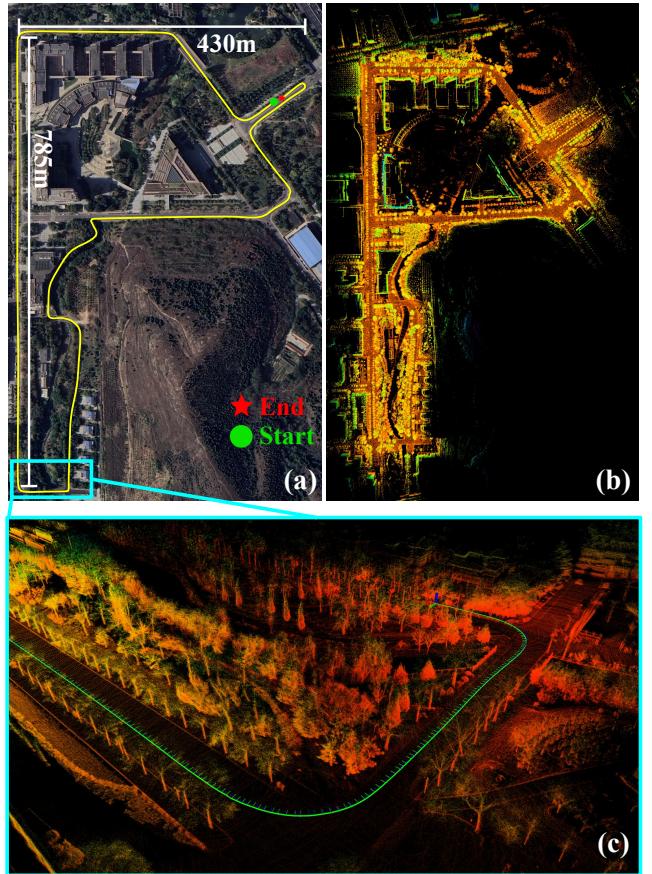


FIGURE 14 Mapping result for Xinglongshan dataset. (a) shows the Google map, which indicates the trajectory and distance metrics. (b) displays overall point cloud map and (c) is captured at a specific scenario.

point cloud mapping generated by our method exhibits high consistency with the scene contours in the satellite map. In contrast, the results from **HDL-SLAM** and **LeGO-LOAM** suffer from noticeable shape distortion, especially in the contours of buildings. From a side view perspective, **ROLO**'s result demonstrates a relatively flat mapping plane. However, the other two methods display significant oblique, with **HDL-SLAM**'s results showing an overlapping phenomenon. This elevation drifts, to a certain extent, contribute to a decline in the overall quality of the mapping. In contrast, **ROLO** has high performance in alleviating the elevation drift problem, which can be showcased in Fig. 16. **ROLO** is able to exhibit the correct trajectory and reconstruct original scenes with slopes.

Regarding the off-road cases, our method still retains stable performance in localization and mapping. In **Offroad3** case, the satellite map exhibits a rugged downhill, while the **ROLO**'s trajectory also has the same trend of decline and reconstructs the surroundings perfectly. Furthermore, Fig. 18 displays the overall point cloud aligned with a satellite map. We can observe

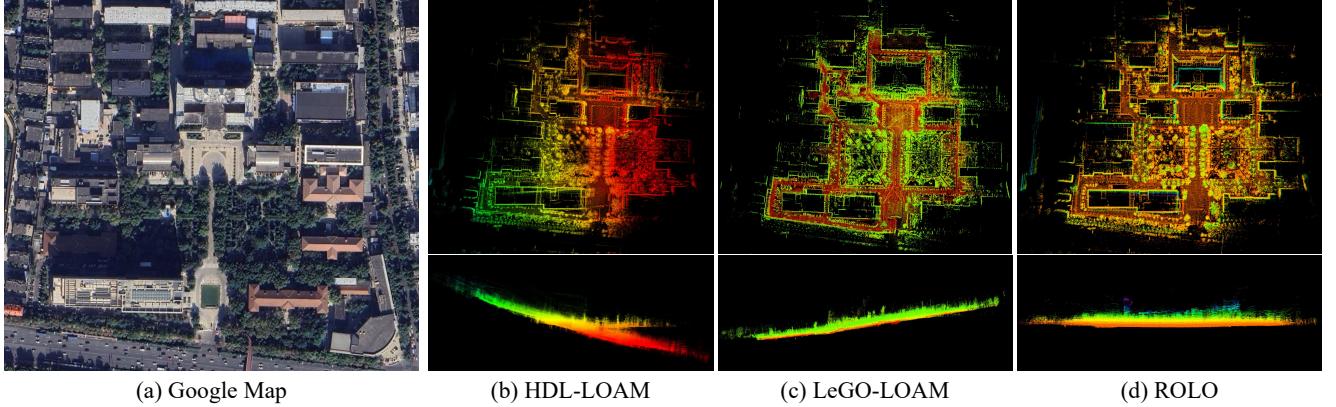


FIGURE 15 Mapping results of **HDL-LOAM**, **LeGO-LOAM** and **ROLO** in **Qianfoshan** dataset. In (b)-(d), the top figures show the overall maps while the bottom figures are captured at the side views.

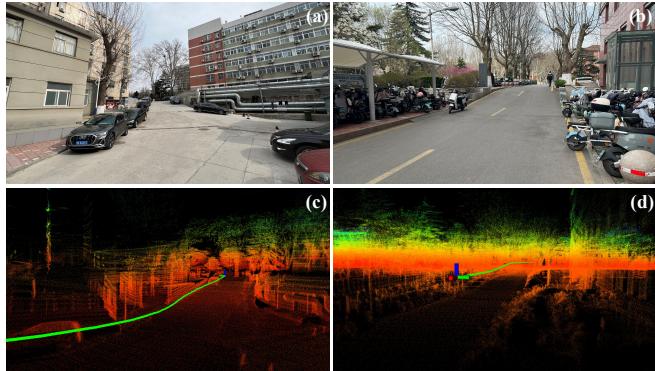


FIGURE 16 Mapping in slope case

that the constructed point cloud coincides with the terrain contours in the satellite map. The right three snapshots show the real scene at driving. The corresponding results are located at left, which both reflect high mapping quality and localization accuracy. In conclusion, ROLO has high efficiency and efficacy in robust and real-time localization and mapping, particularly in reducing elevation drift. These advantages enable ROLO to operate reliably in challenging environments, such as off-road terrain.

6 | RESULT ANALYSIS AND PROSPECT

The overall experimental results illustrate our method has high performance in attitude estimation and elevation error decrement. Specifically, ROLO leverages independent rotation and translation estimation at the front-end. Where the rotation estimation applies the spherical aligning model to achieve precise attitude alignment while the translation estimation considers

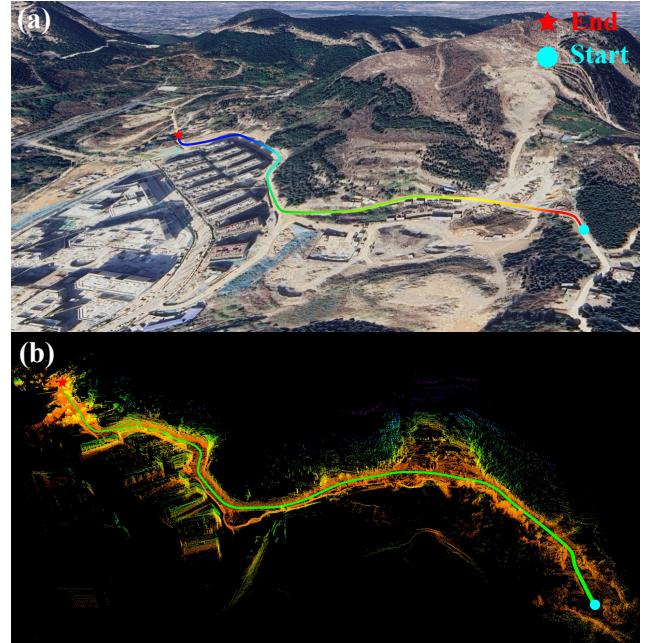


FIGURE 17 Mapping results in **Offroad2** dataset. (a) is the satellite map from Google Earth. (b) shows the point cloud and trajectory.

the geometry and motion model. The front-end offers an excellent pose guess for back-end, and the overall localization accuracy and robustness is guaranteed, whose final translation and rotation errors could be less than 0.26 meter and 0.21 radian respectively. In addition, our method still maintains real-time though the front-end function is divided and rearranged. CT-ICP seems to have similar great accuracy and robustness with ROLO, however, its processing time per scan is over 150 ms. While ROLO's is less than 120 ms. Our analysis indicates that the reason for this is that, without incorporating global

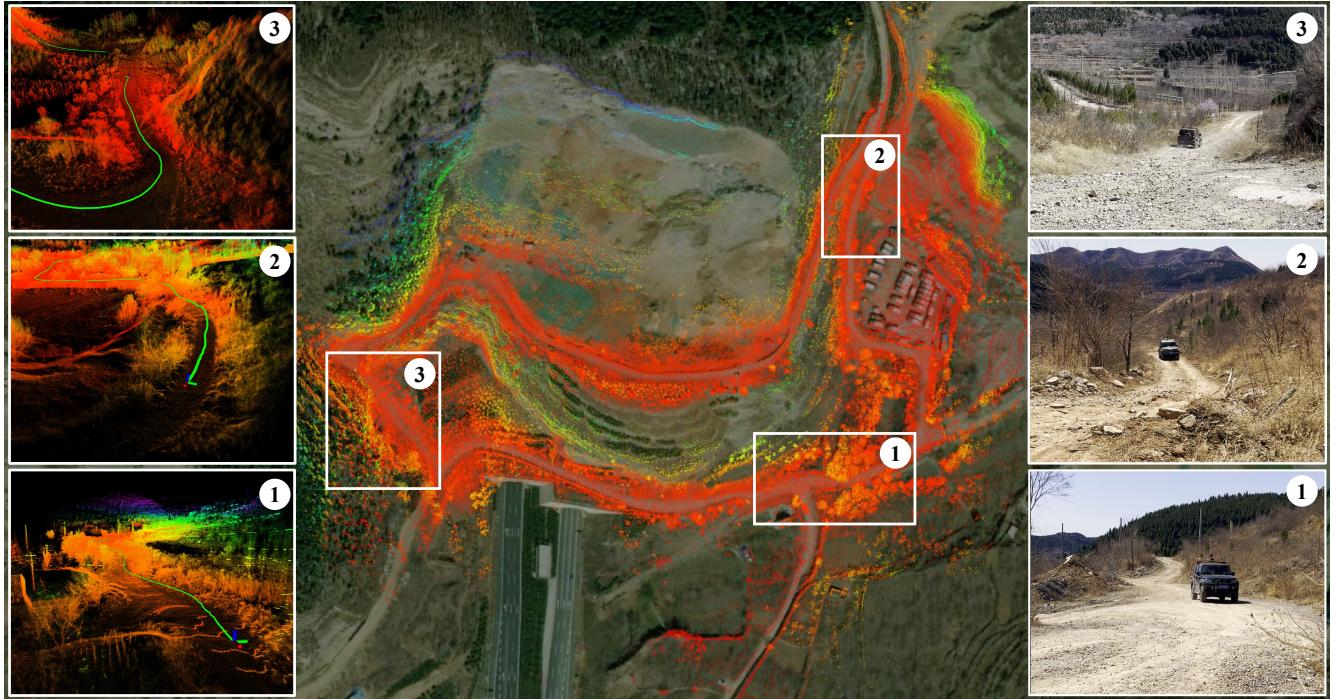


FIGURE 18 Point cloud map aligned with satellite map of **Offroad3** dataset. Three snapshots are captured in different driving scenes, which are sequenced by the top number.

optimization, CT-ICP solely relies on scan-to-map residuals to constrain the transformation for geometric alignment. This process involves frequent point cloud trimming operations and matching, which significantly prolongs the processing time. It's worth noting that ROLO has promising performance in harsh environments, such as off-roads. From the perspective of accuracy and mapping results, it is evident that ROLO's advantages are overwhelming compared to LeGO-LOAM, with a 70.7% improvement in localization accuracy relative to LeGO-LOAM. Moreover, the rough terrains are reconstructed finely with ROLO relying on precise localization.

Although ROLO has the best overall performance compared with the state-of-the-art, its accuracy of rotation estimation is extremely reliant on the state front propagation. Through numerous experiments, we find that the propagated results are easy to get bad solutions when the vehicle has uncertain motion behaviors or suffers emergency braking. ROLO is sensitive to these situations. Subsequently, we attempt to release these constraints for more accurate rotation estimation. Extracting prior environmental information, like slope and time-sequenced information could be expected to integrate in the framework.

7 | CONCLUSION AND FUTURE WORK

Toward pose drift and map distortion problem of LiDAR SLAM in uneven terrain with ground vehicle, we formulate a rotation-optimized LiDAR SLAM framework to achieve more precise pose estimation and mapping tasks. We first analysis the substantial reasons leading to the pose drift in the case of driving on rough terrain and find that the high-drift attitude estimation results in the final drift and distortion problems. Furthermore, the rotation and translation are decoupled by the developed state front propagation method. Subsequently, the spherical rotation registration and continuous-trajectory translation estimation method are used for more accurate pose estimation. We then integrate our method into an efficient SLAM framework. The proposed method is validated by various datasets in challenging scenarios and rough terrains. The final performance shows our method is equipped with satisfactory accuracy and fast process speed. In future work, attitude mutation aroused by the dynamic and momentary ground fluctuations will be focused on solving. Meanwhile, prior information like future pose prediction extracted from limited observation might be useful to improve the accuracy of attitude estimation.

AUTHOR CONTRIBUTIONS

This is an author contribution text. This is an author contribution text.

ACKNOWLEDGMENTS

FINANCIAL DISCLOSURE

None reported.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

References

- Banerjee, A., Dhillon, I. S., Ghosh, J., Sra, S., and Ridgeway, G. (2005). Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9).
- Chen, K., Lopez, B. T., Agha-mohammadi, A.-a., and Mehta, A. (2022a). Direct lidar odometry: Fast localization with dense point clouds. *IEEE Robotics and Automation Letters*, 7(2):2000–2007.
- Chen, S., Ma, H., Jiang, C., Zhou, B., Xue, W., Xiao, Z., and Li, Q. (2021). Ndt-loam: A real-time lidar odometry and mapping with weighted ndt and lfa. *IEEE Sensors Journal*, 22(4):3660–3671.
- Chen, W., Wang, Y., Chen, H., and Liu, Y. (2022b). Eil-slam: Depth-enhanced edge-based infrared-lidar slam. *Journal of Field Robotics*, 39(2):117–130.
- Chen, X., Läbe, T., Milioto, A., Röhling, T., Vysotska, O., Haag, A., Behley, J., and Stachniss, C. (2020). OverlapNet: Loop Closing for LiDAR-based SLAM. In *Proceedings of Robotics: Science and Systems (RSS)*.
- Chen, X., Milioto, A., Palazzolo, E., Giguere, P., Behley, J., and Stachniss, C. (2019). Suma++: Efficient lidar-based semantic slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4530–4537. IEEE.
- Chen, X., Wang, Y., Wang, C., Song, R., and Li, Y. (2022c). Low-drift lidar-only odometry and mapping for ugv in environments with non-level roads. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13174–13180. IEEE.
- Choi, S., Chae, H.-W., Jeung, Y., Kim, S., Cho, K., and Kim, T.-w. (2023). Fast and versatile feature-based lidar odometry via efficient local quadratic surface approximation. *IEEE Robotics and Automation Letters*, 8(2):640–647.
- Dellenbach, P., Deschaud, J.-E., Jacquet, B., and Goulette, F. (2022). Ct-icp: Real-time elastic lidar odometry with loop closure. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5580–5586. IEEE.
- Deng, J., Wu, Q., Chen, X., Xia, S., Sun, Z., Liu, G., Yu, W., and Pei, L. (2023). Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8218–8227.
- Ebadi, K., Chang, Y., Palieri, M., Stephens, A., Hatteland, A., Heiden, E., Thakur, A., Funabiki, N., Morrell, B., Wood, S., et al. (2020). Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 80–86. IEEE. <https://doi.org/10.1109/ICRA40945.2020.9197082>.
- Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Guo, H., Zhu, J., and Chen, Y. (2023). E-loam: Lidar odometry and mapping with expanded local structural information. *IEEE Transactions on Intelligent Vehicles*, 8(2):1911–1921.
- Guo, S., Rong, Z., Wang, S., and Wu, Y. (2022). A lidar slam with pca-based feature extraction and two-stage matching. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11.
- Jian, Z., Lu, Z., Zhou, X., Lan, B., Xiao, A., Wang, X., and Liang, B. (2022). Putn: A plane-fitting based uneven terrain navigation framework. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7160–7166. IEEE. <https://doi.org/10.1109/IROS47612.2022.9981038>.
- Koide, K., Miura, J., and Menegatti, E. (2019). A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, 16(2):1729881419841532.
- Koide, K., Yokozuka, M., Oishi, S., and Banno, A. (2021). Voxelized gicp for fast and accurate 3d point cloud registration. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11054–11059. IEEE.
- Latif, Y., Cadena, C., and Neira, J. (2014). Robust graph slam back-ends: A comparative analysis. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2683–2690. IEEE.
- Li, Q., Chen, S., Wang, C., Li, X., Wen, C., Cheng, M., and Li, J. (2019). Lo-net: Deep real-time lidar odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8473–8482.
- Li, X., Zhang, A., Liu, H., Fan, C., Liu, C., and Zheng, R. (2022). Gicp-loam: Lidar odometry and mapping with

- voxelized generalized iterative closest point. In *2022 China Automation Congress (CAC)*, pages 2103–2108. IEEE.
- Lin, J. and Zhang, F. (2020). Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3126–3131. IEEE.
- Pais, G. D., Ramalingam, S., Govindu, V. M., Nascimento, J. C., Chellappa, R., and Miraldo, P. (2020). 3dregnet: A deep neural network for 3d point registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7193–7203.
- Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. (2013). Comparing icp variants on real-world data sets: Open-source library and experimental protocol. *Autonomous robots*, 34:133–148.
- Ruan, J., Li, B., Wang, Y., and Sun, Y. (2023). Slamesh: Real-time lidar simultaneous localization and meshing. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3546–3552.
- Shan, T. and Englot, B. (2018). Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE. <https://doi.org/10.1109/IROS.2018.8594299>.
- Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., and Rus, D. (2020). Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5135–5142. IEEE.
- Wang, H., Wang, C., Chen, C.-L., and Xie, L. (2021). F-loam: Fast lidar odometry and mapping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4390–4396. IEEE.
- Wang, Z., Yang, L., Gao, F., and Wang, L. (2022a). Fevo-loam: Feature extraction and vertical optimized lidar odometry and mapping. *IEEE Robotics and Automation Letters*, 7(4):12086–12093.
- Wang, Z., Zhang, L., Shen, Y., and Zhou, Y. (2022b). D-liom: Tightly-coupled direct lidar-inertial odometry and mapping. *IEEE Transactions on Multimedia*.
- Xue, H., Fu, H., Xiao, L., Fan, Y., Zhao, D., and Dai, B. (2023). Traversability analysis for autonomous driving in complex environment: A lidar-based terrain modeling approach. *Journal of Field Robotics*, 40(7):1779–1803.
- Yang, H., Shi, J., and Carlone, L. (2020). Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333.
- Yuan, Z., Wang, Q., Cheng, K., Hao, T., and Yang, X. (2023). Sdv-loam: Semi-direct visual-lidar odometry and mapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18.
- Zhang, J. and Singh, S. (2014). Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and systems*, volume 2, pages 1–9. Berkeley, CA. <https://doi.org/10.15607/RSS.2014.X.007>.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

AUTHOR BIOGRAPHY

Author Name. Please check with the journal's author guidelines whether author biographies are required. They are usually only included for review-type articles, and typically require photos and brief biographies for each author.