Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Integration of Scan-Matching-based LiDAR Odometry with INS and GNSS for UAVs in Unstructured Natural Environments

Artur C. Fabrício

arfaro@dtu.dk

Technical University of Denmark

Paul D. Groves

University College London

Peter Bauer-Gottwein

University of Copenhagen

Daniel H. Olesen

Technical University of Denmark

Research Article

Keywords: UAV, INS, GNSS, LiDAR, ICP, river

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10903953/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Integration of Scan-Matching-based LiDAR Odometry with INS and GNSS for UAVs in Unstructured Natural Environments

Artur C. Fabrício[1*], Paul D. Groves[2], Peter Bauer-Gottwein[3], Daniel H. Olesen[1]

[1*]DTU Space, Technical University of Denmark, Elektrovej 328, Kongens Lyngby, 2800, Denmark.
[2]Dept. of Civil, Environmental & Geomatic Engineering, University College London, Gower Street, London, WC1E 6BT, United Kingdom.
[3]Department of Geosciences and Natural Resource Management, University of Copenhagen, Øster Voldgade 10, Copenhagen, 1350, Denmark.

*Corresponding author(s). E-mail(s): arfaro@dtu.dk;
Contributing authors: p.groves@ucl.ac.uk; pbg@ign.ku.dk; danole@dtu.dk;

Abstract

Uncrewed Aerial Vehicles (UAVs) offer a low-cost and flexible means of collecting river hydrometry data, but the quality of the resulting measurements depends on accurate georeferencing. In unstructured natural environments, such as rivers, this is challenging as vegetated riverbanks and tree-canopy occlusion can degrade Global Navigation Satellite System (GNSS) reception, which in turn degrades the vehicle's Inertial Navigation System (INS)/GNSS navigation solution, worsening its position, velocity, and attitude estimates. This paper proposes an INS/GNSS/Light Detection and Ranging (LiDAR) architecture based on an error state Kalman filter, in which double-differenced GNSS measurements are tightly coupled, while velocity and attitude from a lightweight, scan-matching-based Iterative Closest Point (ICP) LiDAR odometry are loosely coupled. This odometry avoids reliance on both large local maps, and discernible geometric features, since water absorption and foliage scattering produce sparse, feature-poor point clouds. The proposed method is evaluated on two real-world airborne datasets from natural environments and compared against standalone Post-Processed Kinematic positioning, a state-of-the-art factor-graph approach, and a filter-based feature-dependent method. The solution outperforms all tested approaches across position, velocity, and attitude. Relative to a conventional double-differenced INS/GNSS integration, the scan-based solution reaches centimeter-level position accuracy across all axes in the river environment (Root Mean Square Error (RMSE) under 0.07 m). LiDAR measurements also contribute substantially to attitude estimation, particularly to heading, reducing the yaw RMSE by 66% compared to INS/GNSS. Moreover, the entire architecture runs in real time on a standard laptop, remaining well within a per-epoch processing budget suitable for onboard deployment.

Keywords: UAV, INS, GNSS, LiDAR, ICP, river

1 Introduction

Uncrewed Aerial Vehicles (UAVs) have become widespread in recent years due to their commercial accessibility and ease of use, and their ability to reach remote areas makes them well suited for field data collection. One such application is river hydrometry, where traditional methods rely on satellite observations with low temporal and spatial resolution, or on cumbersome and expensive manual in-situ measurements (Bandini et al. 2017, 2018). UAVs offer a low-cost, flexible alternative, but the quality of the data depends on the onboard positioning solution, as accurate georeferencing is required to derive products such as water surface elevation, river discharge, and river velocimetry.

Georeferencing currently relies on Real-Time Kinematic (RTK) or Post-Processed Kinematic (PPK) solutions (Bandini et al. 2023; Zhou et al. 2024), achieving centimeter-level accuracy (Leick et al. 2015). However, Global Navigation Satellite System (GNSS) signals received by the rover remain susceptible to attenuation, diffraction, and multipath. Along rivers, vegetated banks and tree canopies can block line of sight to satellites, and vegetation has been shown to attenuate and scatter GNSS signals, lowering signal strength and introducing unmodelled errors (Zhou et al. 2025).

A robust position, velocity, and attitude solution is also crucial for safe operation. The onboard navigation solution of UAVs rely on GNSS, so degraded GNSS under vegetation or near river banks can lead to unbounded inertial drift. Since hydrometric surveys may require flying as low as 0.5 m above the water (Bandini et al. 2023), GNSS degradation may negatively perturb the navigation solution, which can cause loss of control and a crash.

1.1 Related Works

Research on UAVs in river environments has mostly focused on environmental mapping, flight planning, and autonomous control. Chambers et al. (2011) proposed a river-mapping sensor suite in which stereo visual odometry (VO) is fused with IMU and sparse L1 GPS measurements through graph-based optimization, with laser scans used only for georeferencing rather than state estimation. Despite efforts to address VO scale issues in feature-poor scenes (Rehder et al. 2012), position errors reached almost 10 m, with mean absolute errors of up to 5 m. An Unscented Kalman Filter (UKF) fusing GPS, IMU, stereo camera, and barometer data (Chambers et al. 2014) still yielded errors above 3 m. Nuske et al. (2015) instead used specular water-surface laser returns to estimate height above the river, while vision-based SLAM relying on riverbank and water-surface features (Yang et al. 2017) reported average position errors of approximately 10 m. These results highlight the limitations of vision-based state estimation over rivers, where low-texture, reflective, and moving water surfaces provide few stable features. In contrast, Fabrício et al. (2025) successfully used INS/LiDAR navigation over a river when RTK/PPK was unavailable, although LiDAR and GNSS were not used concurrently.

In these approaches, GNSS is used sparsely and treated as a black box: the receiver's position–velocity fix, which requires at least four satellites, is loosely coupled with the inertial solution. By contrast, tightly coupled approaches fuse raw GNSS observables directly with inertial data, so that fewer than four visible satellites can still update the solution (Groves 2013). Ultra-tightly (or deep) coupled INS/GNSS goes further by aiding the receiver's signal tracking with inertial data, which has been shown to improve positioning on small UAVs flying under dense foliage (Olesen et al. 2017). However, this requires access to raw ADC signal samples, which commercial off-the-shelf (COTS) receivers do not provide. Double-differenced observations, as in RTK/PPK, have been fused with IMU and LiDAR in a filter to handle GNSS degradation and outages in urban environments (Wan et al. 2018; Cheng et al. 2025) and to improve ambiguity fixing (Li et al. 2023), while Li et al. (2021) used Precise Point Positioning (PPP) instead. In these methods, LiDAR is also tightly coupled, with residuals from planar and edge features serving as filter measurements.

More broadly, LiDAR-based state estimation has proven highly successful in urban environments, from LiDAR Odometry and Mapping (LOAM) (Zhang and Singh 2014) to the iterated Kalman filter-based FAST-LIO2 (Xu et al. 2022), the extended Kalman filter-based LIO-EKF (Wu et al. 2024), and the factor-graph-based LIO-SAM (Shan et al. 2020), which exploit the abundant planar and line features and dense scan returns of building facades, signs, and poles. In river environments, however, LiDAR has so far been used mainly for mapping or altimetry (Chambers et al. 2011;

Nuske et al. 2015), and, where used for state estimation, not concurrently with GNSS (Fabrício et al. 2025). This leaves a notable gap: the bank vegetation and canopy that degrade GNSS also produce dense LiDAR returns, making LiDAR a natural complement to GNSS precisely where GNSS struggles. Over the water itself, near-infrared LiDAR receives weak backscatter, as many pulses return no detectable echo, leaving sparse point clouds (Höfle et al. 2009).

With sparse returns over water and irregular returns from vegetation, reliable geometric features are hard to extract, making direct scan-matching techniques a promising alternative (Lee et al. 2024). Iterative Closest Point (ICP) (Besl and McKay 1992) computes the rotation and translation between two, usually downsampled, point clouds, either scan-to-scan or scan-to-map. KISS-ICP (Vizzo et al. 2023), for example, employs point-to-point ICP with adaptive thresholding to register each scan against a local map. The contributions of this paper are:

- An INS/GNSS/LiDAR architecture based on an error-state Kalman filter (ESKF) that tightly couples double-differenced GNSS measurements and loosely couples velocity and attitude from LiDAR odometry. It is designed to operate from highly degraded GNSS with dense LiDAR returns (near river banks or under dense vegetation) to open-sky conditions with few or no returns.
- A simple, lightweight LiDAR odometry algorithm, decoupled from the ESKF and loosely based on KISS-ICP, that uses keyframes (selected past scans kept in memory) to build a local map for scan matching.
- An assessment on real airborne datasets from representative natural environments, including a river, against standalone PPK, INS/GNSS, a feature-based INS/GNSS/LiDAR filter, and a state-of-the-art feature-based factor-graph method.

The paper is organized as follows. The LiDAR odometry algorithm is described first, followed by the filter states, propagation equations, and GNSS and LiDAR measurement models. The test sites and data collection are then introduced, followed by the results and discussion. Finally, the conclusion summarizes our findings and outlines future work.

2 Methodology

2.1 Keyframe-based LiDAR Odometry

Rather than registering each scan against only the previous scan (scan-to-scan) or against a continuously maintained map (scan-to-map), the proposed LiDAR odometry adopts an intermediate scan-to-keyframes approach. A scan is selected as a keyframe, κ, when the UAV has translated or rotated beyond a set threshold since the previous keyframe. Only the n_κ most recent keyframes are kept: when a new keyframe is added, the oldest is discarded, so the keyframes form a rolling window that is merged into a temporary local map. Although scan-to-map matching generally outperforms scan-to-scan (Lee et al. 2024), limiting the reference to a small, fixed number of keyframes keeps the computational cost low while providing a denser reference than a single scan. LIO-SAM (Shan et al. 2020) follows a similar strategy, although its keyframes consist of edge and planar features rather than full point clouds.

Since each keyframe is stored in its own sensor frame, the keyframes must first be brought into a common frame before they can be merged into the local keyframe-based map. The most recent keyframe is used as this reference, and each older keyframe is transformed into it by chaining the relative poses previously estimated by ICP. Because errors compound along this chain, older keyframes carry more drift. Anchoring the map at the newest keyframe keeps the keyframes that

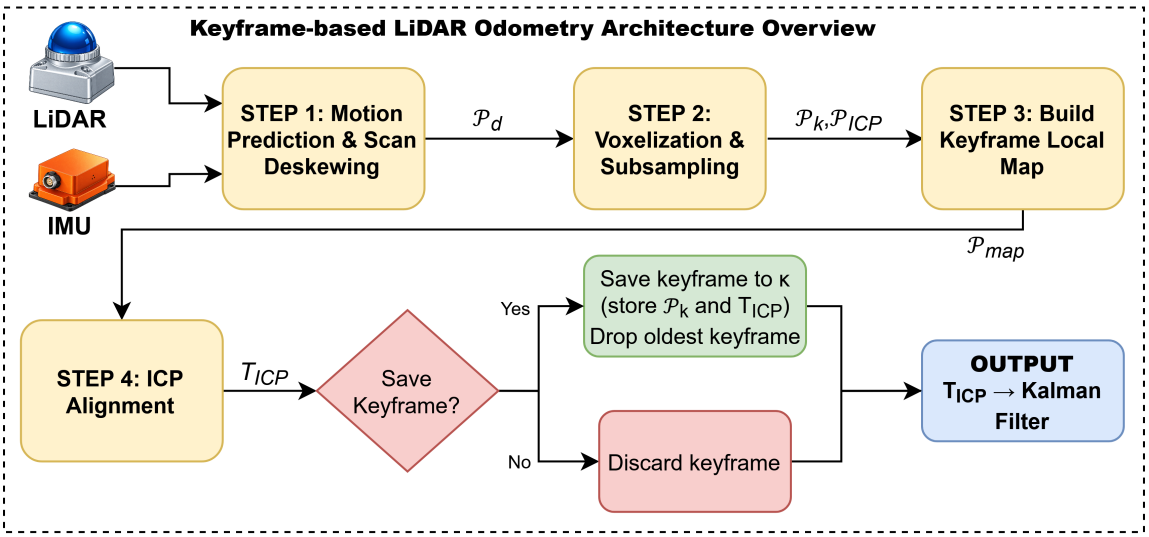


Fig. 1 Overview of the proposed keyframe-based LiDAR odometry

overlap most with the current scan the most accurate. This is important because sparse scans yield few correspondences, leaving ICP weakly constrained and sensitive to inconsistencies in the local map. The architecture is shown in figure 1.

Step 1: Motion Prediction and Scan Deskewing

Each raw scan, $\mathcal{P}_r \subset \mathbb{R}^3$, is expressed in the LiDAR frame $\mathcal{F}_L$. Since a sweep spans a finite scan period, τ_{LiDAR}, platform motion distorts the point cloud (He et al. 2020). We therefore preintegrate the IMU specific forces and angular rates over τ_{LiDAR} (Forster et al. 2015) and use the resulting relative motion to obtain a deskewed (undistorted) point cloud, $\mathcal{P}_d$.

Step 2: Voxelization and Subsampling

For efficiency, $\mathcal{P}_d$ is downsampled twice, inspired by KISS-ICP (Vizzo et al. 2023), with voxel sizes $\alpha \times v$ and $\beta \times v$ for a base voxel size v: the former yields the ICP scan, $\mathcal{P}_{\text{ICP}}$, and the latter the keyframe scan, $\mathcal{P}_k$.

Step 3: Build Keyframe Local Map

The keyframes $\{\kappa_1, \ldots, \kappa_{n_\kappa}\}$ are ordered from most recent, κ_1, to oldest, κ_{n_κ}. For each keyframe κ_i, the subsampled scan $\mathcal{P}_{k_i} \subset \mathbb{R}^3$ and the transform $T_{\kappa_i} \in SE(3)$ estimated by ICP when the keyframe was created are stored, where T_{κ_i} maps points from the frame of κ_i to that of the previous keyframe, κ_{i+1}:

$$T_{\kappa_i} = \begin{bmatrix} R_{\kappa_i} & t_{\kappa_i} \\ 0_{1\times 3} & 1 \end{bmatrix}, \tag{1}$$

with rotation $R_{\kappa_i} \in SO(3)$ and translation $t_{\kappa_i} \in \mathbb{R}^3$. The local map is expressed in the frame of the most recent keyframe, κ_1, and each older keyframe κ_i is brought into this frame by chaining inverse relative transforms:

$$T^{(i)}_{\text{chain}} = T^{-1}_{\kappa_1} T^{-1}_{\kappa_2} \cdots T^{-1}_{\kappa_{n_\kappa}}, \qquad T^{(0)}_{\text{chain}} = I. \tag{2}$$

The local map is then the union of all transformed keyframe scans:

$$\mathcal{P}_{\text{map}} = \bigcup_{i=1}^{n_\kappa} \left\{ R_{\text{chain}}^{(i)} \, p + t_{\text{chain}}^{(i)} \mid p \in \mathcal{P}_{\kappa_i} \right\}, \tag{3}$$

where $R_{\text{chain}}^{(i)}$ and $t_{\text{chain}}^{(i)}$ are the rotation and translation of $T_{\text{chain}}^{(i)}$.

Step 4: Iterative Closest Point and Keyframe Selection

We use the point-to-point ICP algorithm, which takes as input the current scan, $\mathcal{P}_{ICP}$, the keyframe local map, $\mathcal{P}_{\text{map}}$, and an initial guess for the transformation, $T_{\text{init}} \in SE(3)$, which is the previously determined relative motion constraint derived from IMU preintegration. ICP determines a final transformation, $T_{\text{ICP}} \in SE(3)$, which minimizes the sum of squared point-to-point residuals, thus aligning the point clouds. The keyframe is saved based on whether the rotation or translation are above previously defined thresholds (ϵ_ψ and ϵ_t, respectively).

2.2 Error State Kalman Filter

The INS propagates attitude, velocity, and position in the Earth-Centered, Earth-Fixed (ECEF) frame (superscript e), as described by Groves (2013). In a closed-loop architecture, the filter estimates the error states, along with the accelerometer and gyroscope biases b_a and b_g, which are used to correct the IMU measurements; the nominal states are corrected at the end of each epoch. The implemented architecture is shown in figure 2.

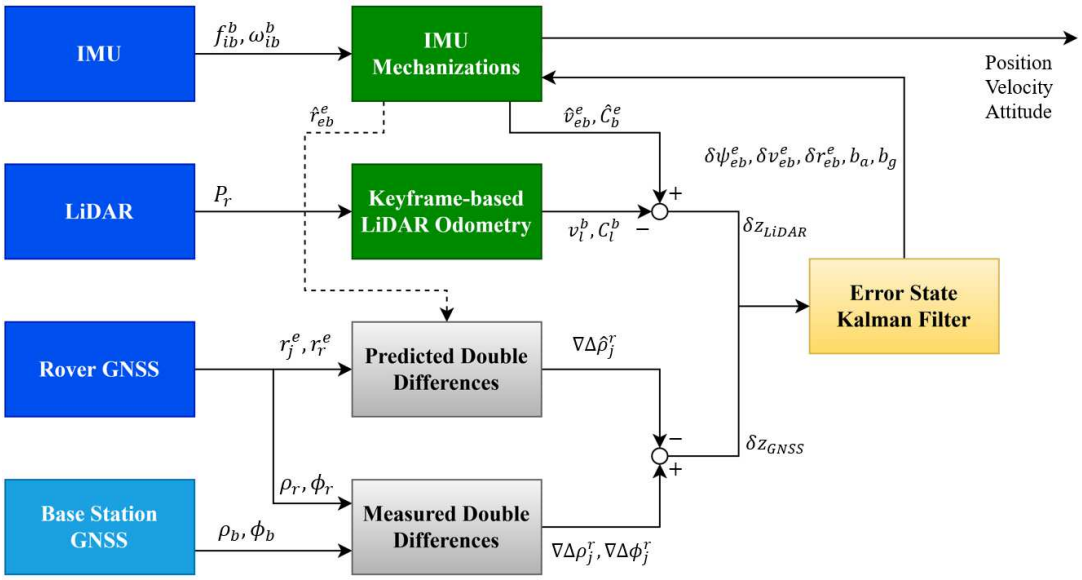


Fig. 2 Block diagram of the implemented INS/GNSS/LiDAR architecture.

State Vector

The state vector for the filter, $\delta\mathbf{x}$, contains the error states for: the INS, including attitude, $\delta\psi_{eb}^e$, velocity, δv_{eb}^e, position, δr_{eb}^e, and IMU biases, b_a and b_g; the observed double-differenced ambiguities between a reference and non-reference satellites, $\delta\nabla\Delta N$; and for a frozen delayed state related to the LiDAR measurement model, $\delta\psi_{eb}^{e-}$. Thus, the state vector is defined as:

$$\delta\mathbf{x} = [\delta\psi_{eb}^e, \delta v_{eb}^e, \delta r_{eb}^e, b_a, b_g, \delta\nabla\Delta N, \delta\psi_{eb}^{e-}]^T \tag{4}$$

State Propagation

The INS error states are propagated during the interval between measurements using the transition and process noise covariance matrices (Φ and Q, respectively) described by Groves (2013). The ambiguity states are assumed constant. The matrices are padded by zeroes to accommodate for the attitude delayed state for the LiDAR measurement model.

2.3 GNSS Measurement Model

To achieve high positioning accuracy, code and carrier phase double differences are used as filter measurements, following the RTK/PPK approach. The formulation follows Groves (2013) unless otherwise noted. The Kalman filter innovation is formed from predicted double differences, derived from the INS-propagated rover position corrected for the IMU–antenna lever arm, $\hat{r}^e_{eb}$, the positions of the jth non-reference satellite, $\tilde{r}^e_j$, and of the reference satellite, $\tilde{r}^e_r$, and the known base station position, r^e_B. The double difference $\nabla\Delta\hat{\rho}^r_j$ is estimated from the single differences for the reference, $\Delta\hat{\rho}^r$, and jth, $\Delta\hat{\rho}^j$, satellites:

$$\begin{aligned}\nabla\Delta\hat{\rho}^r_j = &\underbrace{\left\|\hat{r}^e_{eb} - \tilde{r}^e_r\right\|_2 - \left\|r^e_B - \tilde{r}^e_r\right\|_2}_{\Delta\hat{\rho}^r} \\ &- \underbrace{\left\|\hat{r}^e_{eb} - \tilde{r}^e_j\right\|_2 - \left\|r^e_B - \tilde{r}^e_j\right\|_2}_{\Delta\hat{\rho}^j}\end{aligned} \tag{5}$$

The observed double differences, $\nabla\Delta\tilde{\rho}^r_j$, can thus be used to form the GNSS measurement innovation, δz_{GNSS}, for i satellites as:

$$\delta z_{\text{GNSS}} = \begin{bmatrix} \nabla\Delta\tilde{\rho}^r_1 - \nabla\Delta\hat{\rho}^r_1 \\ \nabla\Delta\tilde{\rho}^r_2 - \nabla\Delta\hat{\rho}^r_2 \\ \vdots \\ \nabla\Delta\tilde{\rho}^r_i - \nabla\Delta\hat{\rho}^r_i \\ \hdashline \nabla\Delta\tilde{\Phi}^r_1 - \lambda\nabla\Delta N^r_1 - \nabla\Delta\hat{\rho}^r_1 \\ \nabla\Delta\tilde{\Phi}^r_2 - \lambda\nabla\Delta N^r_2 - \nabla\Delta\hat{\rho}^r_2 \\ \vdots \\ \nabla\Delta\tilde{\Phi}^r_i - \lambda\nabla\Delta N^r_i - \nabla\Delta\hat{\rho}^r_i \end{bmatrix} \tag{6}$$

where λ corresponds to the wavelength of the used satellite signal, and $\nabla\Delta N^r_j$ denotes the ambiguity of the double difference formed between the reference and jth satellite. The double-differenced unit vector, $\nabla\Delta\mathbf{u}^r_j$, can be determined from the unit vector from the rover to each jth and reference satellite:

$$\nabla\Delta\mathbf{u}^r_j = \underbrace{\frac{\tilde{r}^e_j - \hat{r}^e_{eb}}{\left\|\tilde{r}^e_j - \hat{r}^e_{eb}\right\|_2}}_{\mathbf{u}_j} - \underbrace{\frac{\tilde{r}^e_r - \hat{r}^e_{eb}}{\left\|\tilde{r}^e_r - \hat{r}^e_{eb}\right\|_2}}_{\mathbf{u}_r} \tag{7}$$

The measurement matrix, H_{GNSS}, is formed based on these unit vectors for the code, $H_{\rho}^{\nabla\Delta\mathbf{u}}$, and carrier phase, $H_{\Phi}^{\nabla\Delta\mathbf{u}}$. Assuming that both code and carrier phase measurements use the same satellites, they can be derived as:

$$H_{\rho}^{\nabla\Delta\mathbf{u}} = H_{\Phi}^{\nabla\Delta\mathbf{u}} = \begin{bmatrix} 0_{1,3} & 0_{1,3} & \nabla\Delta\mathbf{u}_1^r & 0_{1,3} & 0_{1,3} \\ 0_{1,3} & 0_{1,3} & \nabla\Delta\mathbf{u}_2^r & 0_{1,3} & 0_{1,3} \\ \vdots & & \vdots & & \vdots \\ 0_{1,3} & 0_{1,3} & \nabla\Delta\mathbf{u}_i^r & 0_{1,3} & 0_{1,3} \end{bmatrix} \tag{8}$$

and thus, based on the number of ambiguities (equal to i number of satellites) the full measurement matrix is derived as:

$$H_{\text{GNSS}} = \begin{bmatrix} H_{\rho}^{\nabla\Delta\mathbf{u}} & 0_{i\times 3} & 0_{i\times 3} \\ H_{\Phi}^{\nabla\Delta\mathbf{u}} & I_i & 0_{i\times 3} \end{bmatrix} \tag{9}$$

The measurement error covariance matrices for the pseudorange, R_{ρ}^{∇}, and carrier phase, R_{Φ}^{∇}, measurements are diagonal matrices based on empirical estimates of pseudorange and carrier phase measurement errors. From this, the GNSS measurement error covariance matrix, R_{GNSS}, can be defined as:

$$R_{\text{GNSS}} = \begin{bmatrix} D_G R_{\rho}^{\nabla} D_G^T & 0 \\ 0 & D_G R_{\Phi}^{\nabla} D_G^T \end{bmatrix}, \qquad D_G = \begin{bmatrix} 1 & 0 & \dots & 0 & -1 \\ 0 & 1 & \dots & 0 & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -1 \end{bmatrix} \tag{10}$$

A two stage normalized innovation squared monitoring scheme based on a chi-squared test is also implemented to detect and handle cycle slips.

2.4 LiDAR Measurement Model

Based on the transformation matrix T_{ICP}, obtained from the described keyframe-based LiDAR Odometry method (subsection 2.1), we derive velocity and attitude measurements to be used in the Kalman filter. From the translation component of T_{ICP} and the scan duration τ_{LiDAR}, we obtain the LiDAR velocity resolved in the body frame, v_l^b:

$$v_l^b = \frac{t_{\text{ICP}}}{\tau_{\text{LiDAR}}} \tag{11}$$

Based on the INS propagated ECEF velocity, $\hat{v}_{eb}^e$, the LiDAR velocity measurement innovation, $\delta z_{\text{LiDAR},v}$, can be derived as:

$$\delta z_{\text{LiDAR},v} = \hat{v}_{eb}^e - \hat{C}_b^e v_l^b \tag{12}$$

where the LiDAR velocity is rotated to the ECEF frame. From the rotation component of T_{ICP} we obtain the inter-scan rotation between the two scans, C_l, which cannot be used directly as an attitude measurement in the filter. Instead, it is composed with a delayed state: the body-to-ECEF attitude estimate, $\hat{C}_b^{e-}$, retained from the previous LiDAR epoch. This

allows us to yield an absolute LiDAR attitude measurement, $C^e_{b,\mathrm{LiDAR}}$:

$$C^e_{b,\mathrm{LiDAR}} = \hat{C}^{e-}_b C_l \tag{13}$$

The attitude residual, A, between the INS-propagated attitude $\hat{C}^e_b$ and the LiDAR attitude measurement is then obtained as:

$$A = \hat{C}^e_b \left(C^e_{b,\mathrm{LiDAR}}\right)^T - I_3 \tag{14}$$

For small angle errors, the skew-symmetric off-diagonal elements of A form the LiDAR attitude measurement innovation, $\delta z_{\mathrm{LiDAR},\psi}$, defined as (Groves 2013):

$$\delta z_{\mathrm{LiDAR},\psi} = \begin{bmatrix} -A_{2,3} \\ A_{1,3} \\ -A_{1,2} \end{bmatrix} \tag{15}$$

From equations (12) and (15), it is thus possible to derive the LiDAR measurement innovation, $\delta z_{\mathrm{LiDAR}}$, as:

$$\delta z_{\mathrm{LiDAR}} = \begin{bmatrix} \delta z_{\mathrm{LiDAR},v} \\ \delta z_{\mathrm{LiDAR},\psi} \end{bmatrix} \tag{16}$$

The measurement matrix, H_{LiDAR}, relates the measurements to the system states. Since the attitude measurement uses the attitude derived from the INS in the previous epoch, the previously introduced delayed attitude state, $\delta\psi^{e-}_{eb}$, must be accounted for in H_{LiDAR}. The measurement error covariance matrix, R_{LiDAR}, is formed from the standard deviations of the velocity and attitude measurements, σ_v and σ_ψ:

$$H_{\mathrm{LiDAR}} = \begin{bmatrix} -C^e_b[v^b_l\times] & I_{3\times3} & 0_{3\times(9+i)} & 0_{3\times3} \\ I_{3\times3} & 0_{3\times3} & 0_{3\times(9+i)} & -I_{3\times3} \end{bmatrix}, \qquad R_{\mathrm{LiDAR}} = \begin{bmatrix} \sigma_v^2 I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \sigma_\psi^2 I_{3\times3} \end{bmatrix} \tag{17}$$

Because LiDAR conditions vary widely, from dense point clouds near riverbanks and vegetation to sparse or empty clouds over open water, LiDAR updates are gated on the current scan's point count. Scans below the threshold are deemed too sparse for reliable odometry and the update is skipped, so the filter propagates on GNSS and inertial data alone.

Measurement Update

The measurement update follows the standard closed-loop error-state Kalman filter formulation (Groves 2013), with the stacked GNSS and LiDAR measurement matrix, noise covariance, and innovation vector given by:

$$H = \begin{bmatrix} H_{\mathrm{GNSS}} \\ H_{\mathrm{LiDAR}} \end{bmatrix}, \quad R = \begin{bmatrix} R_{\mathrm{GNSS}} & 0 \\ 0 & R_{\mathrm{LiDAR}} \end{bmatrix}, \quad \delta z = \begin{bmatrix} \delta z_{\mathrm{GNSS}} \\ \delta z_{\mathrm{LiDAR}} \end{bmatrix}. \tag{18}$$

3 Testing

Two scenarios were selected for airborne data collection. The first, at a river site with tall vegetation and sporadic canopy cover, reflects the UAV's intended operational domain. The second tests abrupt changes in sensor data quality within a single continuous flight, challenging the fusion algorithm: first over an open lake, with near-ideal GNSS but minimal/no LiDAR returns, then along a forested trail under dense canopy, where GNSS is severely degraded but LiDAR returns are abundant.

3.1 River Scenario

Airborne data was collected on the 27th of May 2026 in the Suså river, in Sjælland, Denmark. The specific location of the data collection campaign (55°18'55.0"N, 11°40'31.3"E) was selected due to the fact that the river stretch contains a relevant combination of characteristics, and an adequate area to perform safe takeoff and landing operations. Figure 3 illustrates the test site as well as the reference solution. The drone flew approximately 1 m above the water surface. Total flight time was 470 s (7 min 50 s). A characterization of the test site based on the number of satellites used and the average carrier-to-noise ratio C/N_0 is shown in figure 4.

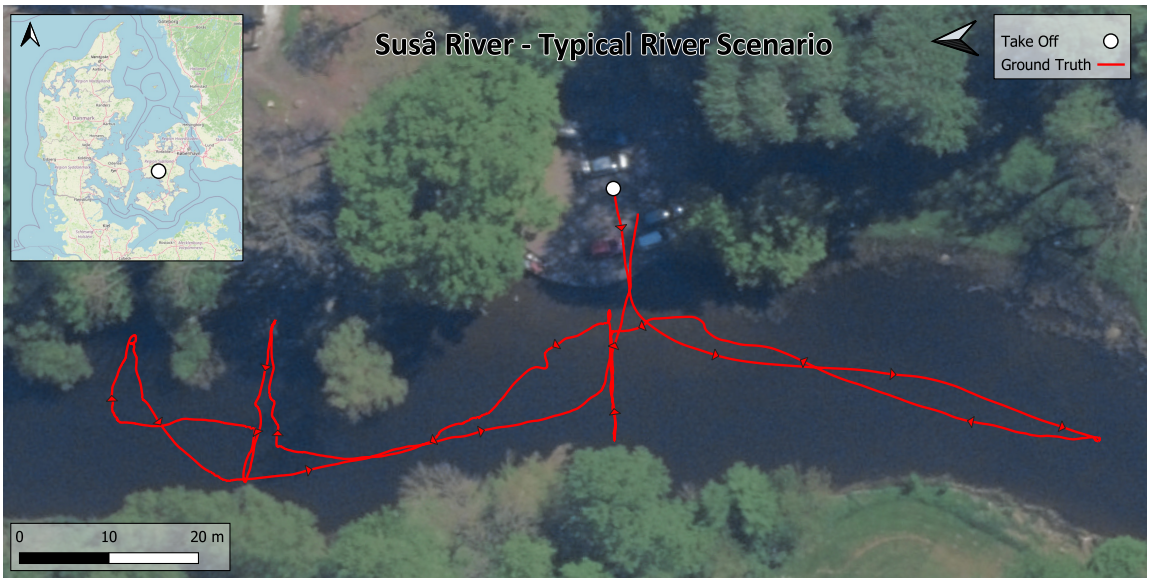


Fig. 3 Test Area in Suså River.

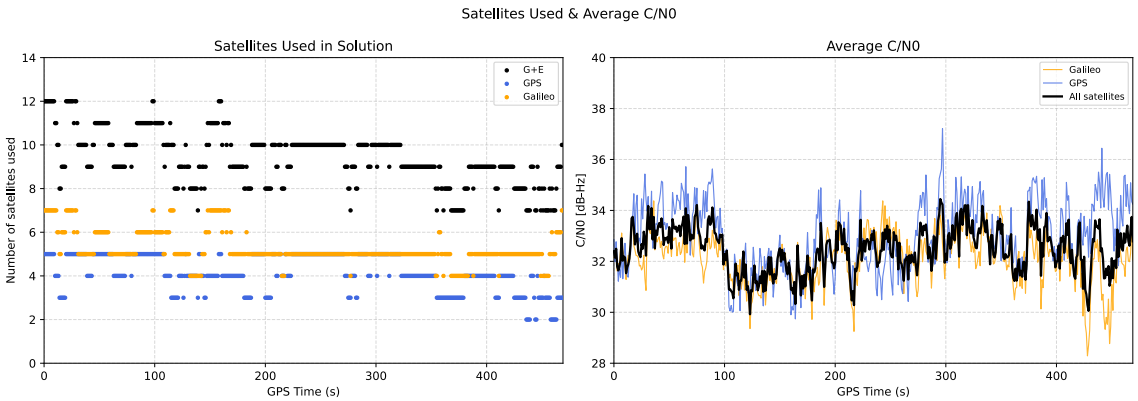


Fig. 4 Number of used satellites and average Carrier-to-Noise Ratio per constellation for the river scenario test site.

3.2 Stress-Test Scenario

The second dataset was collected at the Sjælsø Lake, also in Sjælland, Denmark, on the 1st of July 2026. This location (55°51'38.2"N, 12°26'17.4"E) was selected as it allowed the UAV to fly safely under tree canopies. The test area and reference solution trajectory can be see in figure 5. This site exemplifies a recurrent situation in UAV-based hydrometry: GNSS and LiDAR generally perform best in opposite conditions. Near the banks, vegetation degrades GNSS but returns dense LiDAR point clouds, whereas over open water, such as during higher-altitude flights or over wide rivers, GNSS quality is high but LiDAR returns are sparse or absent. Total flight time was 247 s (4 min 7 s). The number of satellites used and the average carrier-to-noise ratio are shown in figure 6.



Fig. 5 Test Area in Sjælsø Lake.

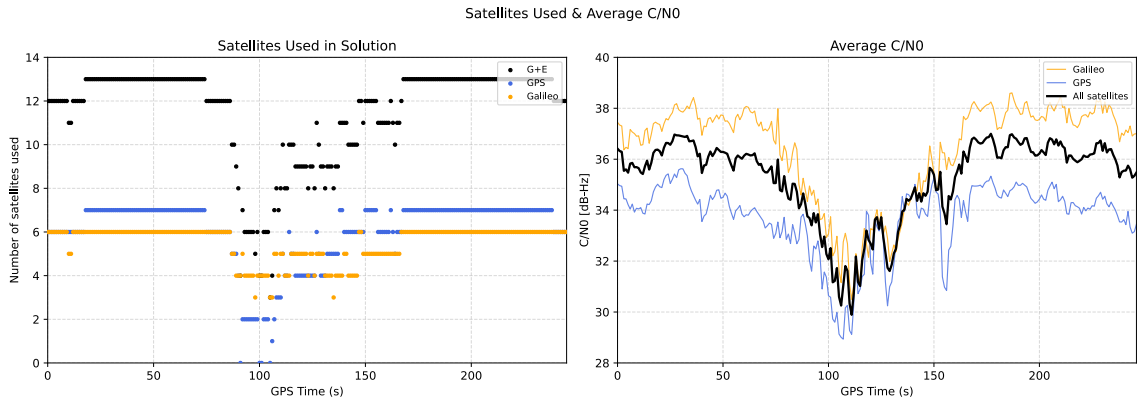


Fig. 6 Number of used satellites and average Carrier-to-Noise Ratio per constellation for the stress-test site.

3.3 Data Collection

For data collection, a UAV was deployed with a payload consisting of a micro-electromechanical system (MEMS) IMU (Xsens MTi-100), LiDAR (Livox Mid-360), and a GNSS receiver (septentrio mosaic-go X-5). The UAV and its payload can be seen in figure 7. The IMU specifications can be seen in table 1. GNSS data was collected from the Galileo and GPS constellations. The GNSS processing parameters can be found in table 2. The rigid transforms between the IMU and LiDAR, and the IMU and GNSS antenna (lever arm) were measured through the use of a CAD model of the UAV.

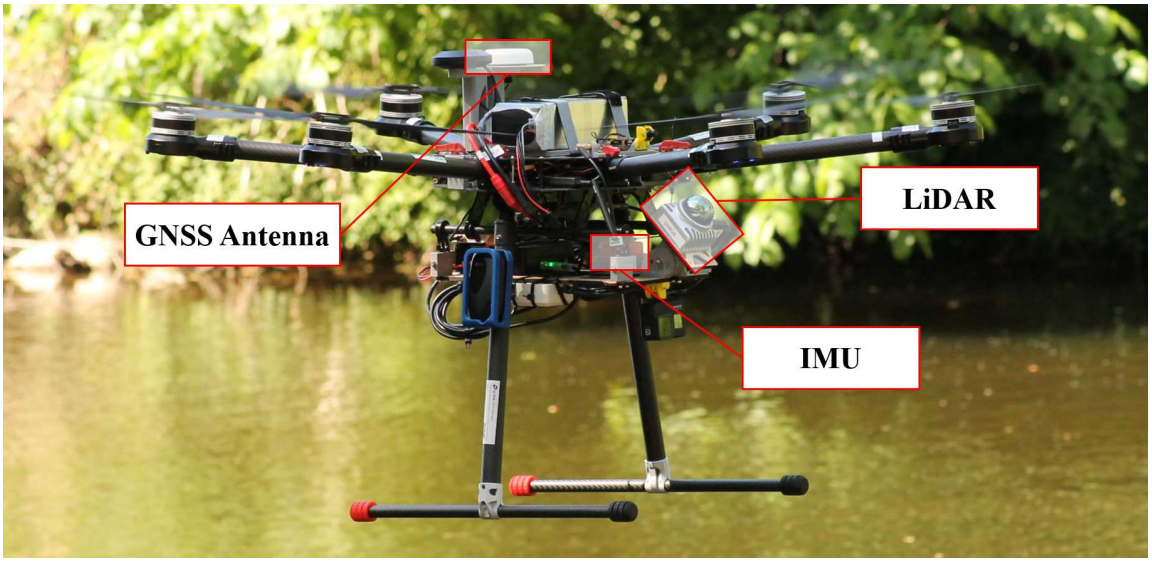


Fig. 7 Experimental UAV used for data collection. The used LiDAR provides 360° scans, and was mounted with a 45° tilt to maximize the returns from the river banks.

Table 1 IMU Specifications (Movella (Xsens) 2023).

Parameter	Specification
Data Rate	100 Hz
Gyro Bias Stability	10 °/h
Gyro Noise Density	0.01 °/s/$\sqrt{\text{Hz}}$
Accelerometer Bias Stability	15 µg
Accelerometer Noise Density	60 µg/$\sqrt{\text{Hz}}$

Table 2 GNSS Processing Parameters.

Parameter	Specification
Constellations	Galileo (E) + GPS (G)
Frequencies	E1/L1, L2, E5a, E5b, E5/L5
Sampling Rate	1 Hz
Elevation Mask	15°
Pseudorange Noise SD	1.0 m
Carrier Phase Noise SD	0.01 cycles
Ambiguity Resolution	Float

Table 3 LiDAR Processing Parameters.

Parameter	Specification
Sampling Rate	10 Hz
Voxel Size, v	1.0
α	0.5
β	2.0
ϵ_ψ	0.1 radians
ϵ_t	0.1 m
Number of Keyframes	4
LiDAR Velocity Noise SD	0.6 m/s
LiDAR Attitude Noise SD	0.1 radians
Minimum Number of Points	4000

In order to ensure time synchronization across all sensors, a Pulse Per Second (PPS) signal provided by the GNSS receiver was used to trigger data capture from the LiDAR and IMU. The parameters used for LiDAR processing can be found in table 3. The reference solution was generated with NovAtel's Inertial Explorer 9.0 (NovAtel Inc. 2023), a commercial INS/GNSS post-processing tool. Forward and reverse filter passes are combined through smoothing to yield the best solution from the available data. All major GNSS constellations (Galileo, GPS, GLONASS and Beidou) were enabled for the reference solution processing.

4 Results & Discussion

Performance metrics including the Root Mean Squared Error (RMSE) and maximum deviation (Max Dev) were computed for the position, velocity and attitude. The solutions computed included:

- **Standalone PPK:** only relevant for position, and to serve as a baseline.
- **INS/GNSS:** a filter using the double-differenced GNSS measurement model (subsection 2.3).
- **LIO-SAM:** a state-of-the-art factor graph feature-based LiDAR Inertial Odometry algorithm aided by RTK GNSS factors proposed by Shan et al. (2020) was included where possible (only considered for position and attitude, as the solution does not output velocity).
- **Feature-based (FB):** a tightly coupled INS/GNSS/LiDAR system based on the double-differenced GNSS model (subsection 2.3), and the feature-matching approach described by Li et al. (2021) for LiDAR measurements.
- **Scan-based (SB):** the INS/GNSS/LiDAR fusion strategy proposed in this work, with tight coupling on GNSS, and loose coupling on LiDAR, following the scan-matching approach previously described.

The two feature-based methods (LIO-SAM and FB) were included in order to assess our assumption that scan-matching yields better results over feature-based methods in natural unstructured environments.

4.1 River Scenario

Position metrics for the river scenario are shown in Table 4. LIO-SAM improves on standalone PPK only in Up RMSE, performing worse in East and North. Despite receiving RTK GNSS factors as global constraints, LIO-SAM relies mainly on LiDAR/IMU odometry, which is likely degraded here by sparse point clouds and a lack of discernible features. INS/GNSS improves substantially on standalone PPK, while the feature-based (FB) solution trades average accuracy for lower peak deviation: relative to INS/GNSS, it has worse RMSE in two of three axes but lower maximum deviation in all. The scan-based solution shows no such trade-off, outperforming both INS/GNSS and FB across all metrics, with improvements over INS/GNSS of 10.2–27.4% in RMSE and 35.5–55.3% in Max Dev. The largest gain is in Up Max Dev, which drops from 0.428 m to 0.191 m.

Table 4 RMSEs and Max Dev of position solutions for the river scenario. Bold denotes lowest values.

	RMSE (m)			Max Dev (m)		
Solution	East	North	Up	East	North	Up
Standalone PPK	0.169	0.315	0.379	1.318	2.834	3.657
INS/GNSS	0.032	0.045	0.075	0.153	0.241	0.428
LIO-SAM	0.965	0.402	0.195	1.584	0.744	0.446
Feature-based	0.035	0.042	0.080	0.118	0.207	0.233
Scan-based	**0.029**	**0.033**	**0.062**	**0.097**	**0.155**	**0.191**
SB Improvement over INS/GNSS (%)	10.2	27.4	17.5	36.8	35.5	55.3

For velocity (Table 5), SB again outperforms INS/GNSS and FB, except in Up Max Dev, where FB performs best, although SB still improves on INS/GNSS by 11.4%.

Table 5 RMSEs and Max Dev of velocity solutions for the river scenario. Bold denotes lowest values.

	RMSE (m/s)			Max Dev (m/s)		
Solution	v_E	v_N	v_U	v_E	v_N	v_U
INS/GNSS	0.015	0.014	0.010	0.083	0.078	0.055
Feature-based	0.014	0.013	0.009	0.076	0.061	**0.039**
Scan-based	**0.010**	**0.011**	**0.008**	**0.038**	**0.049**	0.049
SB Improvement over INS/GNSS (%)	30.2	21.8	24.7	54.8	36.9	11.4

Attitude metrics are shown in Table 6. LIO-SAM is again outperformed by INS/GNSS except in yaw, reinforcing the hypothesis that global RTK GNSS factors cannot compensate for LiDAR/IMU odometry degraded by the sparse, feature-poor river environment. SB outperforms all other solutions, notably improving yaw RMSE by 66.4% over INS/GNSS, while FB improves on INS/GNSS in three of six metrics but has the highest yaw Max Dev (7.057°). The position, velocity and attitude errors for each axis for the river scenario can be seen in figure 8.

Table 6 RMSEs and Max Dev of attitude solutions for the river scenario. Bold denotes lowest values.

	RMSE (deg)			Max Dev (deg)		
Solution	Roll	Pitch	Yaw	Roll	Pitch	Yaw
INS/GNSS	0.056	0.044	2.375	0.236	0.192	5.320
LIO-SAM	0.361	4.403	2.254	2.031	20.659	3.740
Feature-based	0.058	0.058	1.944	0.185	0.194	7.057
Scan-based	**0.045**	**0.035**	**0.799**	**0.176**	**0.163**	**2.242**
SB Improvement over INS/GNSS (%)	18.6	20.5	66.4	25.3	15.2	57.9

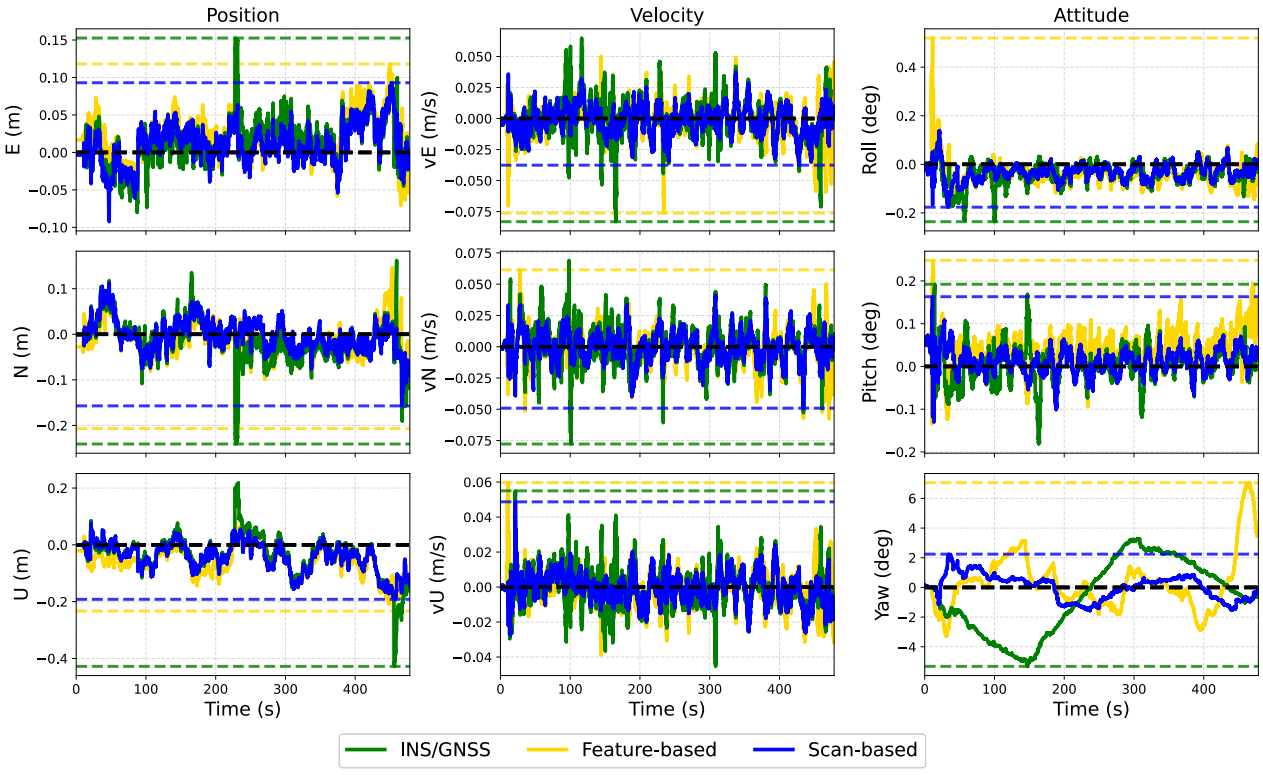


Fig. 8 Error Deviation for position, velocity and attitude for the INS/GNSS, FB and SB solutions for the river scenario. The coloured dashed horizontal lines depict the peak deviation for each solution/axis combination.

4.2 Stress-Test Scenario

The stress-test scenario included portions of the flight where no LiDAR returns were registered. Thus, the LIO-SAM algorithm was incapable of computing a continuos solution through the flight, as it immediately failed after takeoff when the UAV was flown towards the open lake area. Therefore, LIO-SAM results are not included for this scenario.

Table 7 RMSEs and Max Dev of position solutions for the stress-test scenario. Bold denotes lowest values.

	RMSE (m)			Max Dev (m)		
Solution	East	North	Up	East	North	Up
Standalone PPK	0.354	0.531	0.992	2.566	5.141	5.473
INS/GNSS	0.083	0.181	0.242	0.489	0.946	0.430
Feature-based	0.697	0.494	0.911	4.134	2.346	5.125
Scan-based	**0.068**	**0.152**	**0.218**	**0.348**	**0.604**	**0.401**
SB Improvement over INS/GNSS (%)	19.0	16.2	9.6	28.7	36.1	6.7

The position metrics for the stress-test scenario are shown in table 7. The FB solution collapses under challenging GNSS conditions midway through the flight, again suggesting that the lack of discernible features in the vegetated environment is problematic where GNSS is degraded. A compounding factor may be that prop-wash from the rotors perturbed the surrounding vegetation (see figure 9) beyond natural effects such as wind gusts, further hindering feature matching.



Fig. 9 Stress-test flight corridor: a narrow trail under dense canopy, limiting sky visibility and degrading GNSS reception. Propeller wash in the confined space disturbed the vegetation and lifted dust, producing spurious returns and dynamic scene content that challenge LiDAR odometry.

SB outperforms INS/GNSS in all directions, though only marginally in Up: the LiDAR contribution improves the horizontal estimates but constrains height less effectively. We hypothesize that this stems from the limited ground geometry. A well-observed ground plane would strongly constrain height, but the only approximately planar surface is the narrow trail beneath the UAV, flanked by irregular, prop-wash-disturbed vegetation with no coherent horizontal surface. This thin ribbon of usable ground constrains height more weakly than lateral structures such as tree trunks, branches, and trailside vegetation constrain the horizontal position.

Velocity metrics are shown in Table 8, where FB again performs worst, while SB outperforms INS/GNSS in all metrics, improving RMSE by 15.1–36.9% and Max Dev by 10.1–59.9%. For attitude (Table 9), FB again performs worst across all metrics, and SB improves on INS/GNSS with RMSE gains of 15.5–41.9% and a 51.4% reduction in yaw Max

Table 8 RMSEs and Max Dev of velocity solutions for the stress-test scenario. Bold denotes lowest values.

	RMSE (m/s)			Max Dev (m/s)		
Solution	v_E	v_N	v_U	v_E	v_N	v_U
INS/GNSS	0.039	0.029	0.026	0.337	0.149	0.191
Feature-based	0.110	0.192	0.105	0.571	1.321	0.543
Scan-based	**0.025**	**0.025**	**0.018**	**0.135**	**0.134**	**0.084**
SB Improvement over INS/GNSS (%)	36.9	15.1	30.9	59.9	10.1	56.1

Dev. The only exception is pitch Max Dev, where SB regresses. This occurs early in the flight, where the sparse point cloud captures only a small portion of the shoreline, likely providing insufficient geometry to constrain pitch (figure 10).

Table 9 RMSEs and Max Dev of attitude solutions for the stress-test scenario. Bold denotes lowest values.

	RMSE (deg)			Max Dev (deg)		
Solution	Roll	Pitch	Yaw	Roll	Pitch	Yaw
INS/GNSS	0.069	0.061	1.468	0.378	**0.226**	4.137
Feature-based	0.597	0.421	14.644	2.630	2.146	37.512
Scan-based	**0.052**	**0.051**	**0.863**	**0.280**	0.369	**2.012**
SB Improvement over INS/GNSS (%)	24.4	15.5	41.2	26.0	-63.0	51.4

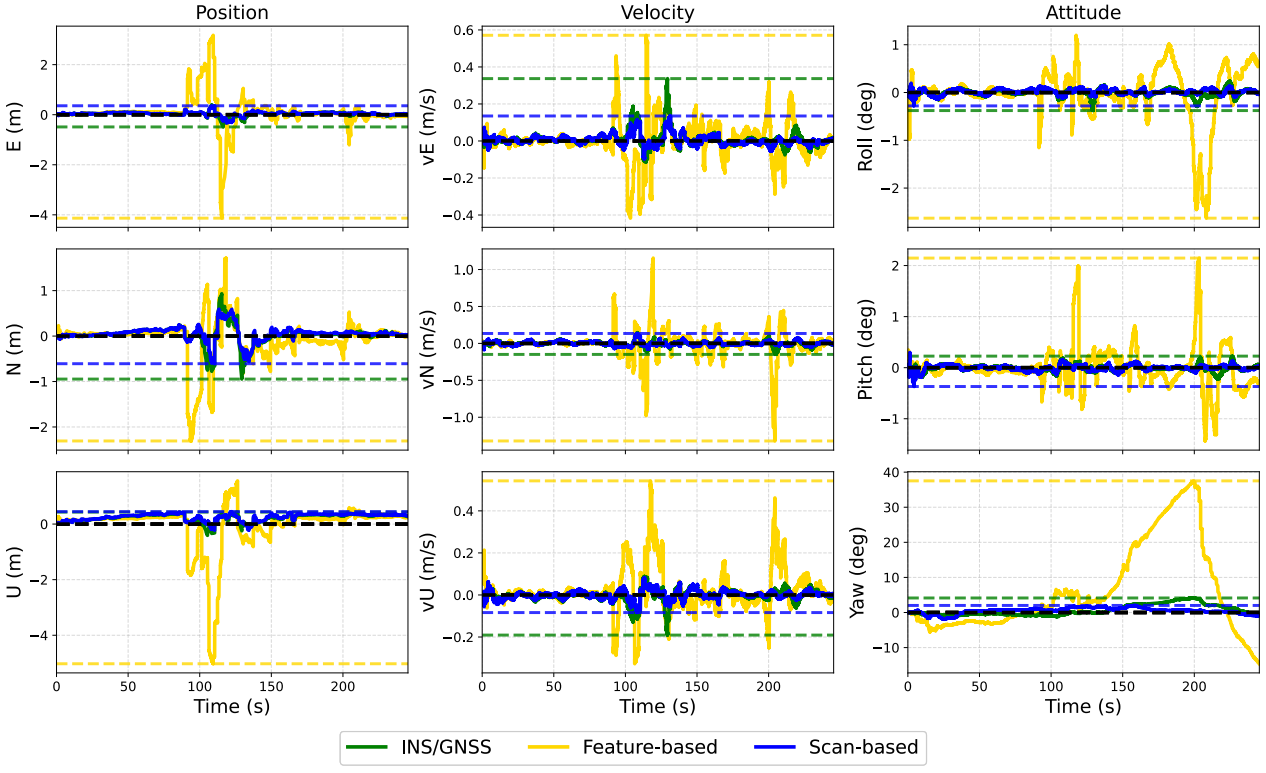


Fig. 10 Error Deviation for position, velocity and attitude for the INS/GNSS, FB and SB solutions for the stress-test scenario. The coloured dashed horizontal lines depict the peak deviation for each solution/axis combination.

4.3 Cross-Scenario Discussion

Across both scenarios, some patterns emerge. In position, the clearest trend is the difference in reliability: SB improves on INS/GNSS in every metric in both scenarios, whereas FB is inconsistent, slightly degrading some RMSE axes in the river scenario and collapsing in the stress test, where its East errors exceed even those of standalone PPK. SB's

improvements in maximum deviation also generally exceed those in RMSE (35.5–55.3% vs. 10.2–27.4% in the river scenario, and in the East and North axes of the stress test), suggesting that the LiDAR mainly bounds worst-case error growth, arresting drift between well-conditioned GNSS epochs, rather than reducing baseline noise. The velocity results reinforce this picture: SB again outperforms INS/GNSS in both scenarios, while FB remains the weakest, consistent with the difficulty of finding reliable feature matches in vegetated and river environments.

Attitude errors are dominated by yaw in every solution, with heading errors over an order of magnitude larger than roll and pitch (e.g., 2.375° yaw versus 0.056° roll and 0.044° pitch for INS/GNSS in the river scenario). Roll and pitch are observable from the gravity vector, whereas heading becomes observable only when the horizontal specific force varies (Groves 2013). The largest SB attitude gains are accordingly in yaw, in both RMSE (66.4% river, 41.2% stress test) and Max Dev (57.9% river, 51.4% stress test), as the LiDAR odometry constrains the heading drift that INS/GNSS alone cannot, yielding the greatest improvement in the least-observable state.

4.4 Computational Performance Evaluation

The core Kalman filter and LiDAR processing algorithms were implemented in *C++* for efficiency, and exposed to the Python pipeline via *pybind11*. Since the implemented method targets real-time operation, per-component processing times were measured for both scenarios on an HP Elite x360 830 notebook with a 13th Gen Intel Core i7-1365UP (1.80 GHz) CPU and 32 GB RAM (see table 10). Each measured component consists of:

- **IMU Mechanization**: computation of the INS mechanization equations.
- **GNSS Processing**: loading pseudorange and carrier phase measurements, computing satellite positions based on ephemerides data, computing double differences from observables.
- **LiDAR Odometry**: loading LiDAR scan, running the keyframe-based LiDAR odometry algorithm.
- **KF Epoch:** running the Kalman filter state propagation and measurement update steps based on the available measurements. Due to differing sampling rates, the Kalman filter can run with either just a LiDAR measurement, or GNSS + LiDAR measurements.

The average epoch falls comfortably within the 100 ms budget in both cases (18.35 ms river, 9.26 ms stress test), and even the highest peaks (94.47 ms and 81.48 ms) remain below budget. LiDAR odometry is the dominant cost in both scenarios, accounting for roughly 76% and 63% of the average epoch, respectively, and thus governs runtime and drives the peaks. The higher river-scenario times reflect the denser point clouds of the vegetated environment, which raise the odometry workload; the stress-test scene is sparser, including stretches with few or no LiDAR returns (e.g. the open lake), lowering the average cost.

5 Conclusion

This work evaluated the proposed INS/GNSS/LiDAR architecture using real-world flight data in scenarios representative of UAV-based river hydrometry. While motivated by this application, the method extends more broadly to unstructured natural environments, i.e. vegetated, feature-poor settings that challenge both GNSS reception and LiDAR odometry. Both datasets capture conditions typical of these contexts: degraded GNSS caused by surrounding vegetation and canopy

Table 10 Average per-epoch processing time for each component and the full epoch, for both scenarios. All values in milliseconds.

Component	River	Stress-test
IMU Mechanization	0.10	0.07
GNSS Processing	1.85	1.42
LiDAR Odometry	14.0	5.83
KF Epoch	2.41	1.93
Average epoch	18.35	9.26
Peak epoch	94.47	81.48

occlusion, together with the sparse, irregular point clouds that arise when LiDAR is absorbed by water or scattered by foliage.

The proposed architecture combines the strengths of two coupling strategies: the tightly coupled relative positioning of the GNSS measurement model and the low implementation cost of a loosely coupled LiDAR measurement model. This keeps the system simple at both ends. The LiDAR odometry reduces to a keyframe-based ICP scan-matching pipeline; the Kalman filter uses a straightforward LiDAR measurement model. The result is a lighter, more easily implemented architecture than feature-based alternatives, with good real-time performance and low computational cost: the method ran comfortably below the per-epoch budget on a standard laptop.

Across both datasets, the scan-based solution demonstrably outperformed all other tested approaches, yielding significant improvements in position, velocity, and attitude. Notably, it outperformed both a state-of-the-art factor-graph method and a filter-based feature-dependent LiDAR approach. LIO-SAM could not derive a solution in segments where no LiDAR point clouds were available, and thus could not transition across the range of naturally occurring conditions encountered in a single flight, from open-sky GNSS to severely degraded GNSS (and, conversely, from empty to dense point clouds). Its LiDAR/INS-anchored architecture also proved difficult to correct globally using RTK GNSS factors (themselves affected by surrounding vegetation) in typical river environments. The filter-based method of Li et al. (2021) was similarly limited, as reliable feature matching is difficult in unstructured natural settings. By avoiding an explicit dependence on discernible features, the proposed method remains robust precisely where these alternatives falter.

Nonetheless, the method has limitations. Most notably, the height estimate remains the weakest compared with the reference solution: for a UAV flying roughly 0.5 m above the water surface, the Up maximum deviation (0.19 m) is still substantial, and the LiDAR appears unable to constrain height drift as strongly as it constrains the horizontal plane. A second practical limitation concerns tuning. Establishing suitable filter parameters required testing several values and inspecting the LiDAR innovation sequences; likewise, the α and β parameters of the keyframe-based odometry could likely be tuned per scenario, but were deliberately held constant across both datasets to ensure a fair comparison.

Several avenues remain open for future work. First, GNSS ambiguities are currently estimated as float rather than fixed, owing to the computational cost of fixing; exploiting LiDAR to aid ambiguity resolution, as in Li et al. (2023), is a promising direction and should improve positional accuracy. Second, the vertical limitation noted above could be mitigated with a downward-facing sensor, such as a mmWave radar, giving direct height measurements relative to the water surface; this would alter the scan-based architecture, but could substantially improve height estimation. Third, the keyframe-based local map built for odometry already encodes nearby structures and could be reused for real-time obstacle detection, a key enabler for automated flight in confined, cluttered environments. Finally, although the proposed

algorithms are real-time capable, they have not yet been deployed as the active navigation solution in a UAV flight controller, the natural next step towards operational use. The implemented solution is also relevant for Beyond Visual Line of Sight (BVLOS) surveying, so this could be a relevant use case within hydrometric surveying to perform tests/active deployment.

Overall, our results demonstrate that the proposed and implemented INS/GNSS/LiDAR architecture based on scan-matching provides an accurate, robust, and computationally lightweight navigation solution for UAVs operating in the degraded-GNSS conditions typical of unstructured natural environments, making it well suited for airborne hydrometric surveying.

Declarations

The authors declare no conflicts of interest. The research leading to these results received funding from the Unmanned Airborne Water Observing System (UAWOS) Horizon Europe project under Grant Agreement No. 101091783.

Acknowledgments

The authors would like to thank Benjamin Elsholm and Pablo León Barriga for their invaluable support during the data collection campaigns for this work. The authors would also like to thank Søren Reime Larsen for providing a GNSS toolbox which has been used extensively alongside the developed algorithms.

References

Bandini F, Jakobsen J, Olesen D, et al (2017) Measuring water level in rivers and lakes from lightweight unmanned aerial vehicles. J Hydrol 548:237–250. https://doi.org/10.1016/j.jhydrol.2017.02.038

Bandini F, Lopez-Tamayo A, Merediz-Alonso G, et al (2018) Unmanned aerial vehicle observations of water surface elevation and bathymetry in the cenotes and lagoons of the Yucatan Peninsula, Mexico. Hydrogeol J 26(7):2213–2228. https://doi.org/10.1007/s10040-018-1755-9

Bandini F, Kooij L, Mortensen BK, et al (2023) Mapping inland water bathymetry with ground penetrating radar (GPR) on board unmanned aerial systems (UASs). J Hydrol 616:128789. https://doi.org/10.1016/j.jhydrol.2022.128789

Besl PJ, McKay ND (1992) A method for registration of 3-d shapes. IEEE Trans Pattern Anal Mach Intell 14(2):239–256. https://doi.org/10.1109/34.121791

Chambers A, Achar S, Nuske S, et al (2011) Perception for a river mapping robot. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp 227–234, https://doi.org/10.1109/IROS.2011.6095040

Chambers A, Scherer S, Yoder L, et al (2014) Robust multi-sensor fusion for micro aerial vehicle navigation in GPS-degraded/denied environments. In: 2014 American Control Conference. IEEE, pp 1892–1899, https://doi.org/10.1109/ACC.2014.6859341

Cheng J, Zhang X, Zhu F, et al (2025) A tightly coupled integration of GNSS/IMU/LiDAR with parameterized semantic line and plane features to improve pose accuracy in complex environments. Measurement 247:116843. https://doi.org/10.1016/j.measurement.2025.116843

Fabrício AC, Joerger M, Bauer-Gottwein P, et al (2025) INS/GNSS/LiDAR fusion for accurate positioning of UAVs in complex river environments. In: Proceedings of the 38th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2025). Institute of Navigation, pp 2011–2020, https://doi.org/10.33012/2025.20449

Forster C, Carlone L, Dellaert F, et al (2015) IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In: Robotics: Science and Systems XI. Robotics: Science and Systems Foundation, https:

//doi.org/10.15607/RSS.2015.XI.006

Groves PD (2013) Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, 2nd edn. Artech House, Boston

He L, Jin Z, Gao Z (2020) De-skewing LiDAR scan for refinement of local mapping. Sensors 20(7):1846. https://doi.org/10.3390/s20071846

Höfle B, Vetter M, Pfeifer N, et al (2009) Water surface mapping from airborne laser scanning using signal intensity and elevation data. Earth Surf Process Landf 34(12):1635–1649. https://doi.org/10.1002/esp.1853

Lee D, Jung M, Yang W, et al (2024) LiDAR odometry survey: recent advancements and remaining challenges. Intell Serv Robot 17(2):95–118. https://doi.org/10.1007/s11370-024-00515-8

Leick A, Rapoport L, Tatarnikov D (2015) GPS Satellite Surveying, 4th edn. John Wiley & Sons, Hoboken, New Jersey, https://doi.org/10.1002/9781119018612

Li X, Wang H, Li S, et al (2021) GIL: a tightly coupled GNSS PPP/INS/LiDAR method for precise vehicle navigation. Satell Navig 2(1):26. https://doi.org/10.1186/s43020-021-00056-w

Li X, Wang S, Li S, et al (2023) Enhancing RTK performance in urban environments by tightly integrating INS and LiDAR. IEEE Trans Veh Technol 72(8):9845–9856. https://doi.org/10.1109/TVT.2023.3257874

Movella (Xsens) (2023) MTi-100 Series Datasheet. URL https://www.xsens.com/hubfs/Downloads/Leaflets/MTi-100.pdf, accessed: July 2026

NovAtel Inc. (2023) Waypoint Software 9.00 User Manual. NovAtel Inc. / Hexagon Autonomy & Positioning, v11 edn., URL https://hexagondownloads.blob.core.windows.net/public/Novatel/assets/Documents/Manuals/Waypoint-Software-User-Manual-OM-20000166/Waypoint_9.00_Software_User_Manual.pdf, accessed: July 2026

Nuske S, Choudhury S, Jain S, et al (2015) Autonomous exploration and motion planning for an unmanned aerial vehicle navigating rivers. J Field Robot 32(8):1141–1162. https://doi.org/10.1002/rob.21596

Olesen D, Jakobsen J, Knudsen P (2017) Ultra-tightly coupled GNSS/INS for small UAVs. In: Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017). Institute of Navigation, pp 2587–2602, https://doi.org/10.33012/2017.15282

Rehder J, Gupta K, Nuske S, et al (2012) Global pose estimation with limited GPS and long range visual odometry. In: 2012 IEEE International Conference on Robotics and Automation. IEEE, pp 627–633, https://doi.org/10.1109/ICRA.2012.6225277

Shan T, Englot B, Meyers D, et al (2020) LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp 5135–5142, https://doi.org/10.1109/IROS45743.2020.9341176

Vizzo I, Guadagnino T, Mersch B, et al (2023) KISS-ICP: In defense of point-to-point ICP – simple, accurate, and robust registration if done the right way. IEEE Robot Autom Lett 8(2):1029–1036. https://doi.org/10.1109/LRA.2023.3236571

Wan G, Yang X, Cai R, et al (2018) Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp 4670–4677, https://doi.org/10.1109/ICRA.2018.8461224

Wu Y, Guadagnino T, Wiesmann L, et al (2024) LIO-EKF: High frequency LiDAR-inertial odometry using extended Kalman filters. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp 13741–13747, https://doi.org/10.1109/ICRA57147.2024.10610667

Xu W, Cai Y, He D, et al (2022) FAST-LIO2: Fast direct LiDAR-inertial odometry. IEEE Trans Robot 38(4):2053–2073. https://doi.org/10.1109/TRO.2022.3141876

Yang J, Dani A, Chung SJ, et al (2017) Vision-based localization and robot-centric mapping in riverine environments. J Field Robot 34(3):429–450. https://doi.org/10.1002/rob.21606

Zhang J, Singh S (2014) LOAM: Lidar odometry and mapping in real-time. In: Robotics: Science and Systems X. Robotics: Science and Systems Foundation, https://doi.org/10.15607/RSS.2014.X.007

Zhou J, Guillemaille T, Meurie C, et al (2025) Scalable GNSS performance assessment and environment categorization under vegetation. In: 2025 IEEE/ION Position, Location and Navigation Symposium (PLANS). IEEE, pp 1204–1215, https://doi.org/10.1109/PLANS61210.2025.11028472

Zhou Z, Riis-Klinkvort L, Jørgensen EA, et al (2024) Measuring river surface velocity using UAS-borne Doppler radar. Water Resour Res 60(11). https://doi.org/10.1029/2024WR037375