

Article

Deskewed LiDAR Odometry for Quadruped Robots in Environments with Varying Elevation

Eunhui Han  and Heoncheol Lee 

Department of IT Convergence Engineering, Kumoh National Institute of Technology,
Gumi 39177, Republic of Korea; silverhee@kumoh.ac.kr

* Correspondence: hlee@kumoh.ac.kr; Tel.: +82-54-478-7458

Abstract

As robotics technology advances, quadruped robots have become capable of operating in complex environments with varying elevation, including ramps and level changes that are challenging for conventional wheeled platforms. While this terrain adaptability opens new opportunities for inspection, rescue, and exploration tasks, the repetitive impacts, frequent ground-contact transitions, and abrupt postural changes inherent to legged locomotion pose significant challenges for LiDAR odometry. High-frequency gait vibrations and abrupt attitude changes introduce intra-scan motion distortion that conventional single-twist deskewing cannot adequately suppress. In addition, sparse vertical geometric constraints in elevation-varying environments weaken Z-axis observability, allowing vertical drift to corrupt the horizontal pose estimate through Hessian coupling. To address these failure modes within a LiDAR-only framework, we propose a Piecewise-Constant Velocity deskewing scheme that partitions each scan into multiple temporal segments with safety clamping on vertical and attitude components, together with a two-stage ICP that decouples SE(3) optimization into horizontal (x , y , yaw) and vertical (z , $roll$, $pitch$) stages and applies observability-aware weighting in the vertical update. The proposed odometry front-end is evaluated on four real-world sequences collected with a Unitree Go2 quadruped robot equipped with a Velodyne VLP-16 LiDAR. Experimental results show consistently lower Absolute Pose Error (APE) than ICP, KISS-ICP, and F-LOAM across all sequences. Vertical drift suppression is most pronounced in the ramp-containing sequences, where baseline methods exhibit substantial Z-axis divergence.

Keywords: LiDAR odometry; quadruped robot; motion deskewing; iterative closest point; z-axis observability; point-cloud registration



Academic Editor: Shuo Shi

Received: 27 March 2026

Revised: 22 May 2026

Accepted: 29 May 2026

Published: 2 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

LiDAR odometry is a core front-end component for autonomous mobile robots, providing incremental pose estimation from sequential 3D point clouds without relying on external positioning infrastructure. Owing its robustness to illumination changes and its ability to provide accurate three-dimensional range measurements, LiDAR-based odometry has been widely adopted in robot navigation. In practical robotic systems, the accuracy of front-end odometry directly governs drift accumulation [1] and strongly affects downstream modules, including mapping and long-term navigation. In particular, accumulated drift can degrade localization reliability in environments with repetitive structures [2].

Quadruped robots have attracted growing interest for inspection, rescue, and exploration tasks because they can traverse unstructured terrain such as stairs, ramps, and level

changes that are difficult for wheeled platforms. However, physical traversability does not guarantee reliable pose estimation. During locomotion, quadrupeds experience repeated foot–ground impacts, gait-induced body oscillations, and abrupt attitude changes that severely undermine the spatiotemporal consistency of LiDAR measurements, violating the static-pose assumption commonly used during scan acquisition. These dynamics induce intra-scan motion distortion, degrade point correspondence quality, and destabilize front-end estimation, making registration failures and drift accumulation more likely than on wheeled platforms. Similar effects have also been reported on dynamically unstable platforms, where abrupt attitude changes cause unintended LiDAR observations and degrade scan matching stability [3].

These challenges are particularly critical for vertical registration and Z-axis stability. In environments with varying elevation, small front-end errors can propagate into *pitch/roll* bias and accumulated vertical inconsistency. This instability is further compounded by both sensor and environmental factors. Multi-channel LiDAR sensors with sparse vertical resolution inherently weaken Z-axis observability, while dominant structures in stair, ramp, and uneven-terrain environments often provide limited sensitivity to vertical displacement in point-to-plane registration. As a result, horizontal constraints dominate the optimization, whereas vertical components remain underconstrained, causing recurrent Z-axis drift.

Existing LiDAR odometry methods have been developed primarily for wheeled or handheld platforms. Feature-based methods such as LOAM [4] and F-LOAM [5] rely on edge and planar feature extraction, which can degrade under the high-frequency vibrations of legged locomotion. IMU-aided front-end methods such as FAST-LIO2 [6] enable direct intra-scan deskewing, but their performance depends on IMU quality and calibration. Moreover, the impulsive dynamics of quadruped locomotion can amplify IMU integration errors. LiDAR-only methods such as KISS-ICP [7] assume a constant-velocity motion model, which becomes unreliable under abrupt attitude changes. Furthermore, single-stage SE(3) optimization in these methods allows vertical instability to propagate into all pose components.

This paper proposes an improved LiDAR odometry method for quadruped robots operating in unstructured terrain with varying elevation. The main contributions are as follows:

- **Quadruped-Aware Deskewing:** We propose a Piecewise-Constant Velocity deskewing method tailored to quadruped high-frequency vibration and abrupt attitude changes, and introduce Z-axis motion clamping to suppress early-interval distortion.
- **Decoupled Registration for Z-Axis Drift Mitigation:** We design a two-stage ICP framework that decomposes SE(3) optimization into separate horizontal and vertical stages, preventing vertical instability from corrupting horizontal pose estimates.
- **Observability-Based Damping in Vertical Optimization:** We introduce point-to-plane constraints with observability-score-based damping in the vertical stage to enhance Z-axis pose estimation robustness in environments with elevation changes.

2. Related Work

LiDAR odometry estimates the 6-DoF pose of a sensor through registration between consecutively acquired 3D point clouds. The most widely adopted registration framework is the ICP (Iterative Closest Point) family, and recent studies differ in (i) residual formulation (point-to-point/point-to-plane), (ii) map representation (voxel/surfel/implicit surface), (iii) scan distortion correction (deskewing), and (iv) handling of degeneracy and observability. This work adopts a LiDAR-only design philosophy, targeting scan distortion and z-axis drift.

2.1. LiDAR Registration and Odometry

ICP constructs correspondences between two point clouds and iteratively estimates the rigid-body transformation that minimizes the defined objective function [8]. Convergence characteristics and noise sensitivity vary with the residual formulation and data association strategy [9]. LOAM [4] performs separate scan-to-scan odometry and scan-to-map registration using edge and planar features, and F-LOAM [5] retains this pipeline while reducing computational overhead. LeGO-LOAM [10] incorporates ground segmentation to improve efficiency on uneven terrain, though this segmentation step has been reported to under-segment in non-flat or cluttered scenes [11]. More broadly, the curvature thresholds used for feature discrimination in LOAM-family methods are sensitive to sensor resolution and scene geometry [12], which can reduce robustness under unstable sensing and geometric conditions. In the dense representation category, SuMa performs frame-to-model ICP using a surfel-based map with projective data association [13], IMLS-SLAM reduces drift via scan-to-model registration on an implicit IMLS surface [14], and MULS proposes a unified LiDAR odometry framework based on multi-metric linearized ICP, while further incorporating back-end pose graph optimization (PGO) [15]. These methods, however, depend on the quality of normal/surface estimation [13,14], and projective association can be vulnerable to abrupt viewpoint changes [13], requiring additional consideration in environments with frequent *roll/pitch* variation such as legged robots. KISS-ICP presents a LiDAR-only pipeline combining point-to-point ICP with adaptive correspondence thresholding, a robust kernel, constant-velocity (CV) deskewing, and voxel downsampling, eliminating data-dependent preprocessing stages [7]. Nevertheless, in quadruped locomotion, the approximation error of a single CV model can grow due to high-frequency vibrations and abrupt pose changes, and in environments with varying elevation, *z*-axis observability may weaken, causing vertical estimation errors to accumulate coupled with other DoF updates.

2.2. Scan Motion Distortion Correction (Deskewing)

Rotating LiDARs acquire each point within a sweep at a different timestamp; sensor motion during a sweep therefore introduces intra-scan motion distortion that degrades registration quality [4]. KISS-ICP demonstrates that a CV model alone can provide practical deskewing [7], but model mismatch accumulates when velocity or attitude changes abruptly within a sweep. CT-ICP integrates a continuous-time trajectory into the registration process to improve robustness to high-frequency motion [16], at the cost of increased optimization variables and computational complexity. IMU-coupled approaches (LIO-SAM [17], FAST-LIO2 [6]) enable direct intra-scan deskewing under high-dynamic conditions but depend structurally on IMU quality, time synchronization, and extrinsic calibration. Retaining the LiDAR-only philosophy [7], this work introduces Piecewise-Constant Velocity (PCV) deskewing, which partitions the sweep into time segments and approximates motion as constant within each segment, and combines it with *z*-axis motion clamping to suppress instantaneous vertical displacement induced by the gait cycle, thereby blocking the propagation of locomotion-induced distortion to registration quality.

2.3. LiDAR Odometry for Legged/Quadruped Robots

Quadruped robots are prone to registration degeneracy and distortion accumulation due to large *roll/pitch* variations, repetitive foot impacts, and discontinuous ground contact, leading to increased deskewing errors, degraded correspondence quality, and weakened *z*-axis constraints [18–20]. Ramezani et al. proposed a LiDAR odometry system for legged robots combining inertial-kinematic state-estimator-based scan accumulation, ICP registration, and learning-based loop closure [18]. VILENS by Wisth et al. tightly couples vision-IMU-LiDAR-leg odometry in a factor graph to maintain robustness under indi-

vidual sensor degeneracy, reporting mitigation of vertical drift accumulation through on-line velocity bias estimation [19]. Additional works including Leg-KILO [20], LIKO [21], LLIO [22], and Okawara et al. [23] propose tightly coupled LiDAR-IMU-kinematics/contact frameworks for robust estimation under high-dynamic gait conditions. While these approaches demonstrate that proprioceptive information can substantially reinforce observability constraints [20–24], they introduce increased sensor dependency and calibration burden [19,20], and constraint validity varies with contact state estimation and ground model quality [22,24]. This work targets the suppression of locomotion-induced scan distortion and z-axis instability directly in the front-end registration stage, while preserving a LiDAR-only structure without additional sensors.

Although LiDAR-inertial and LiDAR-inertial-kinematic methods provide strong robustness for legged robots, this work intentionally focuses on a LiDAR-only odometry front-end. The goal is not to replace tightly coupled multi-sensor systems, but to isolate and improve the robustness of LiDAR registration itself under quadruped locomotion. If IMU measurements were incorporated, the deskewing prior and attitude information would be externally provided by the inertial sensor, making it difficult to isolate whether the observed improvement comes from the proposed motion compensation and vertical stabilization strategies or from the additional inertial information. Moreover, IMU-aided methods introduce additional dependencies, including accurate time synchronization, LiDAR-IMU extrinsic calibration, and reliable bias/noise modeling. Therefore, this study compares against LiDAR-based baselines under the same input modality and evaluates how much the LiDAR-only front-end can be stabilized without relying on additional sensing modalities.

2.4. Degeneracy and Observability-Based Stabilization

Point-cloud registration becomes numerically degenerate when the local geometry fails to provide sufficient constraints along certain degrees of freedom (e.g., long corridors, repetitive structures). Nobili et al. proposed a framework for predicting alignment risk based on overlap and geometric constraint analysis [25]; Zhang et al. presented a remapping-based stabilization method that identifies degenerate directions via Hessian spectral analysis and restricts updates to well-constrained subspaces [26]; Zhen and Scherer proposed a localizability quantification model for geometrically deficient scenes [27]; Hinduja et al. extended direction-selective updates to graph optimization via partial factor construction [28]; Tuna et al. proposed X-ICP, which analyzes per-direction alignment strength from correspondences and enforces it as a constraint [29]; and Hatleskog and Alexis proposed a degeneracy-aware ICP that continuously attenuates per-direction updates based on Hessian-based degeneracy estimation [30]. On the mapping side, OR-LIM proposes an observability-aware design that evaluates per-frame observability and incorporates it into map updates [31]. Regarding robustification, Chebrolu et al. proposed an adaptive robust kernel that automatically adjusts to the residual distribution [32]. KISS-ICP combines adaptive thresholding and a robust kernel to suppress outliers [7], but when the environment geometry structurally fails to constrain certain DoFs, erroneous updates along weakly constrained directions can still occur [26,28,29]. This work focuses on the intermittent weakening of z-axis observability in environments with varying elevation encountered by legged robots. A two-stage ICP structure structurally blocks the propagation of vertical instability to horizontal pose estimation, and in the decoupled vertical stage, point-normal-geometry-based observability scoring is directly applied as damping weights for vertical updates.

3. Proposed Method

Quadruped locomotion introduces two compounding challenges for LiDAR odometry: intra-scan motion distortion caused by high-frequency gait vibrations, and intermittent Z-axis observability degradation on slopes and stairs that destabilizes the full 6-DoF pose estimate. As illustrated in Figure 1, the proposed front-end addresses these challenges through two targeted modifications. In the motion compensation stage, a Piecewise-Constant Velocity deskewing scheme with Z-axis motion clamping suppresses locomotion-induced scan distortion before registration. In the registration stage, SE(3) optimization is restructured into a horizontal solver (x, y, yaw) and a vertical solver ($z, roll, pitch$) with observability-aware weighting, preventing vertical instability from contaminating horizontal pose estimation.

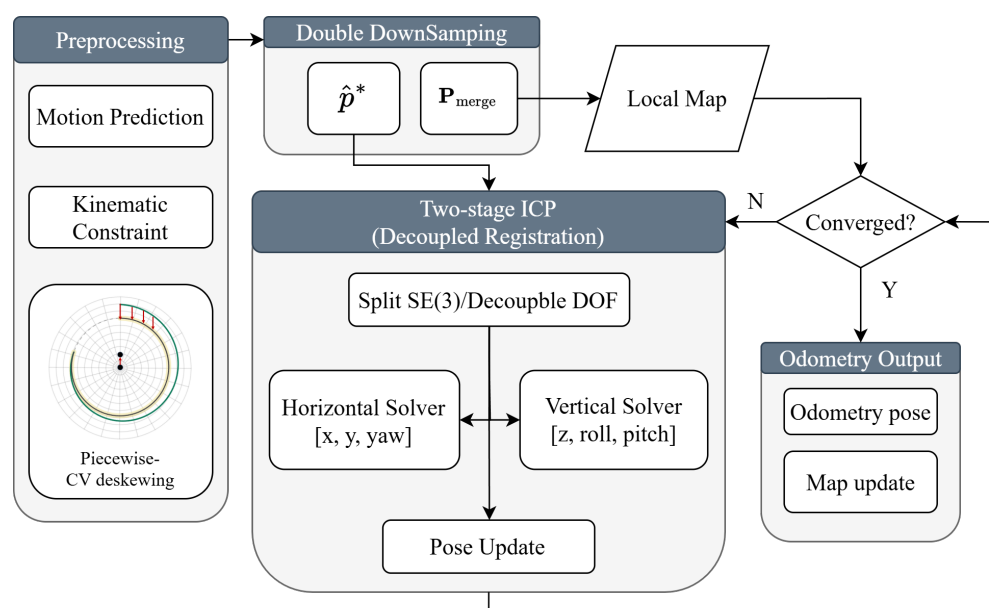


Figure 1. Overview of the proposed LiDAR odometry front-end pipeline.

Figure 2 illustrates the motion components and residual formulations used in the proposed two-stage registration. As shown in Figure 2a, quadruped motion is represented by translation along x, y , and z and rotation in $roll, pitch$, and yaw . Based on this representation, the proposed method separates the pose update into horizontal and vertical components. Figure 2b shows that the horizontal stage uses point-to-point residuals to estimate x, y , and yaw , whereas the vertical stage uses point-to-plane residuals with local normals estimated from neighboring map points to refine $z, roll$, and $pitch$. The following subsections formalize this decomposition and describe each component in detail.

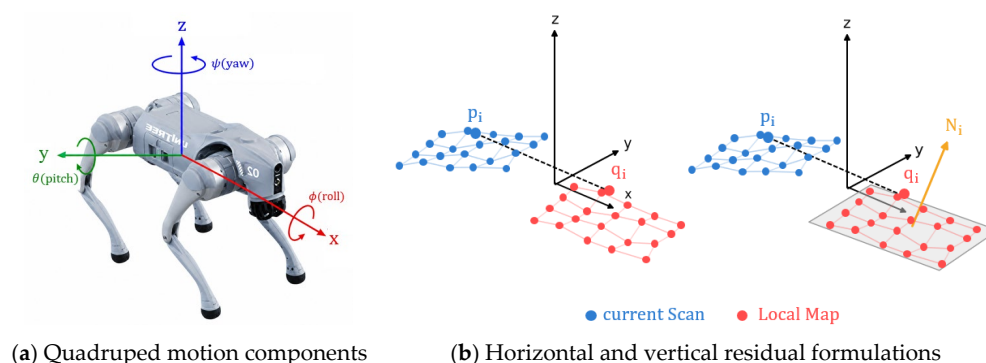


Figure 2. Conceptual illustration of quadruped motion components and two-stage ICP residuals.

3.1. Problem Formulation and State Decomposition

A single LiDAR scan is defined as

$$\mathcal{P} = \{(\mathbf{p}_i, \tau_i)\}_{i=1}^N \quad (1)$$

where $\mathbf{p}_i \in \mathbb{R}^3$ is the i -th point and $\tau_i \in [0, 1]$ is the normalized acquisition time within the scan. The pose of the current frame is represented as $T \in SE(3)$, and its incremental update is written as

$$\delta\zeta = [\delta x \quad \delta y \quad \delta z \quad \delta\phi \quad \delta\theta \quad \delta\psi]^\top. \quad (2)$$

Here, $\delta\zeta$ is treated as a twist-like incremental pose update in $SE(3)$, where $(\delta x, \delta y, \delta z)$ represent translational increments and $(\delta\phi, \delta\theta, \delta\psi)$ represent *roll*, *pitch*, and *yaw* increments, respectively. Throughout this paper, the variable ordering of the full state vector is fixed as $[\delta x, \delta y, \delta z, \delta\phi, \delta\theta, \delta\psi]^\top$, and all subsequent Jacobian, Hessian, and state update expressions follow this ordering. In conventional coupled 6-DoF registration, this full state is estimated simultaneously in a single optimization problem. However, on slopes and level changes, the observability of z , *roll*, and *pitch* may weaken or change abruptly, so the instability of the vertical state can propagate to the horizontal state through the coupling terms of the Hessian.

To make this more explicit, the horizontal and vertical states are defined from the full state vector as

$$\delta\zeta_H = [\delta x \quad \delta y \quad \delta\psi]^\top, \quad \delta\zeta_V = [\delta z \quad \delta\phi \quad \delta\theta]^\top, \quad (3)$$

where $\delta\zeta_H$ and $\delta\zeta_V$ are partial state vectors that extract only the horizontal components (x, y, ψ) and vertical components (z, ϕ, θ) from $\delta\zeta$, respectively. Viewing the state in horizontal–vertical order to clarify the block structure, the coupled normal equation can be expressed conceptually as

$$\begin{bmatrix} H_{HH} & H_{HV} \\ H_{VH} & H_{VV} \end{bmatrix} \begin{bmatrix} \delta\zeta_H \\ \delta\zeta_V \end{bmatrix} = - \begin{bmatrix} g_H \\ g_V \end{bmatrix}, \quad (4)$$

where H_{HH} and H_{VV} are the self-coupling terms for the horizontal and vertical states, respectively, and H_{HV} , H_{VH} denote the coupling between the two states. When vertical observability decreases, H_{VV} can become ill-conditioned, and as a result, the horizontal state is also destabilized through the coupling terms H_{HV} and H_{VH} . That is, insufficient information in the vertical state leads not only to degraded estimation of z , *roll*, and *pitch*, but also to reduced stability of the full pose estimate including x , y , and *yaw*. It should be noted that this decomposition is not based on an assumption that the horizontal and vertical motions are physically independent or weakly coupled. On sloped terrain, horizontal translation, vertical displacement, and attitude changes are geometrically related. However, in LiDAR-only quadruped odometry, the vertical components can become intermittently weakly constrained due to sparse vertical LiDAR resolution, gait-induced motion, and elevation-varying geometry. Under such conditions, a joint $SE(3)$ update may allow uncertainty in the vertical subspace to affect the horizontal update through the coupling blocks of the Hessian. Therefore, the proposed decomposition is introduced as a stabilization-oriented separation of update variables, aiming to regulate the propagation of poorly constrained vertical corrections into the horizontal pose estimate.

Rather than solving this coupled structure directly, the proposed method adopts a decoupled optimization structure in which the horizontal state $\delta\zeta_H$ is estimated first in a stable manner, and the vertical state $\delta\zeta_V$ is then corrected separately. This state decomposition serves as the direct design basis for the decoupled two-stage ICP proposed in

Section 3.3, and is realized as a sequential estimation structure that first secures horizontal pose stability and then conditionally refines the vertical pose. This decomposition corresponds to the horizontal and vertical registration branches of the two-stage ICP module shown in Figure 1.

3.2. Quadrupedal-Motion-Aware Piecewise-Constant Velocity Deskewing

A quadrupedal robot undergoes repeated foot contacts and liftoffs throughout its gait cycle, generating high-frequency vibrations and instantaneous *pitch/roll* changes in the body frame. These motions can induce substantial intra-scan distortion even within a single LiDAR scan acquisition window, making it difficult to accurately compensate for them under a single constant-twist assumption. To address this, the proposed method partitions a single scan into multiple short temporal segments. Let K denote the number of such segments. In our implementation, the Velodyne VLP-16 LiDAR was operated at approximately 10 Hz, and the odometry pose was updated once per incoming scan. We set $K = 4$ for PCV deskewing, corresponding to an approximate segment duration of 0.025 s per scan segment.

Let the predicted poses at the start and end of the scan be $T_s, T_f \in SE(3)$, respectively. The relative motion between the two poses is defined as

$$\Delta T = T_s^{-1}T_f, \quad \xi = \log(\Delta T), \quad (5)$$

where $\xi \in \mathfrak{se}(3)$ is the twist representation for the full scan interval. However, in quadrupedal environments, the raw motion prior may contain brief vertical spikes or *roll/pitch* transients; using it directly could amplify abnormal distortion during deskewing. Therefore, a corrected twist $\tilde{\xi}$ is constructed by clamping the vertical translation and attitude components:

$$\tilde{\xi} = \begin{bmatrix} \xi_x & \xi_y & \text{clip}(\xi_z; -\zeta_z, \zeta_z) & \text{clip}(\xi_\phi; -\zeta_r, \zeta_r) & \text{clip}(\xi_\theta; -\zeta_p, \zeta_p) & \xi_\psi \end{bmatrix}^T, \quad (6)$$

where ζ_z , ζ_r , and ζ_p denote the allowable limits for vertical translation, *roll*, and *pitch*, respectively. Importantly, x , y , and *yaw* are retained because they reflect the actual locomotion direction during walking, whereas only z , *roll*, and *pitch*, which are more susceptible to gait-induced oscillations and abrupt attitude changes in quadruped locomotion, are restricted.

In this work, the clamping thresholds were set to $\zeta_z = 0.5$ m, $\zeta_r = 0.2$ rad, and $\zeta_p = 0.2$ rad, and the same values were used across all evaluated sequences without sequence-specific tuning. These values were selected in a platform-aware manner by considering the LiDAR scan period, body-height oscillation, and attitude variation of the Unitree Go2 during walking and ramp traversal. However, they were intentionally set as loose safety bounds rather than tight physical motion limits. These values are used only to bound the LiDAR-only motion prior during deskewing, not to constrain the final pose estimated by registration. Therefore, they do not directly limit the actual terrain-induced elevation change estimated by ICP. The purpose of the clamping operation is not to suppress normal gait-induced vertical motion or slope-induced attitude changes, but to prevent extreme and physically implausible spikes in the LiDAR-only motion prior from being propagated into the deskewing process.

For other robots or gaits, the same principle can be applied by selecting the thresholds according to the expected intra-scan vertical and angular motion range of the target platform. As a practical guideline, ζ_z can be initially set based on the maximum expected vertical displacement within one LiDAR scan period, e.g., the product of the platform's maximum climbing speed and the LiDAR scan period, while ζ_r and ζ_p can be selected

according to the maximum expected *roll* and *pitch* variations within a single scan. These bounds should be large enough to preserve genuine terrain-induced motion, but small enough to reject clearly abnormal motion-prior spikes. The corrected end pose is then reconstructed as

$$\tilde{T}_f = T_s \exp\left(\tilde{\xi}\right). \quad (7)$$

The scan is then partitioned into multiple temporal segments, and the pose at each segment boundary T_k is computed by interpolating between the start pose T_s and the corrected end pose $\tilde{T}_f$ using SLERP for rotation and linear interpolation for translation:

$$T_k = \mathcal{I}\left(T_s, \tilde{T}_f, \frac{k}{K}\right), \quad k = 0, \dots, K, \quad (8)$$

where $\mathcal{I}(\cdot)$ denotes the combined rotation-translation interpolation operator.

$$\xi_k = \log\left(T_k^{-1} T_{k+1}\right), \quad k = 0, \dots, K-1. \quad (9)$$

Rather than using a single twist for the entire scan, a separate local twist is permitted for each segment, enabling a better approximation of the temporal variation of intra-scan motion. Let $\tau_i \in [0, 1]$ denote the normalized acquisition time of the i -th point, obtained from its LiDAR timestamp, and let $\tau_k = k/K$ denote the normalized boundary time of the k -th temporal segment. For each point, the segment index k is selected such that $\tau_k \leq \tau_i < \tau_{k+1}$. The intra-segment interpolation coefficient is then computed as

$$u_i = \frac{\tau_i - \tau_k}{\tau_{k+1} - \tau_k}, \quad \tau_k = \frac{k}{K}. \quad (10)$$

The sensor pose at the corresponding time instant is computed as

$$T(\tau_i) = T_k \exp(u_i \xi_k). \quad (11)$$

The mid-scan pose is used as the reference frame, and all points are corrected to this common coordinate system:

$$T_{\text{mid}} = T(0.5), \quad p_i^* = T_{\text{mid}}^{-1} T(\tau_i) p_i. \quad (12)$$

The mid-scan instant is chosen as the reference because, compared to using the start or end of the scan, it distributes the correction deviations symmetrically along the time axis, thereby reducing the maximum per-point correction error. Consequently, even in the presence of prediction error, the accumulation of deskewing error due to reference-time bias is mitigated.

In summary, the proposed deskewing approximates the nonlinear intra-scan motion on a per-segment basis while simultaneously suppressing instantaneous vertical spikes and abrupt *roll/pitch* changes—which occur frequently during quadrupedal locomotion—through clamping. As shown in Figure 1, the resulting corrected point cloud serves as the input to the decoupled two-stage registration proposed in Section 3.3.

3.3. Horizontal–Vertical Decoupled Two-Stage Registration

Taking the point cloud corrected in Section 3.2 as input, the proposed method performs two registration stages sequentially at each outer ICP iteration. First, horizontal registration based on point-to-point correspondences estimates the planar motion components (x, y, ψ) . Then, vertical registration based on point-to-plane correspondences refines the elevation-related components (z, ϕ, θ) . This section describes the residual, Jacobian, and state update

structure of each stage, while the correspondence weighting and update stabilization strategy for the vertical stage are detailed in Section 3.4.

Point-to-point residuals are used in the horizontal stage because the horizontal motion components (x, y, ψ) are generally well constrained by the spatial distribution of surrounding LiDAR points. This choice preserves the robustness and simplicity of the KISS-ICP baseline and avoids relying on potentially noisy normal estimates during the primary horizontal alignment. Although point-to-plane residuals may provide stronger directional constraints, their performance depends heavily on the quality of local normal estimation. Under quadruped locomotion, gait-induced vibration, sparse LiDAR measurements, and residual motion distortion can make local normals noisy or inconsistent, which may introduce biased constraints into horizontal estimation.

In contrast, the vertical components (z, ϕ, θ) are more susceptible to weak observability in elevation-varying environments. Therefore, point-to-plane residuals are introduced only in the vertical refinement stage, where surface normals provide useful directional constraints for suppressing Z-axis drift and stabilizing attitude estimation.

3.3.1. Horizontal Stage: Planar Motion Estimation via Point-to-Point Constraints

The horizontal stage estimates only x, y , and yaw using point-to-point residuals. Letting s_i and q_i denote the current source point and its corresponding target point, respectively, the residual is defined as

$$\mathbf{r}_i^H = \mathbf{s}_i - \mathbf{q}_i. \quad (13)$$

The full 6-DoF Jacobian is given by

$$\mathbf{J}_i^p = \begin{bmatrix} \mathbf{I}_{3 \times 3} & -[\mathbf{s}_i]_{\times} \end{bmatrix} \in \mathbb{R}^{3 \times 6}, \quad (14)$$

where $[\mathbf{s}_i]_{\times}$ denotes the skew-symmetric matrix. Since the horizontal stage estimates only $\delta\zeta_H = [\delta x, \delta y, \delta\psi]^\top$, only the columns of the full Jacobian corresponding to x, y , and yaw are used. Defining the state selection matrix as

$$\mathbf{S}_H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{6 \times 3}, \quad (15)$$

the reduced Jacobian is expressed as

$$\bar{\mathbf{J}}_i^H = \mathbf{J}_i^p \mathbf{S}_H \in \mathbb{R}^{3 \times 3}. \quad (16)$$

Conceptually, the normal equation of the horizontal stage is formulated over $\delta\zeta_H \in \mathbb{R}^3$; in the implementation, this is equivalently realized as a masked solve that sets the inactive components of the full 6-DoF update to zero. Written in full-state form, the horizontal-stage update is

$$\hat{\delta\zeta}_H = [\delta x \quad \delta y \quad 0 \quad 0 \quad 0 \quad \delta\psi]^\top. \quad (17)$$

The Hessian and gradient of the horizontal stage are computed as

$$\mathbf{H}_H = \sum_i w_i^H \bar{\mathbf{J}}_i^{H\top} \bar{\mathbf{J}}_i^H, \mathbf{g}_H = \sum_i w_i^H \bar{\mathbf{J}}_i^{H\top} \mathbf{r}_i^H, \quad (18)$$

where w_i^H is a squared Lorentzian robust weight defined by the residual magnitude, formulated as

$$w_i^H = \frac{\kappa_H^2}{\left(\kappa_H + \|\mathbf{r}_i^H\|_2\right)^2}, \quad (19)$$

where κ_H is a robust kernel scale parameter that controls the attenuation of large-residual correspondences. This weight monotonically decreases with residual magnitude, suppressing the influence of large-residual correspondences while preserving the contribution of well-matched ones.

To prevent convergence failure due to insufficient correspondences in the early iterations, the horizontal stage applies an iteration-dependent correspondence distance. For a base maximum correspondence distance $d_{\max}$, the search radius at the t -th outer ICP iteration is defined as

$$d_{\max}(t) = \begin{cases} d_{\text{relaxed}}, & t \in \mathcal{T}_{\text{early}} \\ d_{\max}, & \text{otherwise} \end{cases} \quad (20)$$

where d_{relaxed} denotes a temporarily relaxed correspondence distance, $d_{\max}$ is the nominal maximum correspondence distance, and $\mathcal{T}_{\text{early}}$ denotes the set of early ICP iterations. Afterward, the search radius returns to the nominal value to suppress excessive false correspondences. The update $\delta\hat{\xi}_H$ computed in the horizontal stage is applied to the current estimated pose, aligning only the planar motion while keeping vertical translation and *roll/pitch* fixed.

3.3.2. Vertical Stage: Attitude Refinement via Point-to-Plane Constraints

After the horizontal state has been corrected, the vertical state (z, ϕ, θ) is refined in a separate stage. Because the vertical direction is difficult to observe stably from point-to-point residuals alone, point-to-plane residuals are used to exploit the structural geometry of ground surfaces and sloped structures in elevation-varying environments.

First, the local surface normal $\mathbf{n}_i$ is estimated around each source point s_i . Neighboring points are collected from adjacent voxels in the target map, and the mean $\bar{\mathbf{p}}_i$ and covariance matrix of those points are computed:

$$\mathbf{C}_i = \frac{1}{|\mathcal{N}_i|} \sum_{\mathbf{p} \in \mathcal{N}_i} (\mathbf{p} - \bar{\mathbf{p}}_i)(\mathbf{p} - \bar{\mathbf{p}}_i)^\top, \quad (21)$$

where $\mathcal{N}_i$ is the local neighborhood of s_i . The eigenvector corresponding to the smallest eigenvalue of the covariance matrix is used as the normal:

$$\mathbf{n}_i = \mathbf{v}_{\min}(\mathbf{C}_i). \quad (22)$$

Not all points are valid for vertical state estimation. To avoid unstable vertical updates, a point is admitted as a vertical constraint only when its local neighborhood, normal orientation, and range satisfy predefined reliability conditions.

The point-to-plane residual for a selected correspondence is defined as

$$r_i^V = \mathbf{n}_i^\top (\mathbf{s}_i - \mathbf{q}_i), \quad (23)$$

where $\mathbf{q}_i$ is the nearest target point selected within the local voxel neighborhood. The full 6-DoF Jacobian is

$$\mathbf{J}_i^V = \begin{bmatrix} \mathbf{n}_i^\top & \mathbf{n}_i^\top (-[\mathbf{s}_i]_\times) \end{bmatrix} \in \mathbb{R}^{1 \times 6}. \quad (24)$$

Only the columns corresponding to z , $roll$, and $pitch$ are used in the vertical stage. Defining the state selection matrix as

$$S_V = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{6 \times 3}, \quad (25)$$

the reduced Jacobian is

$$\bar{J}_i^V = J_i^V S_V \in \mathbb{R}^{1 \times 3}. \quad (26)$$

The normal equation of the vertical stage is therefore formulated over $\delta \zeta_V = [\delta z, \delta \phi, \delta \theta]^\top \in \mathbb{R}^3$; the definition of the correspondence weight w_i^V and the explicit form of the resulting Hessian and gradient are given in Section 3.4. Written in full-state form, the vertical-stage update is

$$\delta \hat{\zeta}_V = \begin{bmatrix} 0 & 0 & \delta z & \delta \phi & \delta \theta & 0 \end{bmatrix}^\top. \quad (27)$$

That is, x , y , and yaw—already aligned in the horizontal stage—are preserved, while only vertical translation and $roll/pitch$ are refined to conform to the terrain geometry. In the implementation, the vertical stage may be iterated multiple times within each outer ICP iteration to improve convergence of the vertical state.

Additionally, the vertical stage applies a per-correspondence reliability weight and a bounded update strategy to suppress excessive vertical corrections.

3.4. Observability-Aware Vertical State Stabilization

This stabilization process corresponds to the observability-aware weighting and bounded vertical update components in the two-stage ICP module of Figure 1. The quality of vertical state estimation depends not simply on the magnitude of residuals, but on how well each correspondence actually constrains z , $roll$, and $pitch$. For instance, a slope point observed at close range can strongly explain the vertical state, whereas a sparse point at long range or a nearly vertical wall point carries significantly less information about the vertical state. Treating all correspondences equally therefore causes the vertical optimization to easily become unstable, unable to distinguish between informative terrain observations and unreliable ones.

To address this, the proposed method formalizes the geometric validity of each correspondence as an observability score and combines it with a residual-based robust weight to form the final weight for the vertical stage. First, the distance-based observability term is defined as

$$\gamma(d_i) = \begin{cases} 1, & d_i < d_0, \\ \frac{1}{1 + \beta(d_i - d_0)^2}, & d_i \geq d_0, \end{cases} \quad (28)$$

where $d_i = \|s_i - o\|$ is the distance from the sensor origin o to source point s_i . This function maintains maximum confidence for near-range points while progressively reducing the contribution of far-range points as distance increases, where d_0 and β control the transition point and decay rate of the distance-based observability term, respectively.

Next, to reflect how well the surface orientation constrains the vertical state, the normal-based observability term is defined as

$$\eta(n_i) = |n_{i,z}|. \quad (29)$$

A larger z -component of the normal indicates that the surface is closer to horizontal (ground or slope), providing more meaningful information for estimating z , *roll*, and *pitch*. Conversely, when $|n_{i,z}|$ is small—as for a nearly vertical wall—the surface poorly constrains the vertical state, and its influence is accordingly reduced. The proposed observability score is designed as a practical geometry-based surrogate indicator rather than a formal probabilistic observability metric. Although a Fisher-information-based formulation could provide a stronger theoretical foundation, the proposed distance- and normal-based terms provide a computationally efficient approximation for evaluating the reliability of vertical point-to-plane constraints. This design is consistent with the geometric and optimization-structure-based reasoning used in existing degeneracy- and localizability-aware registration methods, which identify weakly constrained directions based on local geometric structure, Hessian eigenspace, Jacobian contribution, or surface-normal alignment.

Combining these two factors, the observability score of correspondence i is defined as

$$\Omega_i = \gamma(d_i) \eta(n_i). \quad (30)$$

However, high observability does not justify blindly incorporating residual outliers. Therefore, a robust term ρ_i based on residual magnitude is additionally introduced. In particular, near-range points may carry important evidence of actual vertical change even when they produce large residuals upon entering a steep slope; rather than applying a uniform continuous kernel that would attenuate all such responses, the proposed piecewise design aggressively follows near-range observations and suppresses only clear mismatches. Accordingly, ρ_i is defined as

$$\rho_i = \begin{cases} 1, & d_i < d_0 \text{ and } |r_i^V| \leq \epsilon_0, \\ \epsilon_s, & d_i < d_0 \text{ and } |r_i^V| > \epsilon_0, \\ \frac{\kappa_V^2}{\kappa_V^2 + (r_i^V)^2}, & d_i \geq d_0, \end{cases} \quad (31)$$

where r_i^V is the point-to-plane residual defined in Section 3.3.2, and ϵ_0 , ϵ_s , and κ_V denote the residual gating threshold, the minimum suppression weight, and the robust kernel scale parameter, respectively. The discontinuous transition in the near-range regime is intentional: comparatively large residuals that arise near a steep slope or stair entry are accepted as is so that the vertical state can respond rapidly to terrain change, while only sufficiently large residuals in the near-range regime are immediately suppressed to their minimum weight ϵ_s to block the influence of clear mismatches. In the far-range regime, where measurement uncertainty is relatively high, a Cauchy-type continuous kernel is used to progressively reduce the weight as the residual grows. To prevent this correspondence-level discontinuity from causing unstable state updates, the trust-region bound introduced in Equation (34) limits the norm of the vertical update vector. Thus, the robust weight and the trust-region bound work together as a two-level stabilization mechanism: correspondence-level outlier suppression and state-level update limiting.

The final correspondence weight for the vertical stage is defined as

$$w_i^V = \rho_i \Omega_i. \quad (32)$$

Using the reduced Jacobian $\bar{J}_i^V$ defined in Section 3.3.2, the Hessian and gradient of the vertical stage are then computed as

$$H_V = \sum_i w_i^V \bar{J}_i^{V\top} \bar{J}_i^V, g_V = \sum_i w_i^V \bar{J}_i^{V\top} r_i^V. \quad (33)$$

Correspondences with low observability scores automatically contribute less to the Hessian, so the vertical update is shaped primarily by geometrically informative terrain structures. In this sense, the observability score in the proposed method is not merely a quality indicator but serves to regulate the effective information content of the optimization.

In addition, a trust-region-based update limit is introduced to prevent excessive vertical corrections. Let $\tilde{\delta}_{\zeta_V}$ denote the vertical update obtained by solving the normal equation; the final update $\hat{\delta}_{\zeta_V}$ is determined as

$$\hat{\delta}_{\zeta_V} = \begin{cases} \tilde{\delta}_{\zeta_V}, & \|\tilde{\delta}_{\zeta_V}\|_2 \leq \zeta_V, \\ \zeta_V \frac{\tilde{\delta}_{\zeta_V}}{\|\tilde{\delta}_{\zeta_V}\|_2}, & \|\tilde{\delta}_{\zeta_V}\|_2 > \zeta_V. \end{cases} \quad (34)$$

When the constraint is violated, rather than discarding the update or clipping individual components, the direction of the full vertical update vector is preserved and only its norm is normalized to ζ_V , where ζ_V denotes the trust-region bound for the vertical state update. By directly limiting the magnitude of the $se(3)$ update vector, this suppresses excessive correction in a single iteration even when a momentarily large residual forms at an abrupt elevation transition or steep slope entry. In other words, while observability-aware weighting regulates information reliability at the correspondence level, the trust region guarantees stability at the state update level.

These weighting and trust-region constraints are recomputed and applied at every vertical sub-iteration described in Section 3.3.2. Consequently, the vertical stage operates not by performing a large correction at once, but by iteratively accumulating bounded updates while prioritizing correspondences with high observability. This is advantageous for gradually aligning the vertical state while preserving the horizontal pose in environments with varying elevation.

By combining correspondence-level weighting with state-level update limiting, the proposed method can handle the substantially terrain-dependent observability of the vertical state in a more stable manner. The following section summarizes the complete estimation procedure incorporating the deskewing, state decomposition, and vertical stabilization strategies described above.

3.5. Complete Estimation Procedure

The complete estimation procedure of the proposed front-end is summarized in Algorithm 1 and illustrated in Figure 1. The core of the proposed front-end lies in integrating the deskewing, state decomposition, and vertical stabilization strategies described in the preceding sections into a single sequential estimation procedure.

First, Piecewise-Constant Velocity deskewing is performed using a motion prior constructed from the predicted start and end poses of the current scan. Z-axis motion clamping is applied during this process to suppress abnormal vertical oscillations and abrupt *roll/pitch* changes from being excessively reflected in the initial correction. The purpose of this stage is not merely to reduce scan distortion but to stabilize the input to registration itself, so that subsequent correspondence formation can take place on a more geometrically consistent foundation.

Because the deskewed point cloud is expressed in the mid-scan reference frame, the initial registration estimate T_0 is set to the predicted mid-scan pose derived from the same motion prior used for deskewing. That is, T_0 is the initial alignment pose consistent with the reference frame of Section 3.2, and from an algorithmic perspective it represents the prior pose that ICP is tasked with correcting. ICP then operates by accumulating an incremental correction T_{icp} on top of this initial estimate, and the final pose is recovered as $T_k = T_{icp} T_0$.

Algorithm 1. Proposed Front-End Estimation Procedure

Input	• Current scan $\mathcal{P} = \{(p_i, \tau_i)\}_{i=1}^{n_i}$ • Predicted start pose T_s and end pose T_f • Local map $\mathcal{M}$
Output	• Updated pose T_k • Updated local map $\mathcal{M}$

1. Compute relative motion: $\Delta T \leftarrow T_s^{-1} T_f$, $\xi \leftarrow \log(\Delta T)$
2. Construct corrected twist by clamping vertical translation and attitude components: $\tilde{\xi} \leftarrow \text{clamp}(\xi)$
3. Reconstruct corrected end pose: $\tilde{T}_f \leftarrow T_s \exp(\tilde{\xi})$
4. Partition the scan into K temporal segments.
5. Interpolate segment boundary poses:
 $T_k \leftarrow \mathcal{I}(T_s, T_f, \frac{k}{K})$, $k = 0, \dots, K$
6. Compute local twist for each segment:
 $\xi_k \leftarrow \log(T_k^{-1} T_{k+1})$, $k = 0, \dots, K-1$
7. For each point p_i , compute its deskewed position p_i^* using the corresponding segment pose interpolation.
8. Set the mid-scan pose as the initial registration estimate: $T_0 \leftarrow T(0.5)$
9. Initialize ICP correction: $T_{icp} \leftarrow I$
10. for each outer ICP iteration t do
11. if $t \in \mathcal{T}_{early}$ then
12. $d_{corr}(t) \leftarrow d_{relaxed}$
13. else
14. $d_{corr}(t) \leftarrow d_{max}$
15. end if
16. Find correspondences for the deskewed scan using $d_{corr}(t)$.
17. Estimate horizontal update $\delta \xi_H$ using point-to-point residuals over (x, y, yaw) .
18. Apply the horizontal update: $T_{icp} \leftarrow \exp(\delta \tilde{\xi}_H) \cdot T_{icp}$
19. for $u = 1, \dots, U_V$ do
20. Estimate local normals for candidate correspondences.
21. Filter valid vertical correspondences satisfying:
 $|\mathcal{N}_i| \geq N_{min}$, $|n_{i,z}| \geq \tau_n$, $\|s_i - o\| \leq r_{max}$
22. Compute observability-aware vertical weights w_i^V
23. Estimate vertical update $\delta \xi_V$ using point-to-plane residuals over $(z, \text{roll}, \text{pitch})$.
24. if $\|\delta \xi_V\|_2 > \zeta_V$ then
25. $\delta \xi_V \leftarrow \frac{\zeta_V \cdot \delta \xi_V}{\|\delta \xi_V\|_2}$
26. end if
27. Apply the vertical update: $T_{icp} \leftarrow \exp(\delta \tilde{\xi}_V) \cdot T_{icp}$
28. end for
29. end for
30. Recover final pose: $T_k \leftarrow T_{icp} \cdot T_0$
31. Integrate the aligned scan into the local map $\mathcal{M}$.
32. Return T_k and updated $\mathcal{M}$.

Outer ICP iterations are then performed on the deskewed point cloud. In each iteration, the horizontal stage first executes, aligning x , y , and yaw via point-to-point constraints. During the early ICP iterations, the correspondence distance is temporarily relaxed to prevent a shortage of correspondences caused by abrupt elevation changes or initial deskewing errors. Once the horizontal stage has primarily aligned the planar motion of the current frame, the vertical stage refines only z , $roll$, and $pitch$ via point-to-plane constraints while holding the already-aligned horizontal state fixed. The geometric validity of each correspondence is reflected through observability-aware weighting, and the computed vertical update is applied after passing through a trust-region limit. The vertical stage may also be iterated multiple times within a single outer iteration to progressively correct the vertical pose, which typically converges more slowly than the horizontal pose in environments with varying elevation. As a result, horizontal registration first stabilizes the reference coordinate frame, and vertical registration then iteratively accumulates bounded updates on top of it, allowing the odometry front-end to respond without unstable vertical observations contaminating the horizontal trajectory.

Upon termination of the outer ICP iterations, the aligned scan is integrated into the local map and the updated pose is used as the motion prior and initial registration estimate for the next frame. Each frame is processed through a fixed sequence in which motion compensation first stabilizes the geometric input, horizontal registration then establishes a reliable planar reference, and vertical refinement subsequently adjusts height and attitude within that reference before the aligned scan is incorporated into the map.

4. Experiments

This section evaluates the proposed LiDAR odometry method both quantitatively and qualitatively. We first describe the datasets, compared methods, implementation details, and evaluation protocol. We then present the main comparison against representative baselines, followed by qualitative trajectory analysis.

4.1. Experimental Setup

4.1.1. Datasets and Sequences

We evaluated the proposed method on four sequences. Sequence 1 is the Playground sequence from the public S3E dataset [33], acquired using a wheeled robot platform with a Velodyne VLP-16 Puck LiDAR (Velodyne Lidar Inc., San Jose, CA, USA). Sequences 2–4 were collected using a Unitree Go2 quadruped robot equipped with a Velodyne VLP-16 LiDAR. The experimental platform and sensor configuration are shown in Figure 3. Sequence 2 (DIGI) and Sequence 3 (SanDan) include ramps, whereas Sequence 1 (S3E Playground) and Sequence 4 (Go2 Playground) do not contain sloped segments. The data collection environments for the quadruped robot sequences are illustrated in Figure 4. An overview of the sequences is provided in Table 1.

The four sequences were selected to separate the effects of platform motion and terrain elevation on LiDAR odometry. Sequence 1 was included as a public wheeled-robot sequence with dataset-provided ground truth and relatively smooth platform motion, providing a standard reference case. Sequences 2 and 3 were selected as quadruped-robot sequences containing ramps or slopes, where gait-induced vibration, abrupt $roll/pitch$ variation, and elevation changes jointly challenge LiDAR odometry. Sequence 4 was selected as a flat-terrain quadruped sequence to examine whether the proposed method also improves odometry under quadruped locomotion even without explicit elevation changes. This scenario design allows the evaluation to cover representative conditions for mobile robot navigation, including smooth wheeled motion, elevation-varying quadruped locomotion, and flat-terrain quadruped locomotion.

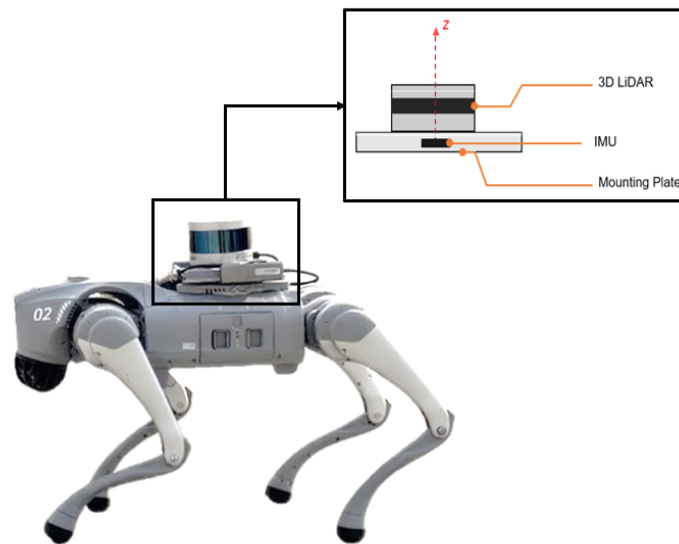


Figure 3. Experimental platform: Unitree Go2 quadruped robot equipped with LiDAR and IMU.

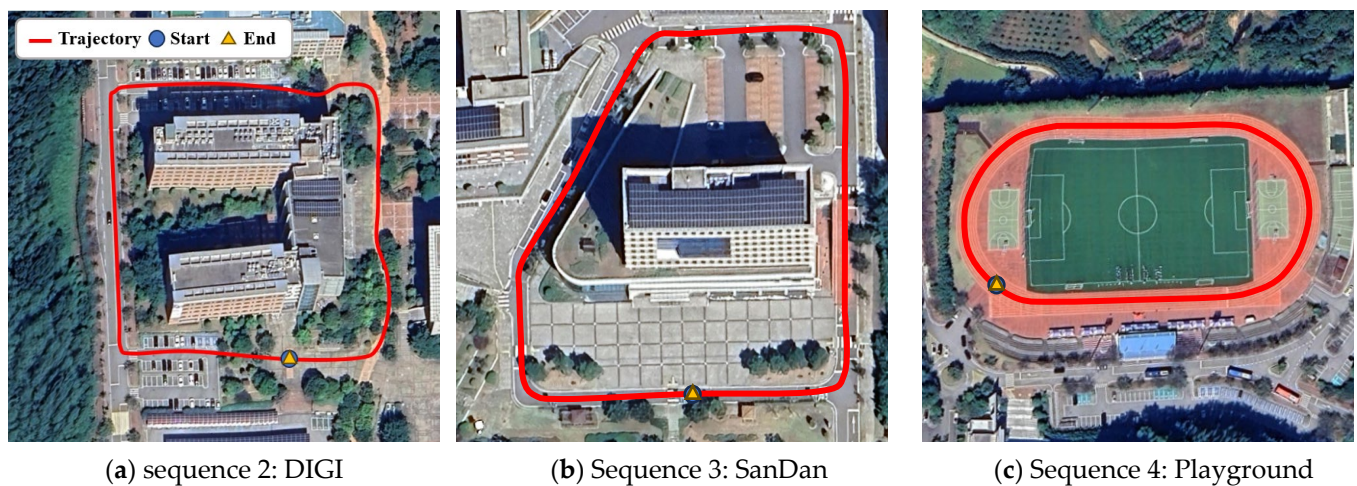


Figure 4. Satellite view of the data collection environments for the quadruped robot sequences.

Table 1. Overview of the evaluated sequences.

Sequence	Dataset/ Environment	Platform	LiDAR	Slope
Seq. 1	S3E Playground	Wheel robot	Velodyne VLP-16	No
Seq. 2	DIGI	Unitree Go2	Velodyne VLP-16	Yes
Seq. 3	SanDan	Unitree Go2	Velodyne VLP-16	Yes
Seq. 4	Playground	Unitree Go2	Velodyne VLP-16	No

All experiments were conducted on the same hardware platform consisting of a 12th Gen Intel Core i7-12700F CPU, 16 GB RAM, and an NVIDIA GeForce RTX 3070 GPU. All compared methods were evaluated on identical input point-cloud sequences, and the GPU was additionally used for pseudo-ground-truth generation with GLIM [34].

4.1.2. Compared Methods and Reference Trajectory

The proposed method was compared against three representative LiDAR registration and odometry methods: ICP [8], KISS-ICP [7], and F-LOAM [5]. ICP represents classical raw point-cloud registration, KISS-ICP is a lightweight point-based LiDAR odometry method, and F-LOAM represents a feature-based LiDAR odometry pipeline. These methods

provide a meaningful comparison across geometry-only registration, lightweight point-based odometry, and feature-based odometry.

The reference trajectory differed depending on the dataset source. For Sequence 1 (S3E Playground), we used the ground-truth trajectory provided by the public dataset; therefore, the errors on this sequence represent evaluation against real dataset-provided ground truth. For Sequences 2–4, which were collected using the Unitree Go2 quadruped robot in outdoor environments, externally measured ground truth such as motion capture, RTK-GNSS, or total-station measurements was not available. Therefore, we used GLIM-generated trajectories as pseudo ground truth (pGT). GLIM estimates the trajectory offline using LiDAR and iAHRS IMU measurements with GPU-accelerated scan-matching factors and fixed-lag smoothing, providing a consistent high-quality reference for relative comparison. On Sequence 1, GLIM also achieved the lowest error among all compared methods against the publicly provided ground truth, which further supports its use as a reference trajectory generator for Sequences 2–4. Nevertheless, this pGT is not an independently measured absolute ground truth. Accordingly, the errors reported for Sequences 2–4 should be interpreted as relative comparisons against a consistent reference trajectory, rather than as strict absolute-error bounds.

Performance was evaluated using Absolute Pose Error (APE). For each sequence, we report the maximum, mean, RMSE, and standard deviation of APE. RMSE and mean APE reflect overall trajectory accuracy, while the maximum error and standard deviation indicate worst-case behavior and estimation stability, respectively. All results were computed using the evo toolkit [35].

4.2. Comparison with State of the Arts

Tables 2 and 3 summarize the detailed quantitative results in terms of mean APE and RMSE across all four sequences. The best results are shown in bold, and the second-best results are underlined.

Table 2. Quantitative results (Sequence 1 from the public S3E wheeled-robot dataset; evaluated against the dataset-provided ground truth).

	Glim	Proposed	KISS-ICP	ICP	FLOAM
Maximum (m)	1.088952	<u>4.449973</u>	9.774980	89.017330	66.333835
Mean (m)	0.576101	<u>1.752100</u>	4.515873	51.924136	20.579071
RMSE (m)	0.691876	<u>2.113185</u>	5.376167	57.460399	29.736672
std (m)	0.383146	<u>1.177584</u>	2.917177	24.608567	21.465588

Table 3. Quantitative Results for quadruped robot sequences.

		Proposed	KISS-ICP	ICP	FLOAM
Seq. 2	Maximum (m)	3.565001	<u>37.354754</u>	72.041353	59.356439
	Mean (m)	1.335711	<u>15.482719</u>	29.493995	28.745720
	RMSE (m)	1.712981	<u>20.935810</u>	37.750446	35.238919
	std (m)	1.072054	<u>14.092188</u>	23.562691	20.382958
Seq. 3	Maximum (m)	7.768837	<u>23.404067</u>	57.881116	32.021459
	Mean (m)	2.850228	<u>12.019096</u>	36.709779	<u>9.706683</u>
	RMSE (m)	3.707536	<u>15.318486</u>	40.972650	<u>14.937945</u>
	std (m)	2.365196	<u>7.771880</u>	18.197531	11.354405
Seq. 4	Maximum (m)	9.360022	<u>19.884098</u>	59.871449	129.465068
	Mean (m)	3.888720	<u>11.954508</u>	38.755214	67.805222
	RMSE (m)	4.932879	<u>13.957694</u>	42.533199	78.637041
	std (m)	3.015417	<u>7.204587</u>	17.524452	39.827580

On Sequence 1 (S3E Playground, wheeled robot), the proposed method reduced the mean APE from 4.52 m to 1.75 m and the RMSE from 5.38 m to 2.11 m relative to KISS-ICP. This result indicates that the proposed deskewing and registration strategy remains beneficial even under comparatively smooth wheeled-motion conditions.

The improvements become substantially larger on the quadruped sequences. On Sequence 2 (DIGI), the proposed method achieved a mean APE of 1.34 m and an RMSE of 1.71 m, whereas KISS-ICP yielded 15.48 m and 20.94 m, respectively. This large gap suggests that the baseline formulation is more vulnerable to high-frequency body motion and abrupt attitude variation in quadruped locomotion.

A similar trend appears on Sequence 3 (SanDan), which also contains sloped segments. The proposed method obtained a mean APE of 2.85 m and an RMSE of 3.71 m, compared with 12.02 m and 15.32 m for KISS-ICP. Although F-LOAM performed better than KISS-ICP on this sequence, it still remained clearly behind the proposed method, indicating that the proposed point-based framework can remain competitive without relying on edge- and plane-feature extraction.

On Sequence 4 (Go2 Playground, quadruped, no slope), the proposed method again achieved the lowest error, with a mean APE of 3.89 m and an RMSE of 4.93 m. In contrast, KISS-ICP produced a mean APE of 11.95 m and an RMSE of 13.96 m, while F-LOAM showed the largest degradation. This result indicates that the benefit of the proposed method is not limited to ramp handling, but also extends to general vibration-induced distortion and vertical inconsistency during quadruped motion.

Overall, the proposed method consistently achieved the lowest error among the compared odometry baselines across all four sequences.

4.3. Qualitative Results

Figure 5 presents three-dimensional trajectory comparisons for all four sequences. On Sequence 1 (S3E, wheeled robot), the baseline methods already show noticeable three-dimensional inconsistency with respect to the reference trajectory. ICP exhibits severe spatial distortion with large vertical excursions, KISS-ICP preserves the loop tendency but still shows clear Z-axis deviation, and F-LOAM forms a contracted inner loop rather than following the reference boundary. In contrast, the proposed method remains most closely aligned with the reference trajectory in all three axes. On Sequence 2 (DIGI), which includes ramps, ICP and F-LOAM diverge substantially from the reference trajectory, while KISS-ICP preserves the overall path geometry but exhibits significant vertical drift on the sloped segment. In contrast, the proposed method remains tightly aligned with the reference throughout the sequence. A similar trend is observed on Sequence 3 (SanDan), where the baseline methods exhibit varying degrees of three-dimensional drift, particularly in the vertical direction, whereas the proposed method maintains a trajectory shape most consistent with the reference. On Sequence 4 (Go2 Playground), which does not contain ramps, ICP and F-LOAM still show large spatial deviations caused by quadruped locomotion, while KISS-ICP preserves the overall loop shape more successfully but accumulates noticeable drift over the full trajectory. The proposed method again remains closest to the reference, preserving the loop geometry most faithfully despite vibration-induced motion distortion. Across all sequences, the proposed method produces the most consistent three-dimensional trajectory, with particularly clear advantages in vertical stability on the elevation-varying sequences.

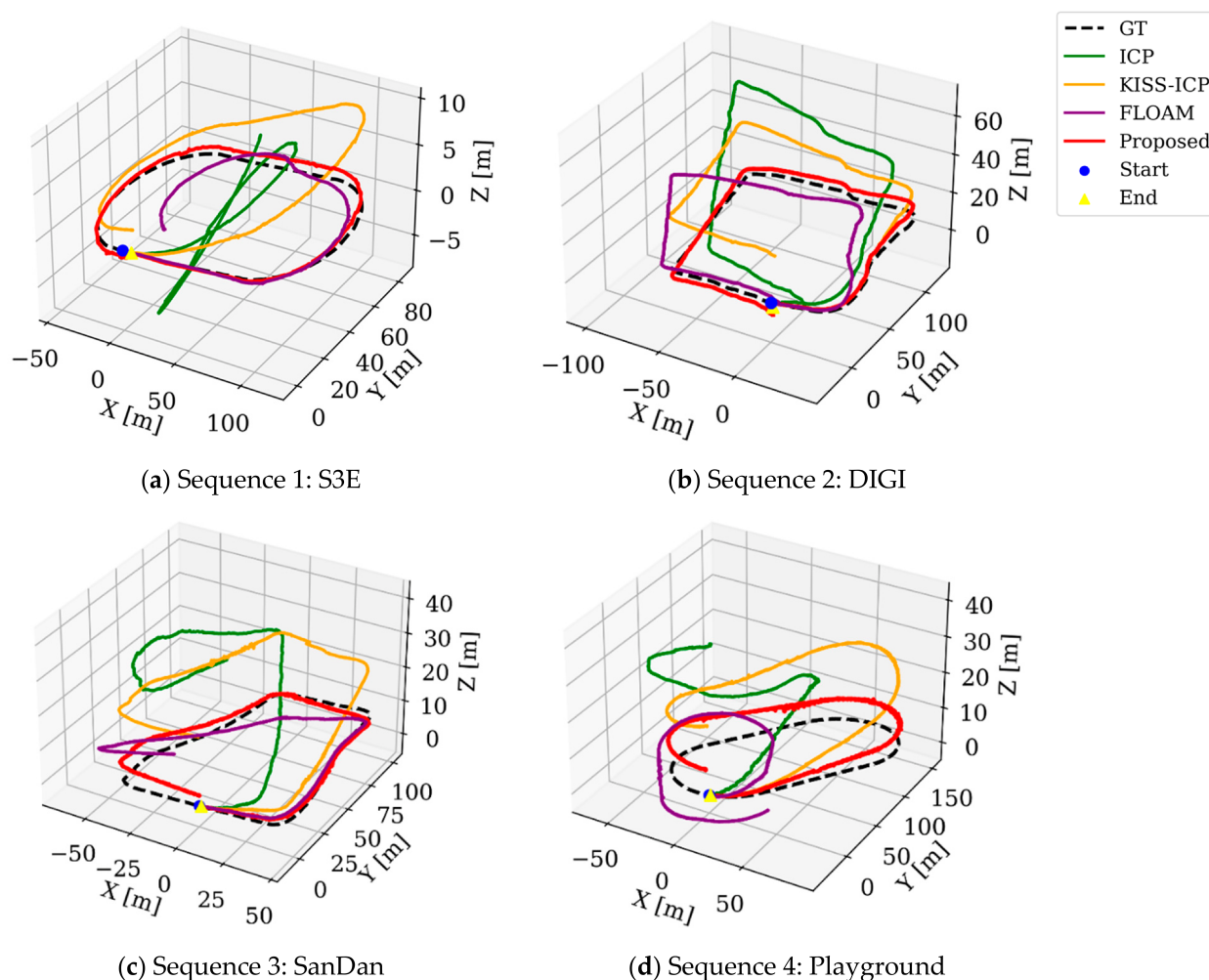


Figure 5. Three-dimensional trajectory comparison across all four sequences.

Figure 6 presents side-view trajectory comparisons in the X–Z plane, directly illustrating the vertical drift behavior of each method. On Sequence 1, ICP produces large negative and positive Z excursions, and KISS-ICP also deviates noticeably upward from the near-flat reference profile. F-LOAM remains relatively bounded in the vertical direction, although it still shows a visible offset from the reference. In contrast, the proposed method remains closely aligned with the reference trajectory and maintains the most stable near-zero height profile throughout. On Sequence 2 (DIGI), ICP and KISS-ICP exhibit severe upward Z drift upon entering the ramp, while F-LOAM also shows a clear elevation mismatch and larger vertical error than the proposed method. The proposed method tracks the gradual elevation change most smoothly without unstable vertical jumps. A similar pattern is observed on Sequence 3 (SanDan), which contains a longer sloped segment: ICP and KISS-ICP exhibit pronounced vertical drift, whereas F-LOAM shows relatively limited Z-axis divergence but still does not follow the reference slope profile as closely as the proposed method. On Sequence 4 (Playground), even without ramps, ICP and KISS-ICP show considerable vertical oscillation and drift caused by quadruped locomotion alone. F-LOAM again remains relatively bounded in Z, but the proposed method stays closest to the flat reference trajectory and suppresses vertical oscillations most effectively. Overall, F-LOAM does not exhibit the largest vertical error in most sequences and tends to remain more bounded in Z than the point-based baselines; however, its reference consistency remains limited, and its large horizontal distortion, shown in Figures 5 and 7, still degrades overall trajectory accuracy. These results indicate that the decoupled horizontal–vertical optimization and

observability-aware vertical refinement effectively reduce Z-axis drift across both flat and elevation-varying terrain conditions.

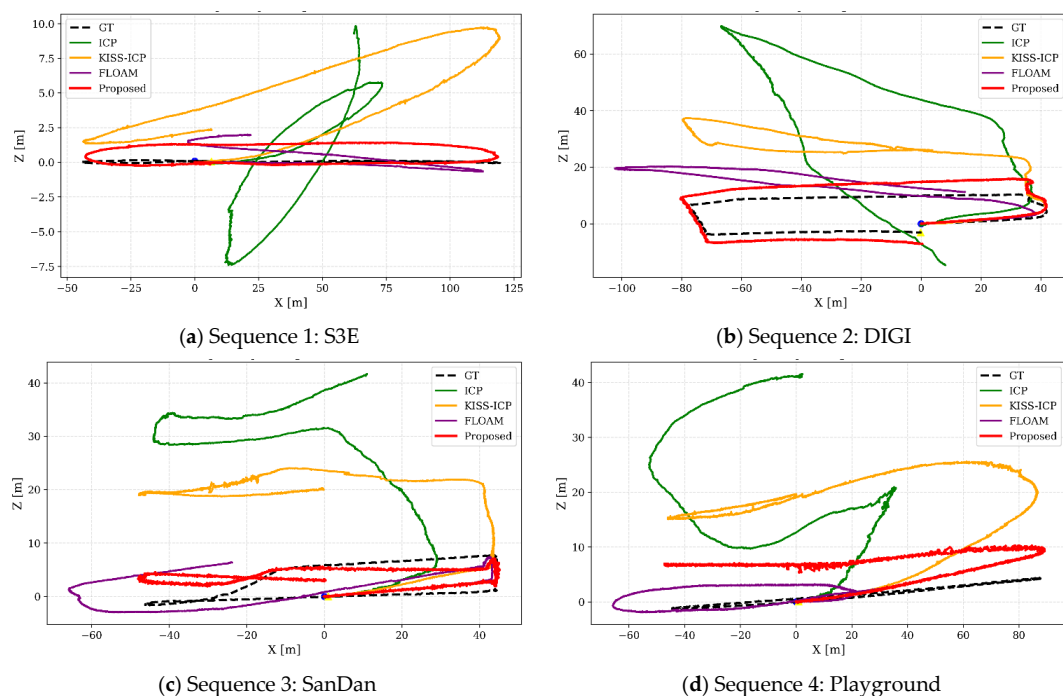


Figure 6. Side-view trajectory comparison illustrating vertical drift suppression across all four sequences. The blue circle indicates the start position, and the yellow triangle indicates the end position.

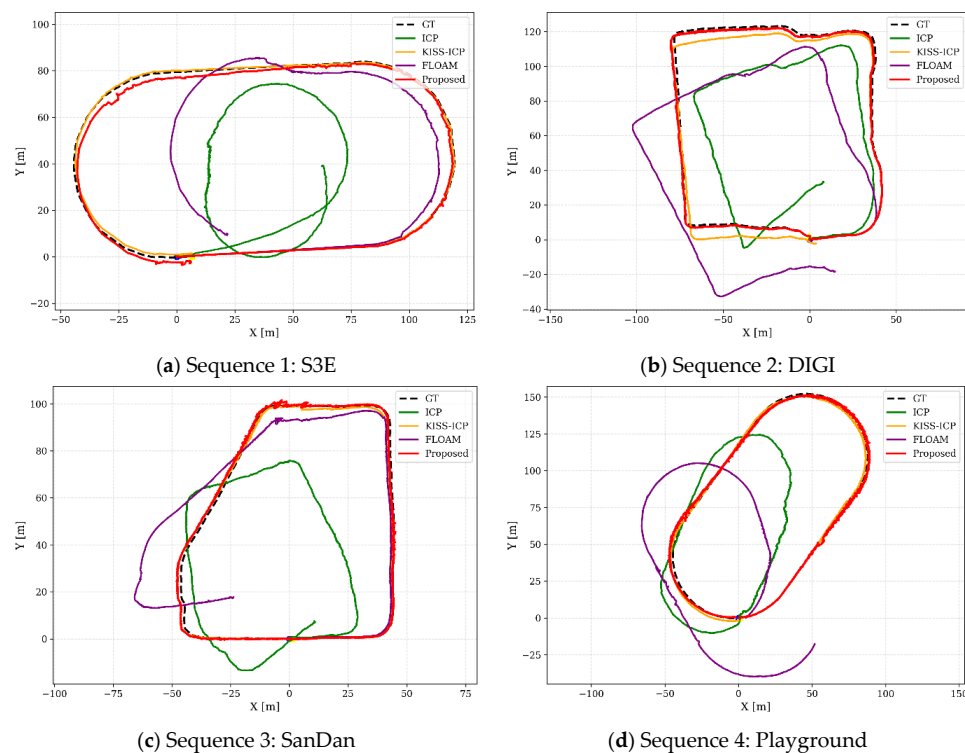


Figure 7. Top-down view trajectory comparison. The blue circle indicates the start position, and the yellow triangle indicates the end position.

Figure 7 presents top-down trajectory comparisons in the X–Y plane. On Sequence 1, the proposed method and KISS-ICP follow the outer loop shape relatively closely, while

ICP forms a contracted inner loop and F-LOAM shows a laterally shifted trajectory with a different loop shape. On Sequence 2 (DIGI), the proposed method remains closest to the reference boundary. KISS-ICP preserves the overall rectangular loop shape but shows a visible offset and scale difference, whereas ICP and F-LOAM deviate substantially from the reference path. On Sequence 3 (SanDan), the proposed method and KISS-ICP again follow the reference loop closely, while ICP produces a contracted inner trajectory and F-LOAM shows a distorted path, particularly on the left side of the loop. On Sequence 4 (Playground), the proposed method and KISS-ICP both preserve the overall loop shape, with the proposed method remaining closest to the reference trajectory. In contrast, ICP forms an inner distorted loop and F-LOAM shows a largely deformed trajectory. Overall, the proposed method remains closest to the reference across all four sequences, while KISS-ICP also shows relatively good horizontal agreement in the X–Y plane.

4.4. Further Analysis and Discussion

A key observation from the experiments is that the proposed method yields much larger gains on the quadruped sequences than on the wheeled sequence. KISS-ICP relies on a constant-velocity motion model for deskewing and a coupled point-to-point registration framework. Such assumptions are adequate when the platform motion within a LiDAR sweep is small and smooth, but become less reliable under rapid vertical oscillation, abrupt *roll/pitch* changes, and irregular body motion, as is common in quadruped locomotion.

The improvements are particularly pronounced on DIGI and SanDan, both of which contain ramps. This suggests that the main advantage of the proposed method is not only more accurate deskewing but also more stable handling of the vertical degrees of freedom. In standard 6-DoF coupled optimization, unstable updates in *z*, *roll*, and *pitch* can contaminate the horizontal estimate. The proposed method mitigates this issue by separating horizontal and vertical updates and introducing an additional vertical refinement process with point-to-plane constraints.

Another notable result is that the advantage of the proposed method remains visible even on Sequence 4, which contains no ramps. This indicates that the proposed design addresses not only slope-induced instability but also the broader scan distortion and vertical inconsistency caused by quadruped locomotion itself. The method can therefore be understood as a LiDAR odometry design tailored to the motion characteristics of legged robots rather than a terrain-specific correction alone.

Figure 8 shows point-cloud maps generated by integrating the proposed odometry front-end with Scan Context [36]-based loop closure and pose graph optimization, demonstrating that the proposed front-end is compatible with standard back-end components without architectural modification.

This study also has several limitations. First, the quadruped sequences were evaluated against GLIM-based pseudo ground truth rather than externally measured absolute ground truth. Although Sequence 1 was evaluated using the ground-truth trajectory provided by the public S3E dataset, Sequences 2–4 did not include external ground truth from motion capture, RTK-GNSS, or total-station measurements. Therefore, the reported errors for these quadruped sequences should be interpreted mainly as relative comparisons against a strong and consistent reference trajectory, rather than as strict absolute accuracy. Future work will include evaluation with externally measured ground truth to further validate the absolute accuracy of the proposed method.

Second, the proposed method includes platform-dependent hyperparameters. The clamping thresholds ($\zeta_z, \zeta_r, \zeta_p$) and the number of temporal segments *K* were configured based on the motion characteristics of the Unitree Go2 platform used in this study. Although the overall framework is structurally platform-agnostic, quadruped robots with different

gait patterns, body masses, walking speeds, or LiDAR mounting configurations may exhibit different levels of vertical oscillation, *roll/pitch* variation, and intra-scan distortion. This tendency is also reflected in the experimental results. The proposed method achieved a relatively moderate improvement on Sequence 1, which was collected using a wheeled robot on flat terrain, reducing the mean APE from 4.52 m to 1.75 m compared with KISS-ICP. In contrast, larger gains were observed on the quadruped sequences; for example, on Sequence 2, the mean APE decreased from 15.48 m to 1.34 m. These results suggest that the proposed design is particularly beneficial under legged locomotion, where the single constant-velocity assumption becomes less reliable. For other quadruped robots, bipedal robots, or different sensor configurations, platform-aware parameter adjustment may further improve performance because gait pattern, body mass, walking speed, and LiDAR mounting configuration can affect the magnitude of vertical oscillation and intra-scan distortion.

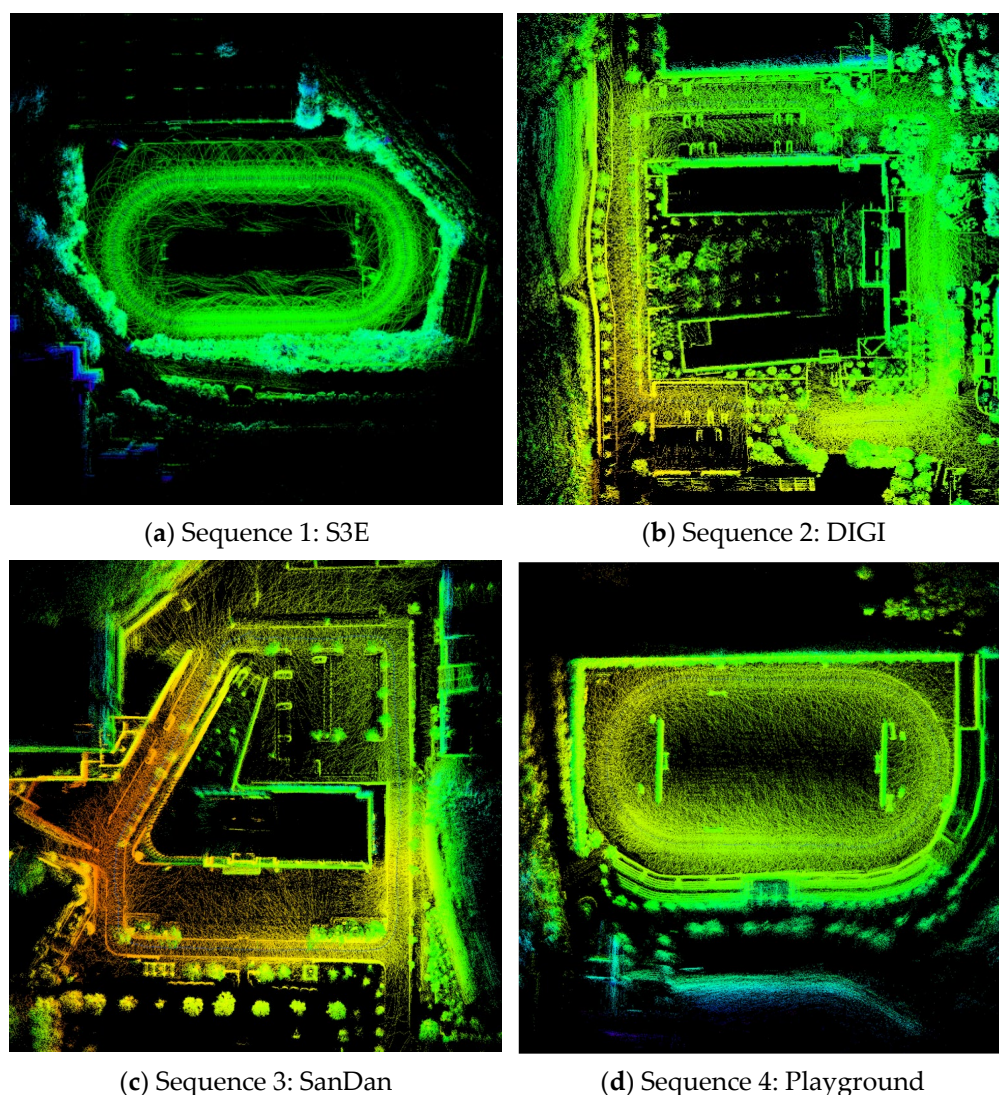


Figure 8. Point-cloud maps generated by the proposed odometry front-end with Scan Context-based loop closure. The color map represents height information, with cooler colors indicating lower elevations and warmer colors indicating higher elevations.

Third, the observability-aware weighting in the vertical stage depends on the availability of terrain surfaces that provide meaningful point-to-plane constraints for the vertical state. In environments dominated by vertical structures, such as narrow corridors or dense indoor scenes, or in geometrically sparse outdoor environments, the vertical observability

score Ω_i may remain low for most correspondences. In such cases, the advantage of the proposed decoupled vertical stage over conventional single-stage ICP may decrease, and the vertical refinement may provide limited additional benefit.

5. Conclusions

LiDAR-based odometry is an important front-end module for autonomous navigation of quadruped robots. However, most existing LiDAR odometry methods have been developed primarily for wheeled platforms and are therefore not well suited to the unique motion characteristics of quadruped robots, including high-frequency vibrations, abrupt postural changes, and weakened vertical observability in environments with varying elevation. These factors introduce point-cloud distortion during scan acquisition and can lead to accumulated Z-axis drift and unstable pose estimation. Our experiments also showed that ICP, KISS-ICP, and F-LOAM all exhibit substantially larger errors on the quadruped sequences than on the wheeled-platform sequence, highlighting the limitation of conventional LiDAR odometry under legged locomotion.

To address these issues, this paper presented an improved LiDAR odometry method for quadruped robots based on three key components. First, the proposed Piecewise-Constant Velocity deskewing divides each scan into multiple temporal segments, relaxing the limitations of a single constant-velocity assumption, while Z-axis and *roll-pitch* clamping suppress abnormal initial motion estimates. Second, the proposed decoupled horizontal-vertical two-stage ICP separates SE(3) optimization into planar and vertical components, preventing vertical instability from contaminating horizontal pose estimation. Third, the vertical stage introduces point-to-plane constraints together with observability-aware weighting and bounded update control to improve robustness of Z-axis estimation in slopes and other elevation-varying terrain.

Experiments on four sequences—S3E, DIGI, SanDan, and Playground—showed that the proposed method consistently achieved lower accumulated error and more stable trajectory estimation than ICP, KISS-ICP, and F-LOAM. Quantitatively, compared with KISS-ICP, the proposed method reduced the mean APE from 4.52 m to 1.75 m, 15.48 m to 1.34 m, 12.02 m to 2.85 m, and 11.95 m to 3.89 m on Sequences 1–4, respectively. The corresponding RMSE values were reduced from 5.38 m to 2.11 m, 20.94 m to 1.71 m, 15.32 m to 3.71 m, and 13.96 m to 4.93 m. These quantitative improvements support the conclusion that the proposed PCV deskewing and decoupled horizontal-vertical registration improve LiDAR-only odometry accuracy, with particularly clear benefits under quadruped locomotion and elevation-varying terrain.

Future work will investigate the integration of the proposed odometry front-end with back-end modules such as loop closure and pose graph optimization to improve long-range global consistency in larger environments.

Author Contributions: Conceptualization, H.L.; Methodology, E.H.; Software, E.H.; Validation, E.H.; Formal analysis, H.L.; Writing—original draft, E.H.; Writing—review & editing, H.L.; Supervision, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2026-RS-2024-00438430, 50%) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), and in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2026-RS-2024-00437190, 50%) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The S3E dataset used in this study is publicly available from its original source. The quadruped robot datasets (DIGI, SanDan, and Go2 Playground) were collected by the authors and are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Kim, D.; Lee, H. FLASH: Fibonacci Lattice Spherical Harmonics for Semantic Place Recognition Using LiDAR. *IEEE Robot. Autom. Lett.* **2026**, *11*, 4521–4528. [\[CrossRef\]](#)
- Kim, D.; Han, E.; Kang, S.; Lee, S.; Lee, H. A Multisensor Dataset for SLAM in Repetitive Scene Environments. *IEEE Sens. Lett.* **2026**, *10*, 5502204. [\[CrossRef\]](#)
- Kim, D.; Lee, H.; Choi, K.H. Pitch Variation Filter for LiDAR-Only SLAM and Localization in Self-Balancing Mobile Robot. *IEEE Robot. Autom. Lett.* **2025**, *10*, 12836–12843. [\[CrossRef\]](#)
- Ji, Z.; Singh, S. LOAM: Lidar odometry and mapping in real-time. *Robot. Sci. Syst.* **2014**, *2*, 1–9.
- Wang, H.; Wang, C.; Chen, C.-L.; Xie, L. F-LOAM: Fast LiDAR Odometry and Mapping. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 4390–4396. [\[CrossRef\]](#)
- Xu, W.; Cai, Y.; He, D.; Lin, J.; Zhang, F. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Trans. Robot.* **2022**, *38*, 2053–2073. [\[CrossRef\]](#)
- Vizzo, I.; Guadagnino, T.; Mersch, B.; Wiesmann, L.; Behley, J.; Stachniss, C. KISS-ICP: In Defense of Point-to-Point ICP—Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1029–1036. [\[CrossRef\]](#)
- Besl, P.J.; McKay, N.D. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [\[CrossRef\]](#)
- Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152. [\[CrossRef\]](#)
- Shan, T.; Englot, B. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765. [\[CrossRef\]](#)
- Seo, D.-U.; Lim, H.; Lee, S.; Myung, H. PaGO-LOAM: Robust Ground-Optimized LiDAR Odometry. In Proceedings of the 2022 19th International Conference on Ubiquitous Robots (UR), Jeju, Republic of Korea, 4–6 July 2022; pp. 1–7. [\[CrossRef\]](#)
- Nikhil, J.; Lyu, Y.; Zhang, Z. Lidar odometry methodologies for autonomous driving: A survey. *arXiv* **2021**, arXiv:2109.06120. [\[CrossRef\]](#)
- Jens, B.; Stachniss, C. Efficient surfel-based SLAM using 3D laser range data in urban environments. *Robot. Sci. Syst.* **2018**, *2018*, 59.
- Deschaud, J.-E. IMLS-SLAM: Scan-to-Model Matching Based on 3D Data. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2480–2485. [\[CrossRef\]](#)
- Pan, Y.; Xiao, P.; He, Y.; Shao, Z.; Li, Z. MULLS: Versatile LiDAR SLAM via Multi-metric Linear Least Square. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 11633–11640. [\[CrossRef\]](#)
- Dellenbach, P.; Deschaud, J.-E.; Jacquet, B.; Goulette, F. CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 5580–5586. [\[CrossRef\]](#)
- Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24–30 October 2020; pp. 5135–5142. [\[CrossRef\]](#)
- Ramezani, M.; Tinchev, G.; Iuganov, E.; Fallon, M. Online LiDAR-SLAM for Legged Robots with Robust Registration and Deep-Learned Loop Closure. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 4158–4164. [\[CrossRef\]](#)
- Wisth, D.; Camurri, M.; Das, S.; Fallon, M. Unified Multi-Modal Landmark Tracking for Tightly Coupled Lidar-Visual-Inertial Odometry. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1004–1011. [\[CrossRef\]](#)
- Ou, G.; Li, D.; Li, H. Leg-KILO: Robust Kinematic-Inertial-Lidar Odometry for Dynamic Legged Robots. *IEEE Robot. Autom. Lett.* **2024**, *9*, 8194–8201. [\[CrossRef\]](#)
- Zhao, Q.; Li, M.; Shi, Y.; Chen, X.; Yu, Z.; Han, L.; Fu, Z.; Zhang, J.; Li, C.; Zhang, Y.; et al. LIKO: LiDAR, Inertial, and Kinematic Odometry for Bipedal Robots. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 1180–1185. [\[CrossRef\]](#)

22. Wang, Y.; Kuang, J.; Niu, X.; Liu, J. LLIO: Lightweight Learned Inertial Odometer. *IEEE Internet Things J.* **2023**, *10*, 2508–2518. [\[CrossRef\]](#)
23. Okawara, T.; Koide, K.; Takanose, A.; Oishi, S.; Yokozuka, M.; Uno, K.; Yoshida, K. Tightly-Coupled LiDAR-IMU-Leg Odometry With Online Learned Leg Kinematics Incorporating Foot Tactile Information. *IEEE Robot. Autom. Lett.* **2025**, *10*, 7947–7954. [\[CrossRef\]](#)
24. Li, J.; Zeng, D.; Luo, Q.; Luo, X.; Chen, C.L.P.; Yang, C. Feature Assessment and Enhanced Vertical Constraint Lidar Odometry and Mapping on Quadruped Robot. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 5008814. [\[CrossRef\]](#)
25. Nobili, S.; Tinchev, G.; Fallon, M. Predicting Alignment Risk to Prevent Localization Failure. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 1003–1010. [\[CrossRef\]](#)
26. Zhang, J.; Kaess, M.; Singh, S. On degeneracy of optimization-based state estimation problems. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 809–816. [\[CrossRef\]](#)
27. Zhen, W.; Scherer, S. Estimating the Localizability in Tunnel-like Environments using LiDAR and UWB. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4903–4908. [\[CrossRef\]](#)
28. Hinduja, A.; Ho, B.-J.; Kaess, M. Degeneracy-Aware Factors with Applications to Underwater SLAM. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 1293–1299. [\[CrossRef\]](#)
29. Tuna, T.; Nubert, J.; Nava, Y.; Khattak, S.; Hutter, M. X-ICP: Localizability-Aware LiDAR Registration for Robust Localization in Extreme Environments. *IEEE Trans. Robot.* **2024**, *40*, 452–471. [\[CrossRef\]](#)
30. Hatleskog, J.; Alexis, K. Probabilistic Degeneracy Detection for Point-to-Plane Error Minimization. *IEEE Robot. Autom. Lett.* **2024**, *9*, 11234–11241. [\[CrossRef\]](#)
31. Lv, J.; Zuo, X.; Hu, K.; Xu, J.; Huang, G.; Liu, Y. Observability-Aware Intrinsic and Extrinsic Calibration of LiDAR-IMU Systems. *IEEE Trans. Robot.* **2022**, *38*, 3734–3753. [\[CrossRef\]](#)
32. Chebrolu, N.; Läbe, T.; Vysotska, O.; Behley, J.; Stachniss, C. Adaptive Robust Kernels for Non-Linear Least Squares Problems. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2240–2247. [\[CrossRef\]](#)
33. Feng, D.; Qi, Y.; Zhong, S.; Chen, Z.; Chen, Q.; Chen, H.; Wu, J.; Ma, J. S3E: A Multi-Robot Multimodal Dataset for Collaborative SLAM. *IEEE Robot. Autom. Lett.* **2024**, *9*, 11401–11408. [\[CrossRef\]](#)
34. Koide, K.; Yokozuka, M.; Oishi, S.; Banno, A. GLIM: 3D range-inertial localization and mapping with GPU-accelerated scan matching factors. *Robot. Auton. Syst.* **2024**, *179*, 104750. [\[CrossRef\]](#)
35. Grupp, M. Evo: Python Package for the Evaluation of Odometry and SLAM. 2017. Available online: <https://github.com/MichaelGrupp/evo> (accessed on 1 March 2025).
36. Kim, G.; Kim, A. Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4802–4809. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.