ISPRS Open Journal of Photogrammetry and Remote Sensing 19 (2026) 100109

Contents lists available at ScienceDirect

ISPRS Open Journal of Photogrammetry and Remote Sensing

journal homepage: www.journals.elsevier.com/isprs-open-journal-of-photogrammetry-and-remote-sensing

Direct 3D mapping with a 2D LiDAR using sparse reference maps

Eugeniu Vezeteu [a,b,*], Aimad El Issaoui [a], Heikki Hyyti [a], Jesse Muhojoki [a], Petri Manninen [a], Teemu Hakala [a], Eric Hyyppä [a], Antero Kukko [a], Harri Kaartinen [a], Ville Kyrki [b], Juha Hyyppä [a]

[a] Finnish Geospatial Research Institute (FGI), National Land Survey of Finland, Remote Sensing and Photogrammetry, Espoo, 02150, Uusimaa, Finland
[b] Aalto University, Electrical Engineering and Automation, Espoo, 02150, Uusimaa, Finland

ARTICLE INFO

Keywords:
LiDAR-based mapping
Airborne laser scanning (ALS)
Mobile laser scanning (MLS)
Sensor fusion
Extrinsic calibration
Localisation
Remote sensing

ABSTRACT

Precise 3D mapping is crucial for a wide range of geospatial applications, including forest monitoring, infrastructure assessment, and autonomous navigation. While 2D Light Detection and Ranging (LiDAR) sensors offer superior range accuracy and higher point density compared to many 3D LiDARs, their limited sensing geometry makes full 3D reconstruction challenging. In this paper, we address these limitations and achieve robust 3D mapping by proposing a direct method for integrating 2D LiDAR with a 6 Degrees of Freedom (DoF) trajectory and sparse 3D reference maps derived from mobile laser scanning (MLS) or airborne laser scanning (ALS). Our method begins with an initial 6 DoF trajectory and performs batch optimisation by jointly co-registering buffered 2D LiDAR scans to a 3D reference map, enhancing both trajectory accuracy and mapping completeness without relying on 2D scans' overlap or segmentation. We also introduce a novel, targetless extrinsic calibration approach between 2D LiDAR, 3D LiDAR, and a Global Navigation Satellite System–Inertial Navigation System (GNSS–INS) system that does not rely on overlapping sensor Field of View (FOV). We validate our approach in forest road environments using sparse ALS or MLS reference maps and initial poses from GNSS–INS or 3D LiDAR-inertial odometry. Experiments in forest roads achieved mean localisation accuracies of 0.1 m (using 3D MLS initialisation) and 0.16 m (using GNSS–INS initialisation), reducing drift by up to nine times in translation and six times in rotation. The extrinsic calibration method converges even with initial misalignments of up to 40° in rotation and 3 m in translation. The proposed framework enables multi-platform, multi-temporal data fusion, offering a practical solution for field deployment and map correction tasks.

1. Introduction

Accurate and up-to-date 3D mapping is essential for numerous geospatial applications, including forest monitoring, infrastructure inspection, and autonomous navigation (Hyyppä et al., 2021; Polewski et al., 2019; Dai et al., 2019). As geospatial technologies evolve, there is a growing demand to integrate point cloud data collected from heterogeneous sensors and platforms across different times (Kukko et al., 2012; Vezeteu et al., 2025). Depending on the operational context and required resolution, various Light Detection and Ranging (LiDAR) systems are employed to acquire 3D point clouds. Among them, airborne laser scanning (ALS) systems, typically mounted on helicopters or drones, offer wide-area coverage and benefit from reliable Global Navigation Satellite System and Inertial Navigation System (GNSS–INS) positioning due to minimal occlusions from buildings or vegetation (Wehr and Lohr, 1999; Baltsavias, 1999; Matikainen et al., 2019). However, ALS systems often struggle with occlusions in dense environments like forests, where tree trunks can be shadowed by the canopy. Yet, mobile laser scanning (MLS) systems, mounted on ground vehicles, trolleys, or backpacks, provide detailed data from ground level and a different field of view (FOV), but suffer from degraded GNSS–INS performance in GNSS-challenged environments, such as under dense tree cover or near tall buildings (Puente et al., 2011; Kukko et al., 2012; Liang et al., 2014; Cheng et al., 2015; Kaartinen et al., 2015a;

* Corresponding author at: Finnish Geospatial Research Institute (FGI), National Land Survey of Finland, Remote Sensing and Photogrammetry, Espoo, 02150, Uusimaa, Finland.
E-mail addresses: eugeniu.vezeteu@maanmittauslaitos.fi (E. Vezeteu), aimad.elissaoui@maanmittauslaitos.fi (A. El Issaoui), heikki.hyyti@maanmittauslaitos.fi (H. Hyyti), jesse.muhojoki@maanmittauslaitos.fi (J. Muhojoki), petri.manninen@maanmittauslaitos.fi (P. Manninen), teemu.hakala@nls.fi (T. Hakala), eric.hyyppa@maanmittauslaitos.fi (E. Hyyppä), antero.kukko@nls.fi (A. Kukko), harri.kaartinen@nls.fi (H. Kaartinen), ville.kyrki@aalto.fi (V. Kyrki), juha.hyyppa@nls.fi (J. Hyyppä).

https://doi.org/10.1016/j.ophoto.2025.100109
Received 1 September 2025; Received in revised form 7 November 2025; Accepted 25 November 2025
Available online 28 November 2025

Sigrist et al., 1999; Feng et al., 2021). In the absence of reliable absolute positioning, MLS systems that rely solely on LiDAR, IMU, or camera sensors accumulate drift over time. A promising approach to address this is to use ALS-derived maps as geospatial priors for MLS trajectories. The fusion of 3D MLS and ALS data has been shown to improve both map completeness (Hyyppä et al., 2020a, 2021; Muhojoki et al., 2024) and trajectory accuracy (Lee et al., 2022; Vezeteu et al., 2025).

Various sensors are installed on the MLS and ALS platforms. Among them, 2D and 3D LiDAR sensors offer complementary advantages. While modern 3D LiDARs (e.g., rotating Ouster, Velodyne, Hesai) can capture rich geometric details, especially for georeferencing, they often have lower point density and range accuracy than 2D LiDARs such as the Riegl VUX series. Conversely, 2D LiDARs offer high accuracy and point density, but they lack the full 3D coverage required for registration and, therefore cannot directly recover 6 degrees of freedom (DoF) motion, making accurate 3D reconstruction with a 2D sensor an ill-posed problem. Several approaches have attempted to overcome this limitation. One approach is to install 2D LiDAR sensors on rotating platforms to acquire 3D coverage of the measured scene (Kang et al., 2018; Schadler et al., 2014; Bosse and Zlot, 2009), though this introduces motion artefacts and synchronisation challenges. Alternatively, a high-accuracy 6 DoF trajectory, for example, from GNSS–INS, can be used to project 2D scans into 3D space (Li et al., 2024a; Lin et al., 2012; Chen et al., 2017; Roca et al., 2016). In this case, the accuracy of the final map depends heavily on both the quality of the trajectory and the extrinsic calibration of the sensors. A notable method in forest mapping (Muhojoki et al., 2024; Kukko et al., 2017) uses segmented tree positions from ALS as references to refine MLS-based mapping using a 2D sensor. However, this approach assumes that the 2D sensor can observe curved tree arcs, which requires it to be oriented closely perpendicular to tree trunks. In our case, the 2D sensor is mounted to scan the ground directly for road-monitoring purposes, making tree arc detection infeasible.

In contrast to prior methods, we propose a direct approach for 3D mapping using a 2D LiDAR without the need for segmentation, feature extraction, or specific sensor mounting configurations. Starting with an initial 6 DoF trajectory estimate, our method registers the 2D scan data into a 3D map and then refines the trajectory through alignment with a sparse reference map (ALS or MLS) using batch update corrections. Here, a batch refers to a list of sequential 2D scans collected over a short time window, and batch updates involve co-registering all scans in the batch simultaneously, rather than one at a time. This joint optimisation leverages the spatial structure of the reference map to enforce geometric consistency across the entire batch, even when individual scans have limited or non-overlapping FOV. Additionally, we introduce a direct method for extrinsic calibration between a 2D LiDAR, a 3D LiDAR, and a 6 DoF system (GNSS–INS in our case). Our approach does not require calibration targets or overlapping sensor FOV, making it practical for field deployment.

We evaluate our proposed method using two types of initial trajectory estimates: one derived from a Post-Processed Kinematic (PPK) GNSS–INS system (NovAtel, 2024), and the other from a 3D MLS system with LiDAR-inertial navigation that uses ALS maps as priors (Vezeteu et al., 2025). For reference maps, we tested one MLS-based 3D map and two different ALS-based maps. Our system performs registration in a buffered (batch) manner, where neighbouring 2D scans mutually support each other's alignment through the constraints imposed by the reference map. We also conduct an ablation study to evaluate the model's robustness to initial trajectory noise and measurement uncertainties. The proposed approach improves both map completeness and trajectory precision, enables multi-platform and multi-temporal data fusion, and applies to both georeferencing new data and correcting old maps with poor alignment.

The paper's main contributions are as follows:

- **Direct 3D mapping with 2D LiDAR:** We propose a method that reconstructs accurate 3D maps from 2D LiDAR scans using an initial 6 DoF trajectory and a sparse reference map. Jointly registering buffered scans improves both trajectory accuracy and map completeness, without assumptions of sensor placement, scan overlap, or environment structure.
- **Targetless extrinsic calibration:** We introduce a method to calibrate 2D LiDAR, 3D LiDAR, and GNSS–INS systems without artificial targets or overlapping fields of view. It is sufficient that they observe the same environment at different times from a moving platform.
- **Comprehensive experimental validation:** We evaluate the system in complex forest environments using MLS- and ALS-based reference maps and conduct ablation studies to assess robustness to trajectory noise and measurement uncertainty.

2. Related work

Point cloud registration is the process of aligning multiple scans into a common reference frame, which may be local or global. This alignment enables the estimation of the scanner's motion over time, commonly referred to as LiDAR odometry (LO). In contrast, georeferencing refers to assigning absolute coordinates, typically in a global coordinate system, to the point clouds. Registration can be performed by aligning consecutive scans directly, while georeferencing can be achieved independently using external positioning systems such as GNSS–INS trajectories. In practice, systems such as MLS platforms often integrate these approaches by fusing 2D LiDARs, 3D LiDAR, GNSS, and IMU data to perform both registration and georeferencing in a unified framework. In the following, we present the extrinsic calibration method for 2D LiDAR, 3D LiDAR, and GNSS–INS, and then describe how these sensors are fused for odometry and mapping.

2.1. Extrinsic calibration

To fuse the 2D LiDAR, 3D LiDAR, and GNSS–INS data, it is essential to know the spatial extrinsic transformations, that is, the rigid transformations between each pair of sensor frames. The transformation between the 3D LiDAR and GNSS–INS is typically estimated using motion-based methods, also known as hand-eye calibration techniques (Horn et al., 2021; Horaud and Dornaika, 1995; Välimäki et al., 2023). In this approach, relative motions from both sensor trajectories are used to solve for the static extrinsic transformation. However, this method assumes low-noise motion estimates from both trajectories and sufficient excitation in all DoFs, which is often unrealistic for ground vehicles. GNSS-based motion, in particular, may be noisy, and these methods frequently require refinement using reference maps, such as ALS or MLS (Vezeteu et al., 2025; Nie et al., 2023; Pi et al., 2024).

In contrast, estimating the extrinsics for 2D LiDAR is more challenging. The 2D LiDAR scanner lacks full 6 DoF pose information, and its narrow FOV often does not overlap with other sensors. Consequently, conventional target-based calibration methods (Fernández-Moral et al., 2015; Vezeteu, 2020; Yin et al., 2018) cannot be directly applied. The most basic alternative is to physically measure the relative position and orientation between sensors. However, accurate manual measurement is difficult. Another option is to rely on Computer-Aided Design (CAD) models to extract extrinsics during system design. Yet, this requires precise and complete 3D modelling of the setup, which is often challenging.

Some studies propose the estimation of 2D LiDAR extrinsics by leveraging the motion of the platform. For example, Maddern et al. (2012) estimate the extrinsic transformation between a 2D and 3D LiDAR by first georeferencing a 3D point cloud using a GNSS–INS trajectory. It then formulates the calibration as an entropy minimisation problem, optimising the extrinsic parameters to sharpen the resulting point cloud combined from 3D and 2D LiDAR measurements, thereby improving alignment between the sensors. Napier et al. (2013) calibrate a 2D LiDAR and camera pair by maximising mutual information between the LiDAR's reflectivity and the camera's intensity image.

2.2. 3D LiDAR registration and fusion: MLS, ALS, and bundle adjustment

A fundamental method for point cloud registration is the Iterative Closest Point (ICP) algorithm (Zhang, 2014; Vizzo et al., 2023), which estimates the rigid transformation between two point clouds by minimising distances between nearest-neighbour correspondences, typically identified using spatial indexing structures such as KD trees or hash maps (Vizzo et al., 2023). The original ICP algorithm minimises a point-to-point error, directly pulling each source point towards its corresponding target point (Zhang, 2014; Vizzo et al., 2023). An important extension of this idea is the Generalised ICP (Segal et al., 2009), which models local neighbourhoods as distributions and minimises a point-to-distribution error. This can be interpreted as a weighted point-to-point error, where the weights are derived from the local covariance of the source and target points, capturing surface structure and improving robustness in noisy or unevenly sampled regions. To further enhance accuracy and efficiency, several other extensions of ICP have been proposed. Voxel-based methods (Yuan et al., 2022) aggregate points in spatial grids to reduce redundancy and improve numerical stability. LiDAR odometry and mapping (LOAM) algorithms (Zhang and Singh, 2014; Shan and Englot, 2018; Wang et al., 2020; Zhou et al., 2021) extend beyond pairwise point correspondences by extracting geometric features, typically edges and planar regions, and formulating cost functions that minimise point-to-feature distances. Local geometry is estimated using Principal Component Analysis (PCA) on the nearest neighbour points, where a dominant eigenvalue indicates a linear (edge) structure and a small eigenvalue indicates a planar surface. These point-to-line and point-to-plane constraints enable more accurate and stable registration, particularly in structured environments.

A major advancement over pure LiDAR registration is the integration of inertial measurements via LiDAR-inertial odometry (LIO). In this paradigm, IMU pre-integration corrects for motion distortion of the point cloud and offers an initial pose guess, while LiDAR observations help compensate for IMU biases and gravity. Fusion pipelines generally fall into two categories: Kalman filter-based estimators and pose graph optimisation frameworks. For example, LIO-SAM (Shan et al., 2020) uses a factor graph formulation (based on iSAM2 Kaess et al., 2011) for joint state and bias optimisation. FAST-LIO and FAST-LIO2 (Xu and Zhang, 2020; Xu et al., 2022) achieve real-time performance through an Iterated Extended Kalman Filter (IEKF) on a Riemann manifold (He et al., 2021).

To further reduce the trajectory drift, bundle adjustment (BA) methods have been adapted to LiDAR mapping. Inspired by computer vision, BA jointly refines a sequence of poses by minimising geometric error terms such as point-to-plane or point-to-edge distances. Recent studies such as Huang et al. (2021), Liu and Zhang (2021), Liu et al. (2024), Li et al. (2024b) demonstrate that local BA over sliding windows of scans can significantly improve map consistency.

Even with LIO and BA techniques, accumulated drift introduces systematic positional errors, particularly over long trajectories. A common strategy is to incorporate absolute GNSS measurements to provide global positional corrections. However, GNSS positional performance degrades in dense forests, urban canyons, or other occluded environments.

To overcome GNSS positional limitations, researchers have explored fusing MLS data with ALS. Since ALS is typically acquired from higher altitudes with fewer obstructions hampering the GNSS signal, it offers a reliable global reference. In urban environments, the approaches of Cheng et al. (2015) and Cheng et al. (2018) align MLS and ALS point clouds by extracting structural features like roads and building outlines. Tree-based methods dominate registration in natural environments, where trees serve as key landmarks (Lee et al., 2022; Zhou et al., 2023; Muhojoki et al., 2024; Hyyppä et al., 2021; Kukko et al., 2017). Recently, a general solution was proposed by Vezeteu et al. (2025), who developed a direct online method for co-registering ALS and MLS data without relying on specific features, achieving good results in both forested and urban settings.

However, most existing MLS-ALS fusion methods come with important limitations: they often assume the use of 3D LiDAR sensors, require significant spatial overlap between consecutive measurements, or depend on the observation of specific features like tree stems. These assumptions restrict their applicability, especially when adapting to diverse sensor configurations or environments.

2.3. 3D mapping with 2D LiDAR sensors

While 3D LiDAR sensors are becoming more accessible, 2D LiDAR remains attractive for 3D mapping tasks, due to its higher point density and better range accuracy. Methods for creating 3D maps from 2D LiDAR data can be categorised into two main strategies, known as mechanical transformation and trajectory-based projection.

In approaches based on mechanical transformation, the 2D LiDAR is physically rotated to mimic 3D scanners. Systems such as those described in Kang et al. (2018), Schadler et al. (2014) mount 2D LiDARs on servo motors or spinning platforms to sweep out a 3D volume. While effective, these setups are sensitive to mechanical inaccuracies, synchronisation issues, and motion artefacts, which can degrade the mapping quality. Continuous spin-matching approaches (Bosse and Zlot, 2009) further extend this idea but remain prone to distortions induced by the rotation.

Alternatively, 2D LiDARs can be combined with 6 DoF motion estimation systems, such as visual-inertial odometry or GNSS–INS trajectories, to georeference 2D scans into 3D space. Systems such as those from Li et al. (2024a), Lin et al. (2012) augment 2D LiDAR with visual sensors to build dense 3D coloured maps. Other studies, such as (Chen et al., 2017), fuse 2D LiDAR, visual SLAM, and IMU data for real-time 3D reconstruction. A more direct approach is to georeference 2D LiDAR scans solely using PPK GNSS–INS trajectories. In aerial mapping, Roca et al. (2016) mounted 2D LiDARs on UAVs and reconstructed 3D maps from GNSS–INS trajectories. While this is feasible outdoors in principle, studies such as Muhojoki et al. (2024) and Kukko et al. (2017) show that without refinement, GNSS–INS-only mapping often suffers from drift-induced distortions caused by an occluded GNSS signal, making post-processing necessary for accurate results. Kaartinen et al. (2015b) reported that under forest canopies, the GNSS–INS system achieved a root mean square error (RMSE) of 0.7 m. In summary, 3D mapping with 2D LiDAR sensors offers a flexible alternative to 3D systems but faces challenges related to mechanical complexity, motion estimation accuracy, and the positioning of the sensor during scanning.

3. Method

The task of 3D mapping by georeferencing 2D LiDAR scans with minimal or no overlap onto a sparse 3D reference map is challenging, as individual 2D scans lack sufficient spatial constraints for reliable registration, often leading to alignment failures. Although an initial registration guess is available, it may contain unknown errors that cause drift and distortions of the point cloud.

To address these challenges, we leverage an initial 6 DoF trajectory estimate and accumulate a buffer of 2D LiDAR scans for registration. In this study, we evaluate two types of initial trajectory estimations. The first is a PPK GNSS–INS solution computed using the Inertial Explorer software (NovAtel Inc., 2023), which performs Kalman smoothing of RTK-corrected GNSS signals combined with IMU measurements. The second method is based on the approach proposed in Vezeteu et al. (2025), which fuses 3D LiDAR and IMU data for MLS and registers the trajectory to a prior ALS map. A brief overview is provided in Section 3.1.

A buffer of 2D LiDAR scans is then co-registered to a pre-existing sparse 3D reference map. We tested reference maps generated using

Table 1
Notation

Notation	Definition
$\mathcal{W}, \mathcal{I}, \mathcal{L}, \mathcal{G}$	The World, IMU, LiDAR, and GNSS frames
$\mathcal{ENU}$	East-North-Up global frame
$\mathcal{V}$	2D LiDAR VUX frame
$\mathbf{x}, \hat{\mathbf{x}}$	The ground-truth and predicted state
$\delta\mathbf{x}$	The error between ground-truth and the estimation
$\mathbf{x}_i$	Vector $\mathbf{x}$ at time i
$\mathbf{a}_m$	Measured vector $\mathbf{a}$
${}^{\mathcal{W}}\mathbf{p}_{\mathcal{I}}$	Position of frame $\mathcal{I}$ w.r.t. frame $\mathcal{W}$ with $\mathbf{p} \in \mathbb{R}^3$
${}^{\mathcal{W}}\mathbf{R}_{\mathcal{I}}$	Rotation of frame $\mathcal{I}$ w.r.t. frame $\mathcal{W}$ with $\mathbf{R} \in SO(3)$
${}^{\mathcal{W}}\mathbf{v}_{\mathcal{I}}$	Velocity of frame $\mathcal{I}$ w.r.t. frame $\mathcal{W}$ with $\mathbf{v} \in \mathbb{R}^3$
${}^{\mathcal{I}}\mathbf{T}_{\mathcal{L}}$	Transformation of frame $\mathcal{L}$ w.r.t. frame $\mathcal{I}$ with $\mathbf{T} \in SE(3)$

a different set of sensors, each with distinct fields of view and measurement accuracies, as detailed in Section 3.2. Additionally, precise georeferencing requires an accurate extrinsic calibration between the 2D sensor and the 6 DoF system. Our proposed calibration method is detailed in Section 3.3. Table 1 summarises the notations used throughout this study.

3.1. Initial registration guess

Our approach builds on the ALS-aided LiDAR-inertial navigation solution proposed in Vezeteu et al. (2025), which is based on the Robust Iterated Error-State Extended Kalman Filter (RIEKF) formulated on a Riemannian manifold (He et al., 2021). This system, referred to as 3D MLS, is employed in our study to provide an initial trajectory for registering 2D LiDAR scans.

However, the original system in Vezeteu et al. (2025) does not support 3D mapping using a 2D LiDAR scanner. To address this limitation, we extend the method by introducing a batch-based registration approach that operates directly on 2D scans. While our method builds on the core principles of the original system, it differs significantly in its adaptation to 2D LiDAR data and its use of batch updates instead of sequential filtering.

The discrete system state is defined in Eq. (1):

$$\begin{aligned}
&\mathcal{M} = \mathbb{R}^3 \times SO(3) \times SO(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3, \\
&\mathbf{x} := \begin{bmatrix} {}^{\mathcal{W}}\mathbf{p}_{\mathcal{I}} & {}^{\mathcal{W}}\mathbf{R}_{\mathcal{I}} & {}^{\mathcal{I}}\mathbf{R}_{\mathcal{L}} & {}^{\mathcal{I}}\mathbf{p}_{\mathcal{L}} & {}^{\mathcal{W}}\mathbf{v}_{\mathcal{I}} & \mathbf{b}_{\omega} & \mathbf{b}_{a} & {}^{\mathcal{W}}\mathbf{g} \end{bmatrix}^{\top}, \\
&\mathbf{u} := \begin{bmatrix} \boldsymbol{\omega}_m & \mathbf{a}_m \end{bmatrix}^{\top}, \\
&\mathbf{w} := \begin{bmatrix} \mathbf{n}_{\omega} & \mathbf{n}_{a} & \mathbf{n}_{b_{\omega}} & \mathbf{n}_{b_{a}} \end{bmatrix}^{\top},
\end{aligned} \tag{1}$$

where IMU position, orientation, and velocity in the world frame $\mathcal{W}$ (with $\mathcal{W} = I_0$) are denoted by ${}^{\mathcal{W}}\mathbf{p}_{\mathcal{I}}$, ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{I}}$, and ${}^{\mathcal{W}}\mathbf{v}_{\mathcal{I}}$, respectively. The inputs $\mathbf{a}_m$ and $\boldsymbol{\omega}_m$ denote the measured linear acceleration and angular velocity, respectively, while $\mathbf{n}_a$ and $\mathbf{n}_{\omega}$ represent the IMU measurement noise. The terms $\mathbf{n}_{b_a}$ and $\mathbf{n}_{b_{\omega}}$ are zero-mean Gaussian driving noises for the accelerometer and gyroscope biases $\mathbf{b}_a$ and $\mathbf{b}_{\omega}$. The gravitational acceleration is denoted as ${}^{\mathcal{W}}\mathbf{g}$. The dimensions of the state, input, and noise vectors are $\dim(\mathbf{x}) = 24$, $\dim(\mathbf{u}) = 6$, and $\dim(\mathbf{w}) = 12$.

Next, the system state is propagated through the process model from Eq. (2):

$$\begin{aligned}
&\mathbf{x}_{i+1} = \mathbf{x}_i \boxplus \left(\Delta t\ \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i)\right), \\
&\mathbf{x}_i \in \mathcal{M}, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathcal{Q}_i),
\end{aligned} \tag{2}$$

where $\mathbf{x}_i$ denotes the system state at time step i, Δt is the time increment, and $\mathcal{Q}_i$ is the process noise covariance. Vezeteu et al. (2025) provide a definition for $\boxplus$. The function $\mathbf{f}(\cdot)$ describes the system dynamics in Eq. (3) as: $\mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i) =$

$$\begin{bmatrix} {}^{\mathcal{W}}\mathbf{v}_{\mathcal{I}_i} + \frac{\Delta t}{2}\left({}^{\mathcal{W}}\mathbf{R}_{\mathcal{I}_i}(\mathbf{a}_{m_i} - \mathbf{b}_{a_i} - \mathbf{n}_{a_i}) - {}^{\mathcal{W}}\mathbf{g}_i\right) \\ \boldsymbol{\omega}_{m_i} - \mathbf{b}_{\omega_i} - \mathbf{n}_{\omega_i} \\ \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} \\ {}^{\mathcal{W}}\mathbf{R}_{\mathcal{I}_i}(\mathbf{a}_{m_i} - \mathbf{b}_{a_i} - \mathbf{n}_{a_i}) - {}^{\mathcal{W}}\mathbf{g}_i \\ \mathbf{n}_{\mathbf{b}_{\omega_i}} \\ \mathbf{n}_{\mathbf{b}_{\mathbf{a}_i}} \\ \mathbf{0}_{3\times 1} \end{bmatrix}. \tag{3}$$

After propagation, LiDAR measurements are integrated into the state update by minimising a point-to-plane cost function. Each LiDAR point is first transformed into the IMU frame $\mathcal{I}$ using the known extrinsic transformation ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{L}} = [{}^{\mathcal{I}}\mathbf{R}_{\mathcal{L}} | {}^{\mathcal{I}}\mathbf{p}_{\mathcal{L}}] \in SE(3)$, and then into the world frame using the latest IMU pose ${}^{\mathcal{W}}\mathbf{T}_{\mathcal{I}} \in SE(3)$.

The point-to-plane residual is computed by projecting the transformed point onto the estimated local surface normal. The normal vector ${}^{\mathcal{W}}\boldsymbol{\nu}$ is obtained from the PCA decomposition of the nearest neighbours of the corresponding map point. A validity check ensures that all neighbouring points lie within a predefined distance threshold from the fitted plane. The point-to-plane measurement is given in Eq. (4):

$$\begin{aligned}
&\hat{\mathbf{z}}_i^k = \mathbf{h}(\mathbf{x}_i, \pi_k) \in \mathbb{R} \\
&\hat{\mathbf{z}}_i^k = {}^{\mathcal{W}}\boldsymbol{\nu}\left({}^{\mathcal{W}}\mathbf{T}_{\mathcal{I}}\ {}^{\mathcal{I}}\mathbf{T}_{\mathcal{L}}\left({}^{\mathcal{L}}\mathbf{s}_i + {}^{\mathcal{L}}\mathbf{n}\right) - {}^{\mathcal{W}}\mathbf{q}\right)
\end{aligned} \tag{4}$$

where $\mathbf{h}(\cdot)$ is the measurement function, π_k is a plane with normal $\boldsymbol{\nu} \in \mathbb{R}^3$ and point on the plane $\mathbf{q} \in \mathbb{R}^3$, $\hat{\mathbf{z}}_i^k$ is the measurement of a plane π_k from state $\mathbf{x}_i$, ${}^{\mathcal{L}}\mathbf{s}_i$ is a source 3D point in LiDAR frame, and ${}^{\mathcal{L}}\mathbf{n}$ is its noise. ${}^{\mathcal{W}}\mathbf{T}_{\mathcal{I}}$ is the transformation from local to global or the initial trajectory guess, and ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{L}}$ is the extrinsic transformation matrix from LiDAR to the IMU frame.

The system maintains a local reference map, organised as a KD tree built from voxel-filtered recent scans within a fixed radius around the latest pose. New LiDAR scans are first motion-compensated using the IMU prediction, aligned to the local map, and then further refined through co-registration with the global ALS reference map. The method also estimates the transformation from the global ALS frame given in the East-North-Up coordinates ($\mathcal{ENU}$), where the ALS map is georeferenced and the PPK GNSS–INS trajectory is defined, to the local MLS frame ($\mathcal{W}$), represented as ${}^{\mathcal{W}}\mathbf{T}_{\mathcal{ENU}} \in SE(3)$. In essence, it aligns the MLS and GNSS trajectories to obtain a rough initial registration guess, which is subsequently refined by co-registering the overlapping regions of the ALS and MLS point clouds. More details about the implementation of the method can be seen in Vezeteu et al. (2025).

3.2. Batch update

Registering a single 2D scan at a time into a 3D map is unstable due to the limited FOV and the lack of overlap between successive scans. To address this, we accumulate a buffer of scans along with their initial pose estimates. Using a sliding window approach, we co-register overlapping buffers, allowing adjacent scans to support each other's alignment. In addition to the scan data, we utilise a prior sparse map as a reference to further constrain the solution. The core idea is that multiple consecutive scans may observe the same planar surface, which allows us to enforce geometric constraints and triangulate the poses. This triangulation arises when multiple poses observe the same plane, creating both pose-to-pose and pose-to-plane measurements. We combine these measurements as a list of graph constraints that help triangulate the trajectory and reduce drift.

Each buffer of scans is co-registered to the reference sparse map using batch optimisation, which we formulate as a maximum a posteriori (MAP) estimation problem. Let $\mathbf{x}_i$ denote the trajectory states, π_k the

plane landmarks, and $\mathbf{z}_i^k$ the set of observations of plane k from state $\mathbf{x}_i$. The MAP problem is defined in Eq. (5):

$$\arg\max_{\{\mathbf{x}_i\}} P\left(\{\mathbf{x}_i\} \mid \{\mathbf{u}_i\}, \{\pi_k\}, \{\mathbf{z}_i^k\}\right), \tag{5}$$

where P denotes the probability function. Assuming independent Gaussian noise models for motion and observation, and applying Bayes' theorem, the MAP estimate becomes equivalent to a maximum likelihood estimation (MLE) in Eq. (6):

$$\arg\max_{\{\mathbf{x}_i\}} \prod_i P\left(\mathbf{x}_i \mid \mathbf{x}_{i-1}, \mathbf{u}_{i-1}\right) \prod_{i,k} P\left(\mathbf{z}_i^k \mid \mathbf{x}_i, \pi_k\right) \tag{6}$$

which can be formulated to minimise a negative log-likelihood (NLL), where the objective function in Eq. (7) balances motion and observation consistency:

$$\arg\min_{\{\mathbf{x}_i\}} \left(\sum_i \left\|\mathbf{x}_i - \mathbf{f}(\mathbf{x}_{i-1}, \mathbf{u}_{i-1})\right\|^2_{\Sigma_i} + \sum_{i,k} \left\|\mathbf{z}_i^k - \underbrace{\mathbf{h}(\mathbf{x}_i, \pi_k)}_{\hat{\mathbf{z}}_i^k}\right\|^2_{\Omega_k} \right). \tag{7}$$

The first term penalises deviations from the initial trajectory. The second term penalises errors between observed and measured planes. Note that the observed $\mathbf{z}_i^k$ is always zero, as the expected value of the point-to-plane cost function is zero when the point lies exactly on the plane. Σ_i is the motion covariance matrix from the initial trajectory, and Ω_k is the observation covariance matrix (from the plane quality). The operators $\|\cdot\|^2_{\Omega}$ and $\|\cdot\|^2_{\Sigma}$ denote the squared Euclidean distance weighted by the Ω and Σ covariances, respectively. To establish plane measurements, we build a KD-tree from the sparse reference map and search for neighbours within a 1 m radius of each scan point. If the local neighbourhood can be approximated as a plane (using PCA), and all points lie within a distance threshold from this plane, the measurement is considered valid, and a plane constraint is added to the graph. The resulting factor graph includes both pose-to-pose and pose-to-plane constraints. An illustration is provided in Fig. 1, where each constraint can be interpreted as a spring. The stiffness of these springs is determined by the covariance matrices: Σ reflects confidence in the initial trajectory (black springs), Ω reflects the quality of the plane estimate (red springs), computed using the smallest eigenvalue λ_0 (the variance along the smallest eigenvector) from PCA. The observation uncertainty of plane π_k is defined in Eq. (8) using the 3-sigma rule:

$$\Omega_k = (3 \cdot \sqrt{\lambda_{0_k}})^2 = 9 \cdot \lambda_{0_k}. \tag{8}$$

To ensure temporal continuity, the last pose from the previous buffer is reused as an anchor in the current window.

Algorithm 1 summarises the optimisation process, which performs one Gauss–Newton (Nocedal and Wright, 2006) iteration given the factor graph. The algorithm updates the system matrices $\mathbf{H}$ and $\mathbf{b}$ using the Jacobians of the motion and plane measurements. It shows how measurement covariances influence the weighting of errors and Jacobians during optimisation. Lines 17–24 compute the motion error between consecutive poses and update the Hessian and gradient. Lines 29–35 handle the point-to-plane measurements, introducing geometric constraints based on plane landmarks. Appendix A provides details of the Jacobian computation and the optimisation update step. To mitigate the impact of outliers while maintaining efficiency for inliers, we apply a double weighting for point-to-plane measurements. The first weighting is given by the plane quality Ω, and further weighting is obtained using a Cauchy robust kernel (Chebrolu et al., 2021) denoted by α, whose formulation is given in Appendix B.

Algorithm 1 Batch update

1: **Input:**
2: $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ ▷ Initial states
3: $\mathcal{C} = \{c_1, \ldots, c_M\}$ ▷ Constraints
4: $\kappa(\cdot)$ ▷ Robust kernel function
5: ϵ ▷ Stopping threshold
6: **Output:**
7: $\mathcal{P}^*$ ▷ Optimised states
8:
9: Initialise $\mathbf{x} \leftarrow$ vectorise($\mathcal{P}$)
10: **while** ¬ converged **do**
11: $\mathbf{H} \leftarrow \mathbf{0}_{6N\times 6N}$ ▷ Hessian matrix
12: $\mathbf{b} \leftarrow \mathbf{0}_{6N\times 1}$ ▷ Right-hand side
13: **for all** constraints $c \in \mathcal{C}$ **do**
14: **if** c.type = pose-pose **then**
15: $\Sigma \leftarrow c$.cov; ▷ Covariance matrix
16: $i, j \leftarrow c$.from, c.to;
17: $\hat{\mathbf{z}}_i^j \leftarrow c.z_i^j$; ▷ Odometry measurement
18: $\mathbf{e}_{ij} \leftarrow \hat{\mathbf{z}}_i^j \boxminus (\mathbf{x}_i^{-1} \boxplus \mathbf{x}_j)$;
19: ▷ Compute Jacobians
20: $\mathbf{J}_i \leftarrow \frac{\partial \mathbf{e}_{ij}(\mathbf{x})}{\partial \mathbf{x}_i}$; $\mathbf{J}_j \leftarrow \frac{\partial \mathbf{e}_{ij}(\mathbf{x})}{\partial \mathbf{x}_j}$;
21:
22: $\mathbf{H}_{[ii]} \mathrel{+}= \mathbf{J}_i^\top \Sigma^{-1} \mathbf{J}_i$; $\mathbf{H}_{[ij]} \mathrel{+}= \mathbf{J}_i^\top \Sigma^{-1} \mathbf{J}_j$;
23: $\mathbf{H}_{[ji]} \mathrel{+}= \mathbf{J}_j^\top \Sigma^{-1} \mathbf{J}_i$; $\mathbf{H}_{[jj]} \mathrel{+}= \mathbf{J}_j^\top \Sigma^{-1} \mathbf{J}_j$;
24: $\mathbf{b}_{[i]} \mathrel{+}= \mathbf{J}_i^\top \Sigma^{-1} \mathbf{e}_{ij}$; $\mathbf{b}_{[j]} \mathrel{+}= \mathbf{J}_j^\top \Sigma^{-1} \mathbf{e}_{ij}$;
25:
26: **else if** c.type = pose-landmark **then**
27: $\Omega \leftarrow c$.cov; ▷ Covariance matrix
28: $i, k \leftarrow c$.from, c.to;
29: $\hat{\mathbf{z}}_i^k \leftarrow c.z_i^k$; ▷ Point-to-plane measurement
30: $\mathbf{e}_{ik} \leftarrow 0 - \hat{\mathbf{z}}_i^k$;
31: $\alpha \leftarrow \kappa(\mathbf{e}_{ik})$; ▷ Robust Kernel Weight
32: ▷ Compute Jacobians
33: $\mathbf{J}_k \leftarrow \frac{\partial \mathbf{e}_{ik}(\mathbf{x})}{\partial \mathbf{x}_i}$;
34:
35: $\mathbf{H}_{[ii]} \mathrel{+}= \alpha \mathbf{J}_k^\top \Omega^{-1} \mathbf{J}_k$; $\mathbf{b}_{[i]} \mathrel{+}= \alpha \mathbf{J}_k^\top \Omega^{-1} \mathbf{e}_{ik}$;
36:
37: **end if**
38: **end for**
39: Solve $\mathbf{H}\delta\mathbf{x} = -\mathbf{b}$; ▷ Gauss-Newton step
40: $\mathbf{x} \leftarrow \mathbf{x} + \delta\mathbf{x}$;
41: **if** $\|\delta\mathbf{x}\| < \epsilon$ **then**
42: **break**
43: **end if**
44: **end while**
45: $\mathcal{P}^* \leftarrow$ reconstruct($\mathbf{x}$);
46: **return** $\mathcal{P}^*$

3.3. Extrinsic calibration

Given a sparse reference map and a known 6 DoF trajectory defined in the same frame, the core idea is to reconstruct the reference map using the 2D LiDAR measurements and the known trajectory. Specifically, we use a 3D MLS system to generate a local sparse map and its associated trajectory over a short segment of motion ($\approx$ 20 m), which was found sufficient in our experiments. Points from the 2D LiDAR frame $\mathcal{V}$ are then projected onto the 3D map using the estimated trajectory and a rough initial guess of the extrinsic transformation ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{V}} \in SE(3)$. To refine the extrinsic transformation, we formulate an optimisation problem that minimises a hybrid cost function combining point-to-plane and point-to-point residuals in Eq. (9):

$$\min_{{}^{\mathcal{I}}\mathbf{T}_{\mathcal{V}} \in SE(3)} \left(\frac{1}{\left|\mathcal{I}_{\text{plane}}\right|} \sum_{i \in \mathcal{I}_{\text{plane}}} \kappa\left(\left\| \boldsymbol{\nu}_i ({}^{\mathcal{W}}\mathbf{T}_{\mathcal{I}}\, {}^{\mathcal{I}}\mathbf{T}_{\mathcal{V}} \mathbf{s}_i - \mathbf{q}_i) \right\|^2 \right) + \frac{1}{\left|\mathcal{I}_{\text{point}}\right|} \sum_{j \in \mathcal{I}_{\text{point}}} \kappa\left(\left\| {}^{\mathcal{W}}\mathbf{T}_{\mathcal{I}}\, {}^{\mathcal{I}}\mathbf{T}_{\mathcal{V}} \mathbf{s}_j - \mathbf{q}_j \right\|^2 \right) \right). \tag{9}$$

Here, ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{V}}$ is the extrinsic transformation from the 2D LiDAR frame to the MLS trajectory frame $\mathcal{I}$, and ${}^{\mathcal{W}}\mathbf{T}_{\mathcal{I}}$ is the known trajectory of frame $\mathcal{I}$ in the world frame $\mathcal{W}$. The loss function uses a robust kernel (defined in Appendix B) $\kappa(\cdot)$ to down-weight outliers. Source and target

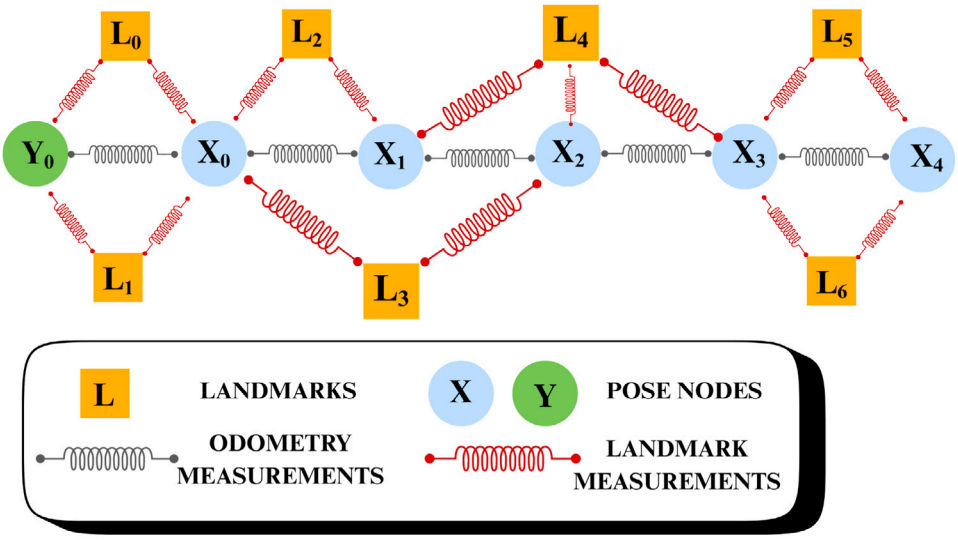


Fig. 1. Example of a graph constructed using initial odometry estimates and fixed plane landmarks. The blue and green circles represent pose nodes, where the green node $\mathbb{Y}_0$ corresponds to the final pose from the previous optimisation window and serves as an anchor to maintain continuity. Yellow squares denote fixed plane landmarks. Black springs indicate pose-to-pose constraints derived from odometry, while red springs represent point-to-plane measurements linking poses to landmarks. Each constraint is visualised as a spring: its stiffness reflects the confidence (inverse uncertainty) of the corresponding measurement.

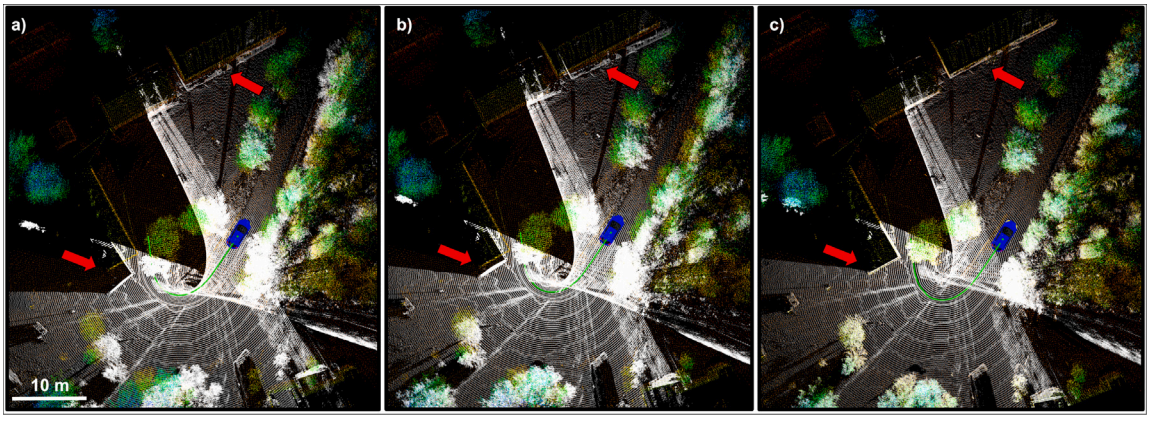


Fig. 2. A demonstration of the 2D LiDAR extrinsic calibration with the 3D MLS system. The coloured point cloud is the sparse 3D MLS point cloud downsampled with a voxel grid size of 0.5 m and coloured by the z-axis values. The grey point cloud is the overlapped 2D scans over the 3D MLS sparse map using the extrinsic transformation and the trajectory estimated by the 3D MLS system (green line). (a) shows the point cloud with the initial extrinsic transformation guess, (b) shows the point cloud after 10 iterations with our method, and (c) shows the georeferenced point cloud over the 3D MLS map with the final estimated extrinsic calibration parameters. The red arrows indicate the initial misalignment and its further correction. A small trajectory of ≈ 20 m was used for this example. The map scale is shown in the bottom-left corner.

point correspondences $(\mathbf{s}_i, \mathbf{q}_i)$ are found via a KD-tree nearest neighbour search. Points are included in the index set $\mathcal{I}_{\text{plane}}$ if a stable plane can be estimated in their local neighbourhood using PCA. Otherwise, they are placed in $\mathcal{I}_{\text{point}}$ and contribute to the point-to-point cost. The terms $\left|\mathcal{I}_{\text{plane}}\right|$ and $\left|\mathcal{I}_{\text{point}}\right|$ represent the number of point-to-plane and point-to-point errors, respectively. The optimisation problem is solved using the Gauss–Newton method. Implementation details are provided in Appendix C. Fig. 2 illustrates a distorted point cloud generated using a poor initial extrinsic guess, followed by the corrected, deskewed point cloud after extrinsic calibration.

Similarly, the extrinsic transformation between the 2D LiDAR and the GNSS–INS can be estimated, where we try to recreate the reference map using the 2D LiDAR data and the GNSS–INS motion defined in this map. The reference map is given in the MLS ($\mathcal{W}$) frame, while the GNSS–INS poses ${}^{\mathcal{ENU}}\mathbf{T}_{\mathcal{G}_i} \in SE(3)$ in the global East-North-Up frame ($\mathcal{ENU}$). Therefore, we first bring the GNSS–INS trajectory into a common reference $\mathcal{W}$ frame. Using the ALS-to-MLS transformation ${}^{\mathcal{W}}\mathbf{T}_{\mathcal{ENU}} \in SE(3)$ estimated by Vezeteu et al. (2025), we transform the GNSS–INS trajectory to the local MLS frame as in Eq. (10):

$$ {}^{\mathcal{W}}\mathbf{T}_{\mathcal{G}_i} = {}^{\mathcal{W}}\mathbf{T}_{\mathcal{ENU}}\,{}^{\mathcal{ENU}}\mathbf{T}_{\mathcal{G}_i} \quad \forall i \in [0, \dots, N], \tag{10} $$

where N is the number of poses in the trajectory. Thus, we minimise the objective function Eq. (11):

$$ \begin{aligned} \min_{{}^{\mathcal{G}}\mathbf{T}_{\mathcal{V}} \in SE(3)} \Bigg(& \frac{1}{\left|\mathcal{I}_{\text{plane}}\right|} \sum_{i \in \mathcal{I}_{\text{plane}}} \kappa \left(\left\| \boldsymbol{\nu}_i ({}^{\mathcal{W}}\mathbf{T}_{\mathcal{G}}\,{}^{\mathcal{G}}\mathbf{T}_{\mathcal{V}} \mathbf{s}_i - \mathbf{q}_i) \right\|^2 \right) \\ & + \frac{1}{\left|\mathcal{I}_{\text{point}}\right|} \sum_{j \in \mathcal{I}_{\text{point}}} \kappa \left(\left\| {}^{\mathcal{W}}\mathbf{T}_{\mathcal{G}}\,{}^{\mathcal{G}}\mathbf{T}_{\mathcal{V}} \mathbf{s}_j - \mathbf{q}_j \right\|^2 \right) \Bigg). \end{aligned} \tag{11} $$

Similar to Eq. (9), $\mathcal{I}_{\text{plane}}$ and $\mathcal{I}_{\text{point}}$ denote the sets of reference planes and points, respectively. The term ${}^{\mathcal{G}}\mathbf{T}_{\mathcal{V}} \in SE(3)$ represents the extrinsic transformation from the 2D LiDAR frame to the GNSS–INS frame $\mathcal{G}$, while ${}^{\mathcal{W}}\mathbf{T}_{\mathcal{G}}$ denotes the known trajectory of frame $\mathcal{G}$ in the world frame $\mathcal{W}$. This method estimates the extrinsic transformation between the 2D

LiDAR and GNSS–INS using a reference map aligned with the GNSS–INS trajectory. While we use an MLS-generated map here, any map (e.g., ALS) can be used, 3D LiDAR is not required, as long as the GNSS–INS trajectory is in the same frame as the reference map.

Note that the presented map-based extrinsic calibration between a 2D LiDAR and a 3D MLS (defined in the $\mathcal{I}$ frame) or GNSS–INS system requires a rough initial guess, which is then refined. When such a guess is unavailable, it can be estimated as follows. In our setup, we have two extrinsic pairs: 2D LiDAR with GNSS–INS, and 2D LiDAR with 3D MLS. A rough transformation between the GNSS–INS and the 3D MLS system, ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}$, can be estimated by aligning their 6 DoF trajectories using motion-based calibration. If the transformation between the 2D LiDAR and the 3D MLS is known, the 3D MLS can serve as an intermediate frame to recover an initial guess for the 2D LiDAR to GNSS–INS calibration.

To estimate the transformation between the 3D MLS and GNSS–INS, we formulate a motion-based calibration problem as a classical hand-eye calibration (Horn et al., 2021; Horaud and Dornaika, 1995; Välimäki et al., 2023), where relative motions from both trajectories are used to solve the static extrinsic transformation. Given a set of 3D MLS LiDAR poses ${}^{\mathcal{W}}\mathbf{T}_{\mathcal{I}_i} \in SE(3)$ and their corresponding PPK GNSS–INS poses ${}^{\mathcal{W}}\mathbf{T}_{\mathcal{G}_i} \in SE(3)$ we aim to estimate the fixed extrinsic transformation ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}} \in SE(3)$ that can be used to approximate PPK GNSS–INS poses with Eq. (12):

$$ {}^{\mathcal{W}}\mathbf{T}_{\mathcal{G}_i} \approx {}^{\mathcal{W}}\mathbf{T}_{\mathcal{I}_i}\,{}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}} \tag{12} $$

or, equivalent, we can approximate 3D MLS poses with Eq. (13):

$$ {}^{\mathcal{W}}\mathbf{T}_{\mathcal{I}_i} \approx {}^{\mathcal{W}}\mathbf{T}_{\mathcal{G}_i}\,{}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}^{-1}. \tag{13} $$

As sensor measurements are provided at discrete time steps, we first fit a cubic spline in $SE(3)$ to interpolate each trajectory, using the sensor measurements as control points. This yields a continuous-time trajectory that can be queried at arbitrary timestamps. Next, for a range of time intervals $\Delta i \in [0.1, 0.5, 1, 2, 3, 4, 5]$ s, we compute pairs of relative motion transforms $\mathbf{A}_i, \mathbf{B}_i \in SE(3)$, defined in Eq. (14) as:

$$ \begin{aligned} \mathbf{A}_i &= {}^{\mathcal{W}}\mathbf{T}_{\mathcal{I}_i}^{-1}\,{}^{\mathcal{W}}\mathbf{T}_{\mathcal{I}_{i+\Delta i}} \\ \mathbf{B}_i &= {}^{\mathcal{W}}\mathbf{T}_{\mathcal{G}_i}^{-1}\,{}^{\mathcal{W}}\mathbf{T}_{\mathcal{G}_{i+\Delta i}} \end{aligned} \tag{14} $$

We then formulate the following constraint as in Eq. (15):

$$ \mathbf{A}_i\,{}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}} = {}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}\mathbf{B}_i \Rightarrow \mathbf{A}_i\,{}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}\mathbf{B}_i^{-1}\,{}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}^{-1} = \mathbf{I} \in SE(3). \tag{15} $$

From this, we define the cost function (Eq. (16)) to be minimised:

$$ \min_{{}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}} \in SE(3)} \sum_i \left\| \log\left(\mathbf{A}_i\,{}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}\mathbf{B}_i^{-1}\,{}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}^{-1} \right) \right\|^2 \tag{16} $$

where $\log(\cdot)$ is the Lie logarithm map from $SE(3)$ to $\mathfrak{se}(3)$ (Sola et al., 2018). We compute the Jacobian of the cost function numerically and solve the optimisation using the Gauss–Newton. An illustration of the method is shown in Fig. 3.

Given the earlier estimated transformation from the 2D LiDAR to the 3D MLS ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{V}} \in SE(3)$, and the transformation from GNSS–INS to MLS ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}} \in SE(3)$. The transformation from the 2D LiDAR to the GNSS–INS system is given in Eq. (17):

$$ {}^{\mathcal{G}}\mathbf{T}_{\mathcal{V}} = {}^{\mathcal{G}}\mathbf{T}_{\mathcal{I}}\,{}^{\mathcal{I}}\mathbf{T}_{\mathcal{V}}. \tag{17} $$

where ${}^{\mathcal{G}}\mathbf{T}_{\mathcal{I}} = {}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}^{-1}$. Although the motion-based calibration performs comparably to the previously introduced map-based calibration, it has both an advantage and a disadvantage. Its main advantage is that it does not require an initial guess of the extrinsic transformation, unlike the map-based approach, which does. However, the motion-based method relies on sufficiently rich and non-degenerate sensor motion, i.e., motion that includes both non-parallel translation and rotation and is not limited to purely linear or rotational movement. This requirement can be challenging to satisfy with ground vehicles, which typically move in planar trajectories and cannot easily rotate and translate around all axes. Motion-based calibration is therefore primarily used to generate an initial estimate of the extrinsic transformation, which is then refined using the map-based approach.

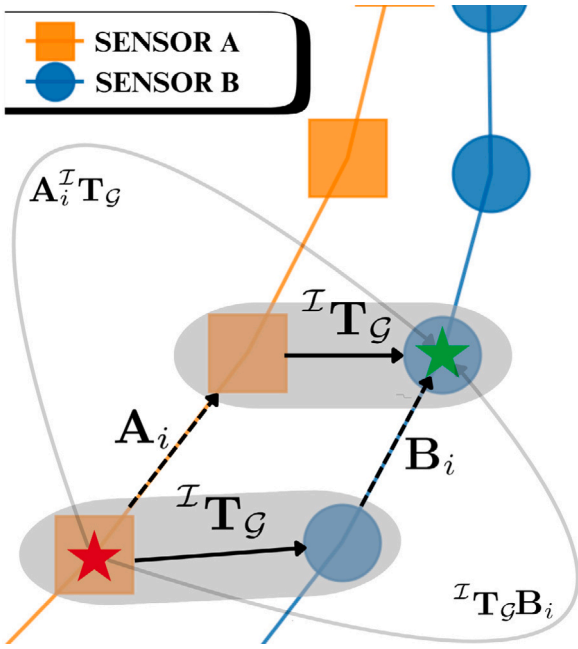


Fig. 3. Illustration of motion-based extrinsic calibration between two sensors. Given two sensor trajectories, the relative motion over any time interval is represented by transformations $\mathbf{A}_i$ and $\mathbf{B}_i$ in their respective frames. The goal is to find the fixed transformation ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}$ between the sensors such that $\mathbf{A}_i\,{}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}} = {}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}\mathbf{B}_i$. This relationship is visualised by showing two equivalent paths from the red star to the green star: one via $\mathbf{A}_i\,{}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}$, the other via ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{G}}\mathbf{B}_i$.

4. Experiments

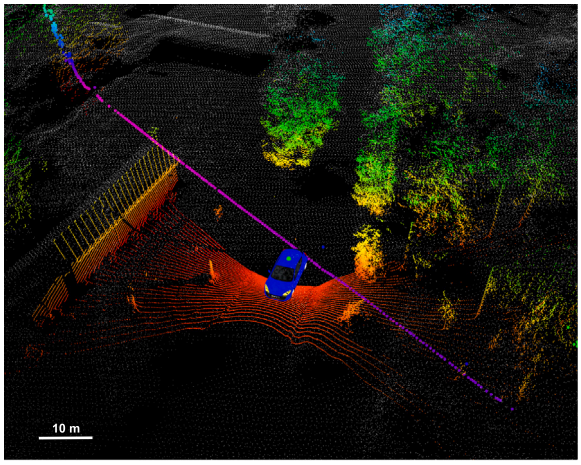
4.1. Platform and data collection

The sensor system illustrated in Fig. 4: (a) shows the physical vehicle, while (b) presents an example of the collected data. The platform is equipped with two sensor setups mounted at the front and rear of the car. The first setup, referred to as the Hesai system (Fig. 4(a) blue window), includes a Hesai Pandar XT-32 LiDAR (Hesai, 2024) integrated with an IMU CPT7 (NovAtel CPT7, 2022) and a NovAtel GNSS antenna (NovAtel, 2024). The Hesai LiDAR can generate up to 640,000 points per scan in single return mode at 10 Hz, with a measurement accuracy of ±1 cm. It has a 360° × 31° FOV, with 0.18° horizontal and 1° vertical angular resolution. The IMU provides inertial measurements at 100 Hz.

The Hesai system was oriented to maximise the point coverage in front of the vehicle. Each scan contained approximately 100,000 points. The 3D MLS coordinate system is defined with the y-axis pointing in the direction of vehicle motion, the x-axis pointing to the right, and the z-axis pointing upwards. Further details about the system are provided in Vezeteu et al. (2025).

The second sensor setup (Fig. 4(a) red window) is based on the Roamer mobile mapping system, previously described in El Issaoui et al. (2021), and includes a NovAtel GNSS–INS system (NovAtel, 2024) and two RIEGL VUX-1HA LiDAR scanners (RIEGL, 2023). Unlike the original configuration, which included a VUX-1HA and a miniVUX-1UAV, this version has two VUX-1HA scanners. In this study, we use only the left-hand scanner (relative to the vehicle's direction of motion). The VUX scanner operates at 250 Hz and delivers high-precision point measurements with survey-grade accuracy of 5 mm, a precision of 3 mm (1σ), and angle measurement resolution of 0.001°. It is mounted with its laser beams oriented nearly perpendicular to the ground, allowing a detailed assessment of road surface quality. The 2D VUX scanner frame has approximately the following orientation w.r.t. the

a)



b)

Fig. 4. (a) The 3D MLS system (blue frame) consists of a Hesai PandarXT-32 LiDAR sensor fused with a CPT7 IMU and a NovAtel GNSS antenna, all mounted on top of the vehicle. The 2D MLS system (red frame) includes a RIEGL VUX LiDAR scanner, fused with a uIMU-1C IMU and a NovAtel GNSS antenna. (b) Visualisation of the field of view (FOV): the grey points represent sparse ALS data from the National Land Survey (NLS), red-to-green points are from the Hesai scanner, and the purple points behind the car are from the VUX scanner. Hesai and VUX data are visualised using different colour mappings (z-axis colouring vs. intensity) to aid visual separation. A scale reference is shown in the bottom-left corner.

vehicle: x-axis points upward, the y-axis points to the left, and the z-axis points back. The synchronisation and triggering of the sensors are controlled via the GNSS clock.

In this study, we refer to the first system as the 3D MLS system (Hesai setup) and the second as the 2D MLS system (VUX setup). The 3D MLS system is considered fully calibrated: the Hesai LiDAR and NovAtel GNSS–INS share a synchronised time base, and the geometric transformations between the sensors have been accurately estimated. The extrinsic transformation between the 2D VUX scanner and the 3D Hesai system was estimated using the method presented in Section 3.3. An initial approximation of the extrinsic rotation was derived from the sensor mounting configuration shown in Fig. 4, providing a rough orientation between the two sensors. The translation was initialised to zero. Appendix D provides further details of the initial orientation estimate. After estimation, the resulting extrinsic transformation was manually inspected, and the quality of the generated point cloud was visually assessed.

Data collection took place on forest roads in the Evo area, in the municipality of Hämeenlinna, Finland, driving at approximately 40 km/h. The experiments presented here focus on a roughly 1 km segment of this route.

4.2. Experimental configuration

In this study, we evaluated the following configurations using initial registration obtained either from the PPK GNSS–INS solution or from the 3D MLS Hesai system, for georeferencing the 2D VUX scanner data:

- The PPK GNSS–INS trajectory is generated using the Inertial Explorer (NovAtel, 2024), which applies Kalman smoothing to fuse GNSS and inertial data.
- The trajectory estimated with the online 3D MLS Hesai system generated with the method of Vezeteu et al. (2025).

For each initial trajectory guess, we performed batch update corrections using three types of reference point cloud maps, illustrated in Fig. 5:

1. MLS map: Generated by the 3D MLS Hesai system and downsampled with a voxel leaf size of 30 cm. This map has a reported mean accuracy of 17 cm (Vezeteu et al., 2025) and an approximate point density of 2500 $\frac{\text{points}}{m^2}$.
2. Dense ALS data from FGI (2023): Georeferenced using a PPK GNSS–INS solution processed in RiPROCESS (RIEGL, 2023) and downsampled with a 30 cm voxel leaf size. This dataset has an accuracy error of up to 20 cm and an RMSE of roughly 5 cm, with an approximate point density of 5000 $\frac{\text{points}}{m^2}$.
3. Sparse ALS data from NLS: Referred to as sparse NLS, this dataset provides only about 5 $\frac{\text{points}}{m^2}$, significantly lower density than the other maps, and was used without downsampling. It exhibits a maximum vertical error of 10 cm and a planimetric error of up to 45 cm.

The 3D MLS Hesai system uses the FGI ALS data as the prior for registration. The buffer size of 50 2D LiDAR scans was used for batch update. It is important to note that no real-time (online) GNSS–INS solution was used. All GNSS–INS data were processed offline with NovAtel (2024).

The tested configurations were the following:

- **GNSS–INS:** 2D VUX point cloud georeferenced using the PPK GNSS–INS trajectory, with no batch update corrections.
- **Hesai:** 2D VUX point cloud georeferenced using the trajectory from the 3D MLS Hesai system, with no batch update corrections.
- **GNSS–INS + MLS:** Initial 2D scans georeferencing based on the PPK GNSS–INS trajectory, followed by batch update corrections using the downsampled Hesai-generated map as a reference.
- **Hesai + MLS:** Initial 2D scans georeferencing based on the Hesai trajectory, followed by batch update corrections using the downsampled Hesai-generated map as a reference.
- **GNSS–INS + D-ALS:** Initial 2D scans georeferencing based on the PPK GNSS–INS trajectory, followed by batch update corrections using the dense ALS map from FGI as a reference.
- **Hesai + D-ALS:** Initial 2D scans georeferencing based on the Hesai trajectory, followed by batch update corrections using the dense ALS map from FGI as a reference.
- **GNSS–INS + S-ALS:** Initial 2D scans georeferencing based on the PPK GNSS–INS trajectory, followed by batch update corrections using the sparse ALS map from NLS as a reference.
- **Hesai + S-ALS:** Initial 2D scans georeferencing based on the Hesai trajectory, followed by batch update corrections using the sparse ALS map from NLS as a reference.

The naming of each configuration indicates the following: ‘GNSS–INS’ or ‘Hesai’ specifies which system was used to obtain the initial

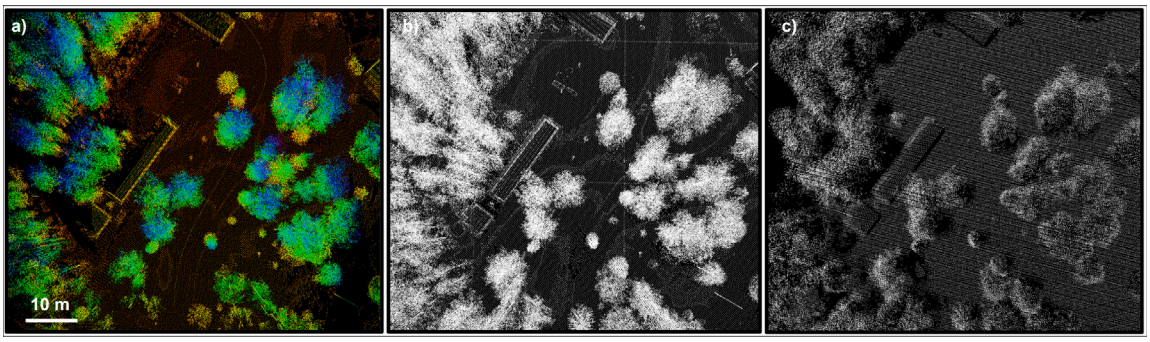


Fig. 5. An example of reference maps used for batch update corrections, visualised from a bird's-eye view. (a) Map generated using the 3D MLS Hesai system, showing the field of view from a moving vehicle, colour-coded by z-axis values. (b) ALS data from FGI, and (c) sparse ALS data from NLS. A voxel leaf size of 0.3 m was applied in (a) and (b), while voxelization was skipped for (c) due to the sparsity of the data. The scale bar is displayed in the bottom-left corner.

Table 2
Summary of tested configurations for 2D VUX georeferencing.

Configuration	Description
GNSS–INS	Only trajectory from PPK GNSS–INS
Hesai	Only trajectory from the 3D MLS Hesai system
GNSS–INS + MLS	PPK GNSS–INS trajectory and Hesai-generated MLS reference map.
Hesai + MLS	3D Hesai trajectory and Hesai-generated MLS reference map.
GNSS–INS + D-ALS	PPK GNSS–INS trajectory and dense ALS reference map.
Hesai + D-ALS	3D Hesai trajectory and dense ALS reference map.
GNSS–INS + S-ALS	PPK GNSS–INS trajectory and sparse ALS reference map.
Hesai + S-ALS	3D Hesai trajectory and sparse ALS reference map.

trajectory, followed by the type of reference map used. The tested configurations are summarised in Table 2, for easy reference throughout the paper.

To assess the accuracy of the 2D VUX point cloud registration, we use another ALS dataset that was recorded in 2024 at FGI, referred to as the evaluation map data. This dataset was georeferenced using a PPK GNSS–INS solution from a UAV platform and exhibits a registration accuracy with an RMSE of approximately 5 cm. It is important to note that the evaluation map data were not used in any registration process. It serves solely as an external reference for evaluation.

Two types of evaluation were performed using the evaluation map: a map-based surface deviation analysis and a tree-based landmark comparison, providing complementary insights into the geometric consistency of the 2D VUX scans registration.

4.3. Evaluation of point-to-surface deviation

We estimate the 2D VUX point cloud registration accuracy by evaluating the consistency of local planar surfaces with respect to the evaluation map data. The assumption is that the planar surfaces on the reference map should remain planar in the registered VUX data. For each point in the registered 2D VUX point cloud, nearest neighbours are identified from the reference map within a 1-metre radius using a KD-tree search. This ball-query approach is generally more robust in sparse environments than a k-nearest neighbours search. A local plane is then fitted to the neighbouring points using PCA of their covariance matrix. The plane's normal vector is defined by the eigenvector corresponding to the smallest eigenvalue, λ_0, and the curvature c is computed with Eq. (18):

$$c = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}. \tag{18}$$

A plane is considered valid if its curvature is less than 10^{-2}. PCA configurations with the eigenvalue ratio $\lambda_1/\lambda_2 < 10^{-3}$ are rejected, to filter out cases of collinear points, which are common in tree branches or overhead cables and may also yield small λ_0 without forming true planes. An illustration of the local plane extraction process is provided in Fig. 6. For points with a valid corresponding plane in the reference map, the point-to-plane distance is computed using Eq. (4) and the estimated extrinsic parameters. This metric quantifies local surface deviation: smaller errors indicate better alignment and higher registration quality. To assess whether reference plane quality influences the point-to-plane error, we compute the Pearson correlation coefficient r (Cohen et al., 2009) between point-to-plane distances and plane curvatures. Here, $r = \pm 1$ indicates perfect linear correlation, $r = 0$ indicates no correlation, $r \approx \pm 0.5$ indicates moderate correlation, and $r \approx \pm 0.1$ indicates a weak correlation. This analysis verifies that the estimated registration errors are not significantly biased by the quality of the reference planes.

4.4. Evaluation based on tree locations

A tree-based registration evaluation was conducted by comparing the tree locations extracted from the evaluation map to those detected in the 2D VUX-generated maps. Tree detection was performed using the method introduced by Hyyppä et al. (2020b), which was later extended to 3D MLS data by Hyyppä et al. (2020a) and to high-density ALS data by Hyyppä et al. (2022). Using the same ALS data as in this study, Muhojoki et al. (2024) demonstrated that this tree detection approach has a horizontal positioning error below 6 cm and a vertical positioning error below 10 cm for the detected trees.

After tree detection, trees in the 2D VUX and ALS maps were associated using a nearest-neighbour search, followed by manual verification. A total of 132 reference trees were used in the evaluation, as Fig. 7 shows.

Two types of errors were computed based on the matched tree locations: Absolute Pose Error (APE) and Relative Pose Error (RPE).

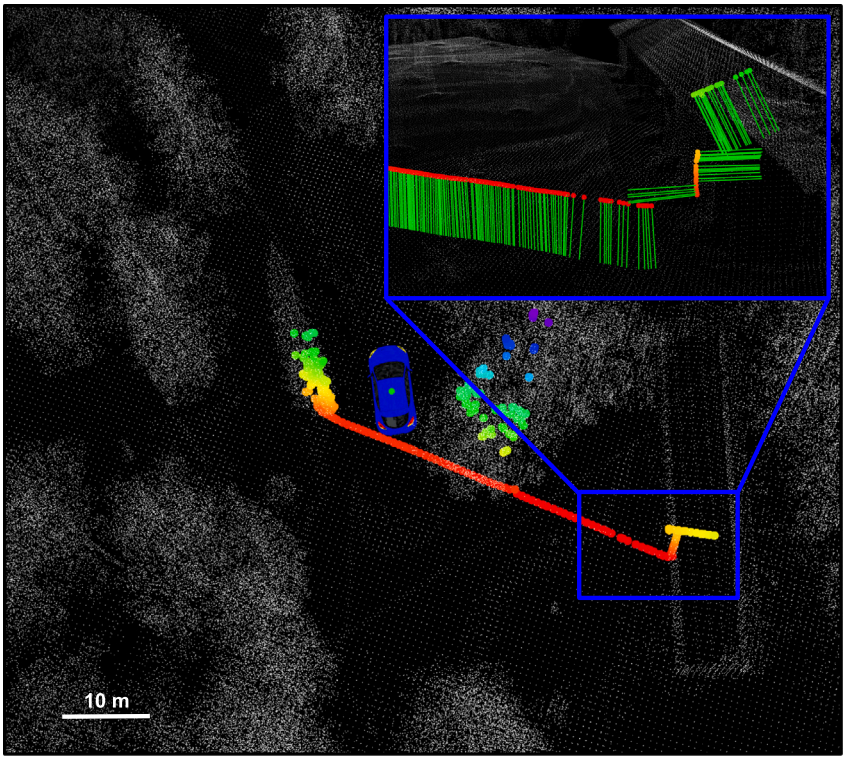


Fig. 6. An example of nearest plane estimation from the reference map. The grey point cloud represents the reference ALS data, while the orange-to-yellow points show a georeferenced VUX scan, colour-coded by z-axis values. In the zoomed-in window, green lines indicate the estimated nearest planes, visualised by their surface normals. A scale bar is provided in the bottom-left corner.

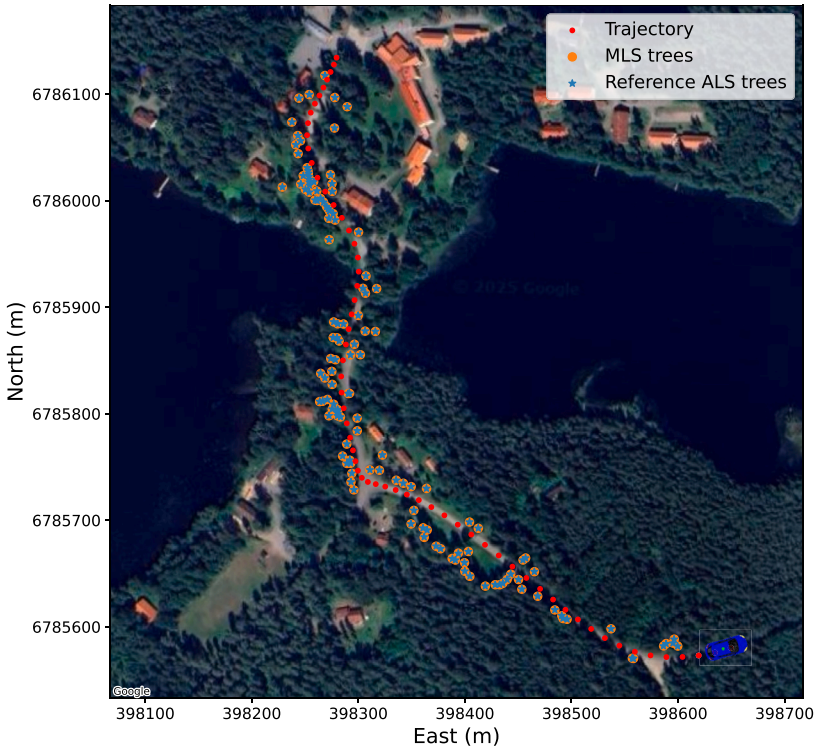


Fig. 7. Test area consisting of a forest road with a length of 1 km, shown over an orthophoto of the Evo region in southern Finland, where the study was conducted. The red dots represent the estimated vehicle positions, plotted at every 30th point. The vehicle is shown in blue at the end of the test road. The blue stars mark the reference ALS tree locations used for trajectory evaluation, while the orange circles denote the corresponding trees detected from the 2D MLS map.

APE measures the positional deviation of each estimated tree from its corresponding reference tree. Given the reference and estimated absolute position $P_{\text{ref},i}$ and $P_{\text{est},i} \in \mathbb{R}^2$ at index i, the absolute error is defined in Eq. (19):

$$\text{APE}_i = \|P_{\text{est},i} - P_{\text{ref},i}\|. \tag{19}$$

To evaluate the local consistency of the tree layout, we used RPE. We compute the pairwise distances between each tree i and all other trees j in both the reference and estimated datasets. The RPE (Eq. (20)) for a pair is calculated as:

$$\text{RPE}_i = \left| \|P_{\text{ref},i} - P_{\text{ref},j}\| - \|P_{\text{est},i} - P_{\text{est},j}\| \right|. \tag{20}$$

Note that tree-based metrics compute the errors based on the planar xy position of the trees, as the height of the tree is sometimes unreliable.

4.5. Ablation study

Given an accurate initial trajectory, we aim to evaluate the system's robustness to noise. To do this, we conduct an ablation study by injecting Gaussian noise into the trajectory and assessing the system's ability to correct for it. The original accurate trajectory serves as the reference, allowing us to determine how effectively the system compensates for the added noise. For each pose (defined as an element of $SE(3)$), we define a local perturbation ξ in the Lie algebra $\mathfrak{se}(3)$ (Sola et al., 2018), where Gaussian noise is applied only to the pitch angle (rotation around the y-axis) and the translation along the z-axis in Eq. (21):

$$\xi = \begin{bmatrix} \rho_x \\ \rho_y \\ \rho_z \\ \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \delta t_z \\ 0 \\ \delta\phi_y \\ 0 \end{bmatrix}, \quad \delta\phi_y \sim \mathcal{N}(0, \sigma_\phi^2),\ \delta t_z \sim \mathcal{N}(0, \sigma_t^2) \tag{21}$$

where ρ denotes the translation and ϕ denotes the rotation part. $\delta\phi_y$ and δt_z are sampled from zero-mean Gaussian distributions with standard deviations σ_ϕ^2 and σ_t^2, respectively. This perturbation is mapped to a transformation using the exponential map in Eq. (22):

$$\mathbf{T}_{\text{noisy}} = \mathbf{T}\exp(\xi) \tag{22}$$

with $\mathbf{T}_{\text{noisy}}, \mathbf{T} \in \text{SE}(3)$. Since we have 2D measurements with a small FOV and no overlap, the point-to-plane cost function used in this study enforces constraints only along the normal direction of planes. This introduces a requirement: both vertical and horizontal planes should be present.

In our case, the ground is a dominant planar surface, and it allows drift to be corrected in the translation along the z-axis and the pitch angle, which corresponds to the direction of vehicle motion. However, to constrain the remaining DoF, vertical planes are needed, which are largely absent in forested environments. In contrast, 3D sensors with wider FOV and overlapping frames can provide enough geometric structure to constrain all 6 DoF. But in our setting, the limited FOV and lack of overlap of 2D sensor data restrict our corrections to only 2 DoF.

As will be shown later, the 'Hesai' configuration yields approximately the most accurate trajectory based on our evaluation methods. We therefore use it as the ground truth for the ablation study. The following perturbation levels were tested: $\sigma_\phi^2 \in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]°$ and $\sigma_t^2 \in [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]$ cm.

In the batch update step, we use relative transformations between poses and inject realistic noise to simulate sensor inaccuracies. This controlled approach allows us to closely mimic real-world conditions, ensuring that our evaluation remains both rigorous and representative of practical scenarios.

Similarly to the tree-based location evaluation, described in Eqs. (19) and (20), we compute both absolute and relative pose errors between the 'Hesai' ground truth trajectory and the poses estimated from noisy inputs via batch update corrections. The reference prior map used in this evaluation is the FGI ALS map.

Finally, we assess the performance of each configuration using standard error metrics, including root mean squared error (RMSE), mean, median, and standard deviation for surface deviation, tree-based location, and ablation study errors. These metrics provide complementary insights, as no single metric fully captures all aspects of the registration accuracy. Because the error distributions may be non-Gaussian, we also compute 95% confidence intervals for the mean using the bootstrap method described in DiCiccio and Efron (1996), ensuring a statistically robust comparison of the evaluated configurations.

We conducted an ablation study on the extrinsic calibration between the 2D LiDAR and the 3D MLS system, denoted as ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{V}} = [{}^{\mathcal{I}}\mathbf{R}_{\mathcal{V}} | {}^{\mathcal{I}}\mathbf{p}_{\mathcal{V}}] \in SE(3)$. The extrinsic parameters were estimated using the method described in Section 3.3 and manually verified. The extrinsic translation vector is ${}^{\mathcal{I}}\mathbf{p}_{\mathcal{V}} = [-0.22, -3.01, -0.80]^\top$. To evaluate the system's robustness to errors in the initial extrinsic guess, we performed experiments in which the initial extrinsic translation was set to zero (${}^{\mathcal{I}}\mathbf{p}_{\mathcal{V}_{\text{init}}} = [0, 0, 0]^\top$), and varying levels of rotational error were added to the ground truth rotation.

Specifically, we introduced rotation errors by composing the ground truth rotation matrix with synthetic perturbation matrices generated from fixed-angle Euler rotations $\Theta \in [5, 10, 15, 20, 25, 30, 35, 40]°$. For each noise level $\theta \in \Theta$, we constructed a rotation matrix $\mathbf{R}_{\text{noise}} \in SO(3)$ by applying a fixed-angle Euler rotation of θ around each axis (roll, pitch, yaw). The composite noise rotation is built using the ZYX convention (yaw–pitch–roll) in Eq. (23):

$$\mathbf{R}_{\text{noise}} = \mathbf{R}_{\text{yaw}}(\theta)\mathbf{R}_{\text{pitch}}(\theta)\mathbf{R}_{\text{roll}}(\theta). \tag{23}$$

The perturbed rotation ${}^{\mathcal{I}}\mathbf{R}_{\mathcal{V}_{\text{perturbed}}}$ was then obtained with Eq. (24):

$${}^{\mathcal{I}}\mathbf{R}_{\mathcal{V}_{\text{perturbed}}} = {}^{\mathcal{I}}\mathbf{R}_{\mathcal{V}}\mathbf{R}_{\text{noise}} \in SO(3). \tag{24}$$

The system was initialised with ${}^{\mathcal{I}}\mathbf{T}_{\mathcal{V}_{\text{init}}} = [{}^{\mathcal{I}}\mathbf{R}_{\mathcal{V}_{\text{perturbed}}} | {}^{\mathcal{I}}\mathbf{p}_{\mathcal{V}_{\text{init}}}] \in SE(3)$, which was then refined during optimisation. This setup enabled a systematic evaluation of the system's performance under progressively larger errors in the initial extrinsic parameters. Specifically, we tested with rotation noise levels of up to 40° while keeping the translation vector set to zero.

5. Results and discussions

5.1. Results on georeferencing the 2D VUX scans

Fig. 8 presents a qualitative comparison between point clouds generated using the 2D VUX LiDAR and the 3D Hesai LiDAR. Subfigures (a) to (c) show both a top-down overview and zoomed-in views of the map. These views highlight differences in point density and structural clarity between the two sensors. Subfigure (d) specifically focuses on a 1×1 m road patch, where the point clouds from both sensors are overlaid, VUX in green and Hesai in blue. In this patch, the VUX scanner produced approximately 5200 points, whereas the Hesai scanner recorded around 2680 points. As expected, it demonstrates that the VUX generates a denser point cloud than the Hesai in the same environment and driving at the same speed (roughly 40 km/h). This increased density allows the VUX-based map to capture finer-grained details of the environment, making it more suitable for tasks requiring high spatial resolution, such as precise road surface modelling or tree structure estimation. The result is particularly important, as it highlights the motivation for using 2D LiDAR for 3D mapping. Table 3 presents the surface-based trajectory results for each configuration (previously summarised in Table 2). They reflect the deviation between planar surfaces in the reference map and those reconstructed by each model configuration. No significant difference is observed between the PPK GNSS–INS and Hesai-based initial trajectories, with both achieving a mean error of approximately 0.14 m. However, GNSS–INS configurations tend to produce a slightly

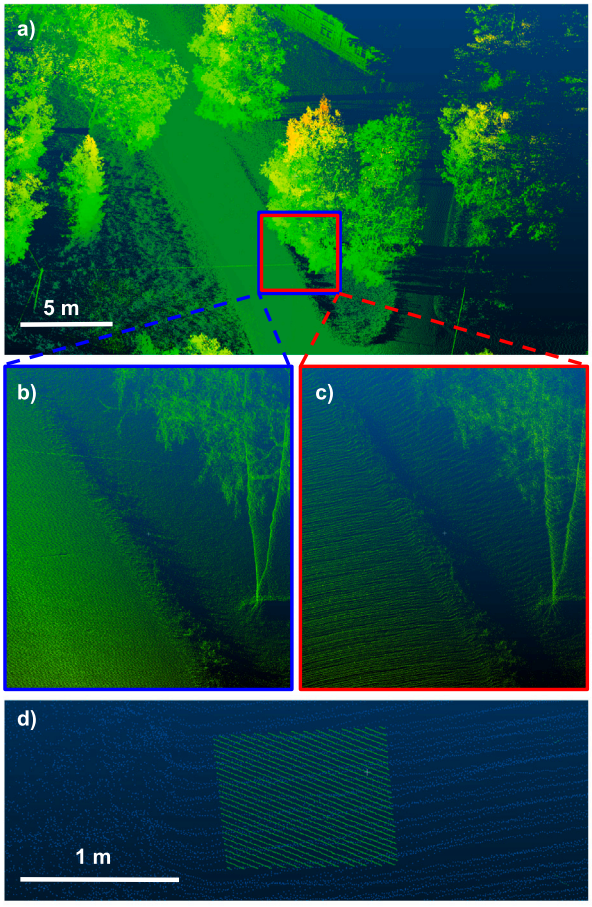


Fig. 8. Qualitative comparison of point cloud maps generated using the 2D VUX LiDAR and the 3D Hesai LiDAR. (a) Top-down view of the overall point cloud. (b) and (c) show a zoomed-in segment comparing point clouds from the VUX (blue) and Hesai (red) LiDAR. The point clouds are coloured by height (z-axis). (d) A 1 × 1 m patch of road from the VUX data (shown in green) overlaid on the Hesai map (shown in blue), highlighting differences in point cloud density.

Table 3
The statistics on point-to-surface evaluation errors. The point-to-surface mean values are accompanied by bootstrap 95% confidence intervals. The last column shows the Pearson correlation factor between the point-to-surface error and the reference plane's curvature.

Statistics					
	Point-to-surface [m]				Pearson correlation
	Mean	Median	RMSE	Std	
GNSS–INS	0.14 ± 0.0012	0.09	0.20	0.15	0.15
GNSS–INS + MLS	0.14 ± 0.0011	0.09	0.20	0.15	0.14
GNSS–INS + D-ALS	0.13 ± 0.0012	0.08	0.20	0.15	0.15
GNSS–INS + S-ALS	0.14 ± 0.001	0.09	0.20	0.14	0.13
Hesai	0.14 ± 0.0009	0.11	0.19	0.13	0.05
Hesai + MLS	0.14 ± 0.001	0.11	0.19	0.13	0.05
Hesai + D-ALS	0.14 ± 0.0008	0.11	0.19	0.13	0.05
Hesai + S-ALS	0.14 ± 0.001	0.11	0.19	0.13	0.04

lower median error (0.09 m) than Hesai configurations (0.11 m). Conversely, the Hesai-based configurations exhibit lower RMSE and standard deviation, suggesting more consistent overall performance.

To evaluate the full distribution of point-to-surface errors, we compare the average performance of the GNSS–INS and Hesai configurations using the cumulative distribution function (CDF), shown in Fig. 9. Here, 'average' refers to all errors obtained using either GNSS–INS or Hesai as the initial trajectory source. The GNSS–INS configuration performs better in the low-error region (up to 0.16 m), with a steeper CDF curve indicating higher precision under typical conditions. However, Hesai achieves a higher cumulative probability for larger errors, indicating greater robustness to outliers. This highlights a trade-off: GNSS–INS provides more accurate estimates in nominal conditions, while Hesai offers improved consistency and resilience to large deviations. It is noteworthy that the point-to-surface deviation was computed only for points with a stable nearest plane, which in our data were predominantly ground points. The ground surface, therefore, dominates the point-to-surface error. To ensure that the point-to-surface errors are meaningful and unbiased by the quality of the reference planes, we examine their correlation with reference surface curvature. Fig. 10 presents the 2D histogram of this relationship for the 'Hesai' and 'GNSS–INS' configurations. In theory, poorly estimated planes would result in higher point-to-plane residuals, which would manifest as a visible linear trend in the 2D histogram. However, such a trend is not observed. This indicates that the reported errors are not significantly influenced by the quality of the reference surfaces. The correlations for all configurations are reported in Table 3. For most Hesai-based models, the correlation is close to 0.05, suggesting no dependency on surface quality. GNSS–INS-based models also show only a weak correlation (r = 0.15), indicating minimal influence. These findings support the conclusion that the point-to-surface errors are largely independent of reference surface quality, reinforcing the reliability of the evaluation results.

The summary statistics on tree-based localisation results are presented in Table 4 and visualised in Fig. 11. These results reflect the deviations between tree positions extracted from the reference map and those estimated by each model. Overall, the Hesai-based models outperform the GNSS–INS-based ones, achieving a lower mean error (0.10 m vs. 0.16 m) and consistently better performance across all metrics, including median, RMSE, and standard deviation of both absolute and relative errors. This suggests that the Hesai-based approach provides higher localisation accuracy in the horizontal (xy-plane) domain.

In terms of initial trajectory quality, both the Hesai-based and PPK GNSS–INS methods produce comparable mean errors when evaluated against surface deviation. This is expected, as the initial trajectories are already accurate due to minimal canopy obstruction, which also ensures reliable GNSS performance. However, the tree-based evaluation reveals that the Hesai-based trajectory provides slightly better accuracy. This suggests that point-to-plane distances, primarily influenced by flat ground planes, are more sensitive to vertical (z-axis) errors, whereas tree-based metrics better capture horizontal (xy-axis) discrepancies. It is noteworthy that the reference map used for evaluation contains minor inaccuracies, so we cannot conclusively determine that the Hesai-based trajectory is definitively superior. Nonetheless, the PPK GNSS–INS approach is an offline method reliant on GNSS availability and signal quality, making it less effective in GNSS-denied environments. In contrast, the Hesai-based method operates online and GNSS-free, making it more robust and better suited to real-time applications.

5.2. Ablation study results

Fig. 12 presents the impact of trajectory noise. Artificial Gaussian noise was added to the trajectory of the 'Hesai' configuration, and the resulting noisy trajectory was used as the initial guess for the 'Hesai + D-ALS' model configuration (chosen arbitrarily). This model effectively corrects for both translational and rotational noise. For translation, the resulting error remains nearly flat up to 30 cm of induced trajectory noise, demonstrating strong robustness. In contrast, the model is slightly less robust to rotational noise, as indicated by a small upward trend in the model error curve. This is also observable, when comparing the slopes of the added noise (red line) with the

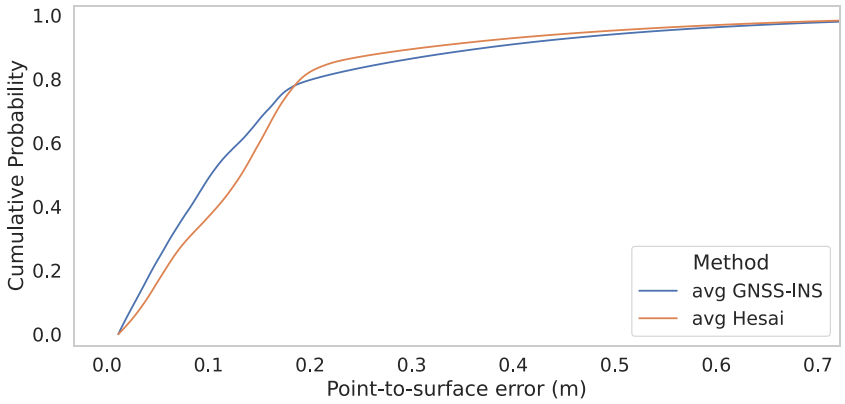
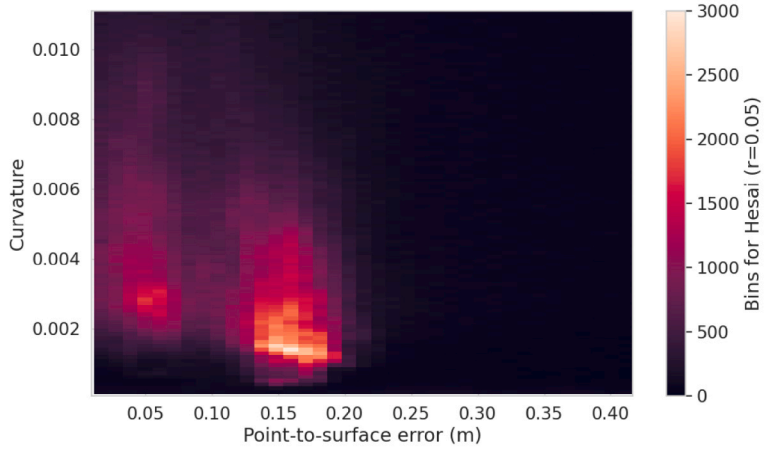
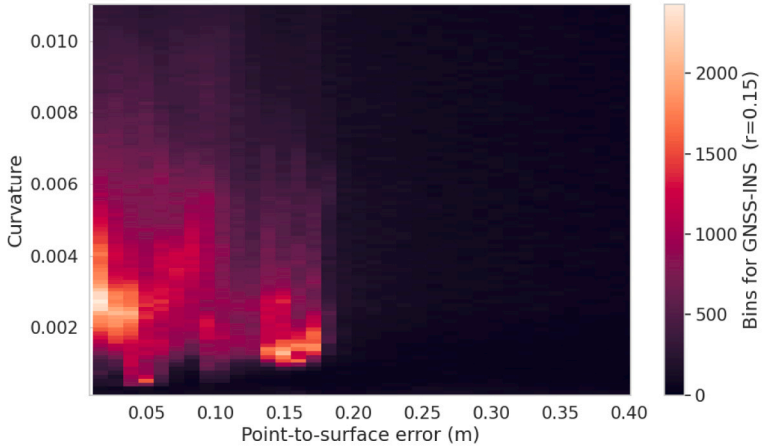


Fig. 9. Cumulative distribution function (CDF) of point-to-surface errors for the average of all GNSS–INS models and the average of all Hesai models. The GNSS–INS estimator shows higher precision for errors below ≈ 0.16 m, while the Hesai estimator surpasses it beyond this threshold, indicating greater robustness and fewer large errors.



a)



b)

Fig. 10. 2D histograms showing the distribution of point-to-surface errors as a function of reference plane quality, measured by the unitless surface curvature. A bin size of 100 was used for both plots. (a) Results for the 'Hesai' configuration. (b) Results for the 'GNSS–INS' configuration. The Pearson correlation coefficient r is shown in each plot.

Table 4
Absolute and relative error statistics based on the reference tree locations. The mean values are accompanied by bootstrap 95% confidence intervals.

Statistics								
	Absolute [m]				Relative [m]			
	Mean	Median	RMSE	Std	Mean	Median	RMSE	Std
GNSS–INS	0.16 ± 0.014	0.13	0.18	0.09	0.10 ± 0.001	0.08	0.14	0.09
GNSS–INS + MLS	0.17 ± 0.016	0.13	0.20	0.10	0.13 ± 0.001	0.10	0.17	0.11
GNSS–INS + D-ALS	0.16 ± 0.019	0.13	0.20	0.10	0.12 ± 0.001	0.09	0.16	0.10
GNSS–INS + S-ALS	0.16 ± 0.016	0.13	0.19	0.10	0.12 ± 0.001	0.09	0.16	0.10
Hesai	0.10 ± 0.011	0.10	0.12	0.06	0.08 ± 0.001	0.06	0.11	0.07
Hesai + MLS	0.10 ± 0.010	0.10	0.12	0.06	0.08 ± 0.001	0.06	0.11	0.07
Hesai + D-ALS	0.11 ± 0.016	0.10	0.14	0.09	0.08 ± 0.002	0.06	0.13	0.10
Hesai + S-ALS	0.11 ± 0.013	0.10	0.14	0.09	0.09 ± 0.002	0.07	0.14	0.10

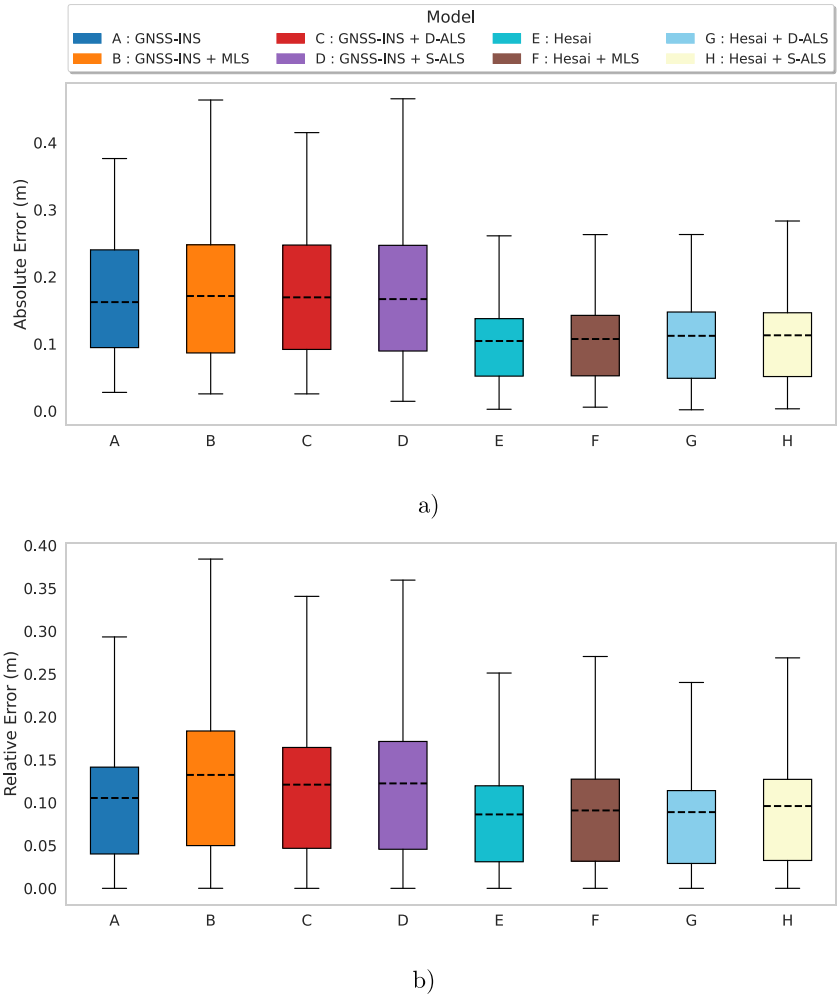


Fig. 11. The box plots of (a) absolute and (b) relative tree-based translation errors, with the mean indicated by a dashed black line.

residual error of the model (green line), where there is a substantial reduction: the translation error reduces by a factor of approximately $\frac{0.79}{0.08} = 9.88$, and the rotation error by a factor of $\frac{0.81}{0.12} = 6.75$. Fig. 13 presents a visual comparison of the improvement, where the refined trajectory results in a significantly sharper and more coherent point cloud. Note that in the ablation study, we added artificial noise in only 2 DoF (pitch and vertical translation) due to the limited availability of planar references, mostly ground planes. To robustly evaluate all 6 DoF, planar surfaces in multiple directions are needed. Although the current study focuses on 2D sensors (VUX), the proposed method applies even better to 3D LiDARs. The larger FOV and scan overlap of 3D LiDARs provide better constraints for full 6 DoF estimation.

When a reliable initial trajectory is available, incorporating a reference map provides little to no benefit and may even slightly reduce performance while increasing computational cost, due to imperfections in the map itself. However, reference maps are still useful in cases where the quality of the initial trajectory is uncertain, because they can assist in verifying or correcting the trajectory, as shown in our ablation study.

5.3. Extrinsic calibration results

Fig. 14 illustrates the system's robustness to errors in the initial extrinsic guess. The figure shows the mean squared error (MSE) of

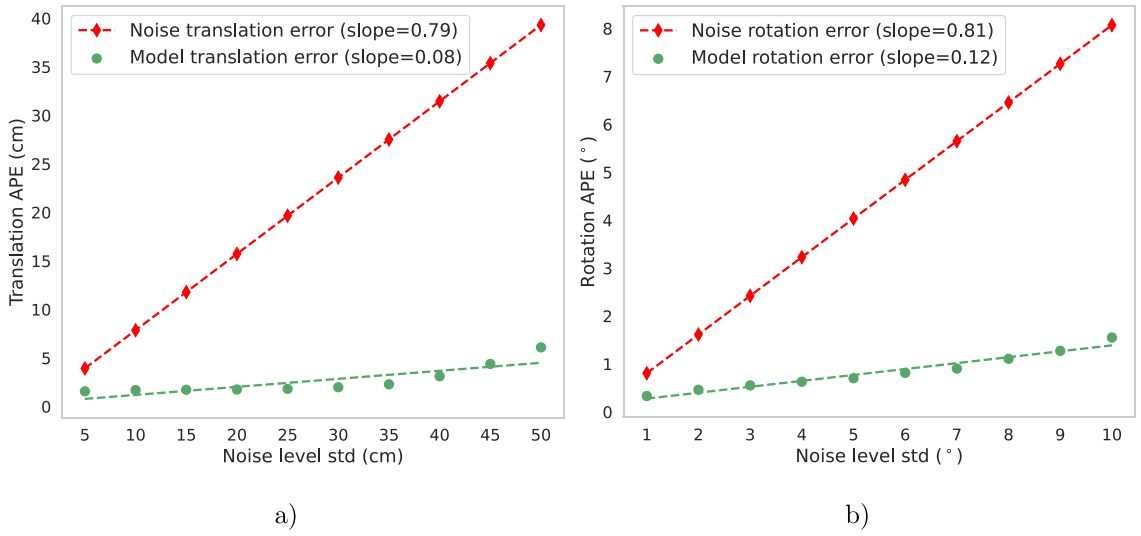


Fig. 12. Mean absolute pose errors (APE) from the ablation study. (a) Mean absolute translation error for varying levels of added noise. (b) Mean absolute rotation error for the same noise levels. In each case, translation and rotation noise are applied simultaneously. The red line represents the error of the noisy trajectory, while the green line shows the error after using the 'Hesai + D-ALS' configuration. The slope of each line indicates the model's robustness to noise, quantifying how much the output error increases per unit increase in input noise standard deviation. A flatter slope suggests higher robustness.

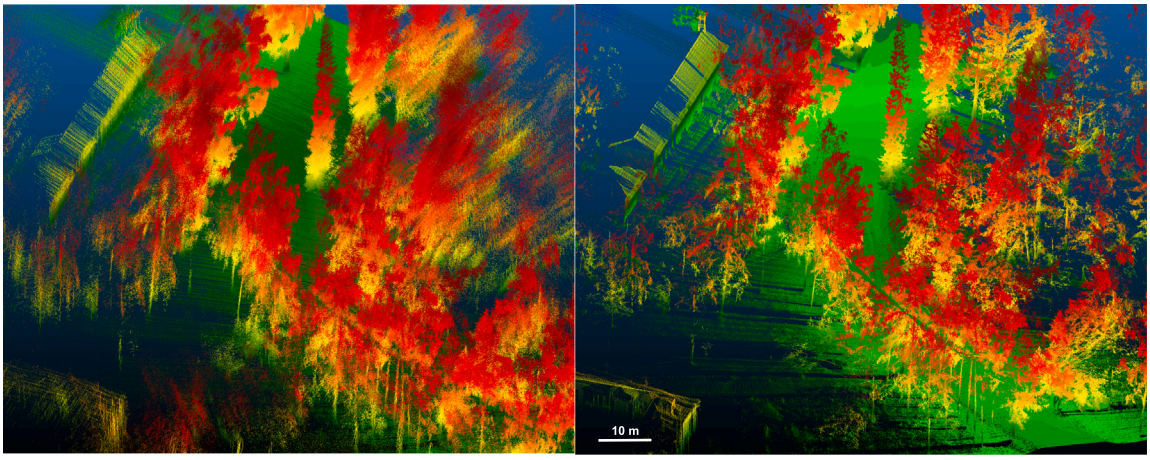


Fig. 13. (a) The ablation study point cloud was generated using a trajectory corrupted with 20 cm translation noise and 2° rotation noise. (b) Point cloud georeferenced from the refined trajectory after noise correction. The point clouds are coloured by z-axis value.

the extrinsic calibration objective (Eq. (9)) across varying levels of rotational noise. Each experiment starts with a zero translation initialisation and a perturbed rotation matrix. The results demonstrate that the system reliably converges to accurate estimates under low to moderate rotational disturbances, up to about 40°. It is noteworthy that the initial extrinsic translation was set to zero, which is far away from the true translation along the vehicle axis (> 3m). Despite this, the proposed calibration method, described in Section 3.3, successfully recovers the full 6-DoF transformation between the 2D LiDAR and the 3D MLS (or GNSS–INS) system, given a rough rotational prior. For improved calibration results, we recommend that the process be performed in a structured environment that benefits scan matching. We demonstrated that the method converges even when the initial extrinsic translation is entirely unknown. Additionally, Appendix D provides a step-by-step guide for manually estimating an initial extrinsic rotation. Although our experiments focus on a ground vehicle setup, the proposed method generalises well to other platforms, including vehicles and drones, and applies to both 2D and 3D LiDAR sensors.

Fig. 15 illustrates the qualitative extrinsic calibration results for the georeferenced 2D VUX data. The figure compares the 2D VUX point cloud georeferenced with the PPK GNSS–INS trajectory before and after applying extrinsic correction and trajectory refinement. The point cloud is overlaid on an MLS map that combines 3D Hesai and ALS data. In the initial state, clear misalignments are visible, particularly around trees, power lines, and building edges, due to inaccurate extrinsic parameters in the 'GNSS–INS + MLS' configuration. After applying our extrinsic calibration and performing map-based refinement, the alignment improves significantly. This demonstrates that even when using a high-quality trajectory, poor extrinsic calibration can lead to local bending and misalignments in the final point cloud.

6. Conclusion

This paper presented a novel solution for 3D mapping using a 2D LiDAR sensor. We achieved this by introducing a direct integration framework that fused 2D LiDAR measurements with comprehensive

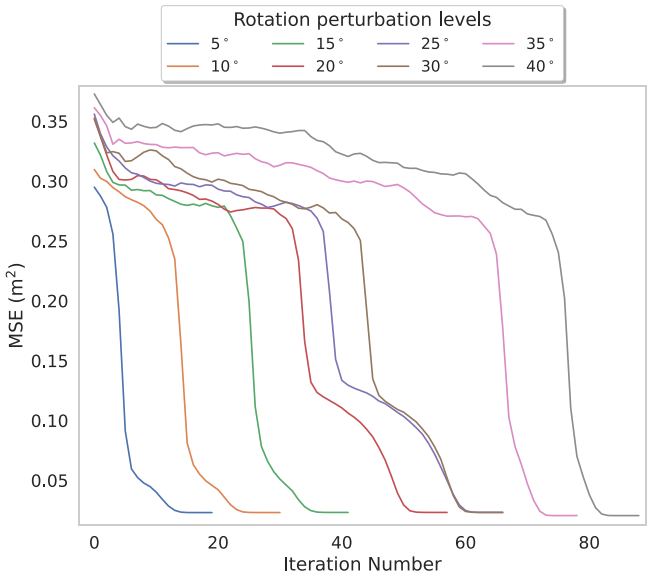


Fig. 14. Convergence behaviour of the extrinsic calibration ${}^{I}\mathbf{T}_{v}$ between the 2D LiDAR and the 3D MLS system. The figure shows the mean squared error (MSE) cost, computed using Eq. (9), for the final calibrated transformation. Each calibration run starts from a zero extrinsic translation guess and an extrinsic rotation, with varying perturbation levels applied to the rotation matrix.

3D information from both MLS and ALS systems, leveraging the high accuracy of PPK GNSS–INS.

Our approach performed a direct batch optimisation by jointly co-registering multiple 2D scans using a given initial trajectory and a sparse reference map. Furthermore, we introduced a direct extrinsic calibration method that estimates the transformation between the 3D LiDAR, GNSS–INS, and 2D LiDAR systems without requiring calibration targets.

Our method was successfully implemented and tested in a real-world forest scenario. Experimental results show that when the initial trajectory guess is accurate, further refinement using the reference map offers little to no additional benefit. However, the system's accuracy still depends on well-calibrated extrinsic parameters. Overall, the method achieved a localisation error of approximately 10 cm in the xy-plane and 14 cm along the z-axis, when evaluated against a reference map with an accuracy of about 5 cm RMSE. Additionally, when the initial trajectory is noisy, the proposed method effectively reduces this noise through scan alignment and batch optimisation.

The successful integration of 3D MLS and ALS systems with 2D LiDAR enables precise localisation and mapping, opening new opportunities for applications in mapping, forestry, and autonomous navigation. Future work could explore extending the proposed framework to directly handle 3D LiDAR scans and operate in more challenging or GNSS-denied environments.

CRediT authorship contribution statement

Eugeniu Vezeteu: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Aimad El Issaoui:** Writing – review & editing, Supervision, Investigation, Formal analysis, Data curation, Conceptualization. **Heikki Hyyti:** Writing – review & editing, Supervision, Formal analysis, Conceptualization. **Jesse Muhojoki:** Writing – review & editing, Validation, Data curation. **Petri Manninen:** Writing – review & editing, Visualization, Investigation, Formal analysis, Conceptualization. **Teemu Hakala:** Writing – review & editing, Resources, Data curation. **Eric Hyyppä:** Writing – review & editing, Supervision. **Antero Kukko:** Writing – review & editing, Supervision, Data curation. **Harri Kaartinen:** Writing – review & editing, Resources, Data curation. **Ville Kyrki:** Writing – review & editing, Supervision. **Juha Hyyppä:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for this journal and was not involved in the editorial review or the decision to publish this article.

Acknowledgements

We gratefully acknowledge the Research Council of Finland for funding this study through the grants 'Forest-Human–Machine Interplay - Building Resilience, Redefining Value Networks, and Enabling Meaningful Experiences' (359175), 'Towards Automatic Road Inspection' (365583), and 'Innovative Methods for measuring and modelling the dynamic quality of forest roads' (362928), "Collecting accurate individual tree information for harvester operation decision making" (359554), and the Business Finland through grant (MIXER 3502/31/2023).

Appendix A. Batch optimisation

In the following, we present the Jacobian estimation only for the $SE(3)$ part of Eq. (1), which has the format translation first, and then rotation in $\mathfrak{se}(3)$. Given:

- $\mathbf{T}_i \in \mathrm{SE}(3)$: pose of scan i, to be optimised
- $\hat{\mathbf{T}}_{ij} \in \mathrm{SE}(3)$: odometry measurement from pose i to pose j
- $\pi_k = (\boldsymbol{\nu}_k, \mathbf{q})$: fixed plane landmark with normal $\boldsymbol{\nu}_k \in \mathbb{R}^3$ and point on the plane $\mathbf{q} \in \mathbb{R}^3$
- $\mathbf{s}_i \in \mathbb{R}^3$: source point in LiDAR frame i observed to lie on plane π_k

The cost function in Eq. (7) can be rewritten as Eq. (A.1):

$$\min_{\{\mathbf{T}_i\}} \left(\sum_{(i,j)} \|\mathbf{e}_{ij}^{\text{odom}}\|^2_{\Sigma_{ij}} + \sum_{(i,k)} \|\mathbf{e}_{ik}^{\text{plane}}\|^2_{\Omega_k} \right) \tag{A.1}$$

where:

- $\mathbf{e}_{ij}^{\text{odom}} \in \mathbb{R}^6$: pose-to-pose error
- $\mathbf{e}_{ik}^{\text{plane}} \in \mathbb{R}$: point-to-plane error
- Σ_{ij}: $\mathbb{R}^{6\times6}$ odometry covariance of $SE(3)$ part
- Ω_k: scalar covariance for plane constraint

The odometry (pose-to-pose) and the pose-to-plane errors are given by Eq. (A.2):

$$\begin{cases} \mathbf{e}_{ij}^{\text{odom}} = \log\left(\hat{\mathbf{T}}_{ij}^{-1}\mathbf{T}_i^{-1}\mathbf{T}_j\right) \in \mathbb{R}^6 \\ \mathbf{e}_{ik}^{\text{plane}} = \boldsymbol{\nu}_k^{\top}(\mathbf{T}_i\mathbf{s}_i - \mathbf{q}_k) \in \mathbb{R}. \end{cases} \tag{A.2}$$

where $\log(\cdot)$ is the Lie logarithm map from $SE(3)$ to $\mathfrak{se}(3)$ (Sola et al., 2018).

The pose is updated with a small perturbation $\delta\boldsymbol{\xi}_i \in \mathbb{R}^6$ applied via Eq. (A.3):

$$\mathbf{T}_i \leftarrow \exp(\delta\boldsymbol{\xi}_i)\mathbf{T}_i. \tag{A.3}$$

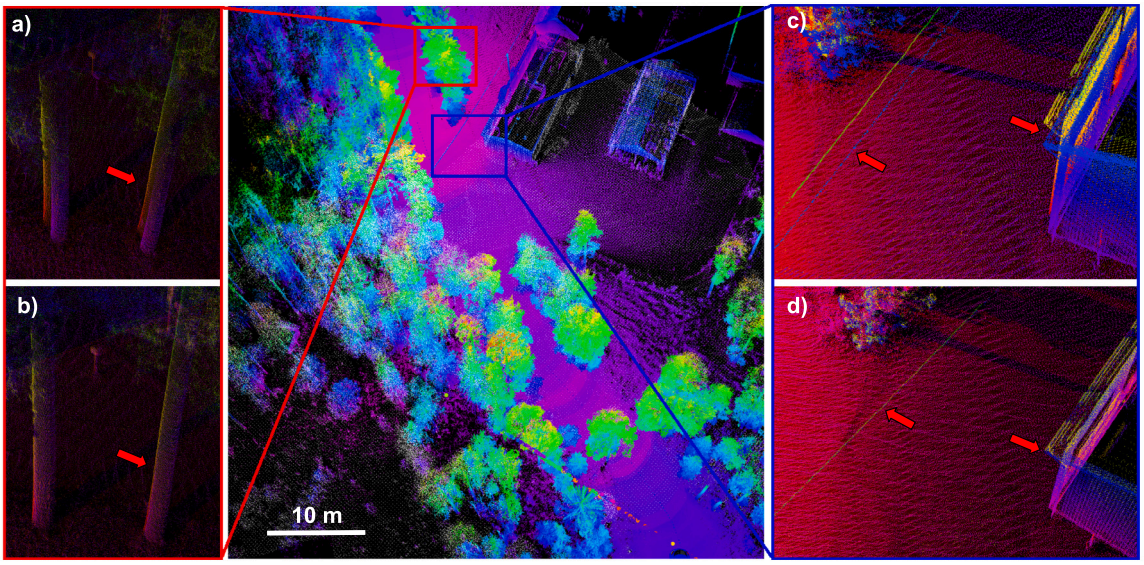


Fig. 15. The comparison of the 2D VUX georeferencing with before and after extrinsic calibration with map refinement. The comparison is between the 2D VUX and the reference 3D MLS, which contains combined Hesai and ALS data. The red arrows in enlarged windows (a) and (c) show the resulting misalignment in trees, electric cables, and buildings achieved with the 'GNSS–INS + MLS' configuration before the extrinsic refinement. The enlarged windows (b) and (d) show the resulting point cloud after the extrinsic estimation and batch update correction done with 'GNSS–INS + MLS' configuration. The 3D MLS map is coloured by the intensity values (purple point cloud), and the VUX point cloud is coloured by the z-axis values. The scale of the map is in the bottom left corner.

By linearising, we can approximate the errors as shown in Eq. (A.4):

$$\begin{cases} \mathbf{e}_{ij}^{\text{odom}} \approx \mathbf{e}_{ij}^{\text{odom}} + \mathbf{J}_i \delta\boldsymbol{\xi}_i + \mathbf{J}_j \delta\boldsymbol{\xi}_j \\ \mathbf{e}_{ik}^{\text{plane}} \approx \mathbf{e}_{ik}^{\text{plane}} + \mathbf{J}_k \delta\boldsymbol{\xi}_i \end{cases} \tag{A.4}$$

Then, the Jacobians are computed with Eq. (A.5):

$$\begin{cases} \mathbf{J}_i = -\text{Adj}\left(\hat{\mathbf{T}}_{ij}^{-1} \mathbf{T}_i^{-1} \mathbf{T}_j\right), \quad \mathbf{J}_j = \mathbf{I}_6 \in \mathbb{R}^{6\times 6} \\ \mathbf{J}_k = \boldsymbol{\nu}_k^{\top} \begin{bmatrix} \mathbf{I}_3 & -[\mathbf{R}_i \mathbf{s}_{ik}]_{\times} \end{bmatrix} \in \mathbb{R}^{1\times 6} \end{cases} \tag{A.5}$$

where the adjoint of a transformation $\mathbf{T} = [\mathbf{R}|\mathbf{p}] \in SE(3)$ is computed with Eq. (A.6):

$$\text{Adj}(\mathbf{T}) = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ [\mathbf{p}]_{\times}\mathbf{R} & \mathbf{R} \end{bmatrix} \tag{A.6}$$

with $[\cdot]_{\times}$ skew-symmetric matrix operator.

Appendix B. Robust kernel

The Cauchy loss function (Chebrolu et al., 2021) is defined in Eq. (B.1):

$$\rho(\mathbf{e}) = c^2 \log\left(1 + \frac{\mathbf{e}^{\top}\mathbf{e}}{c^2}\right) \tag{B.1}$$

where $\mathbf{e}$ is the residual and c is a user defined robustness parameter (threshold). The weight (first derivative) of the Cauchy loss function is given by the Eq. (B.2):

$$\kappa(\mathbf{e}) = \frac{c^2}{c^2 + \mathbf{e}^{\top}\mathbf{e}}. \tag{B.2}$$

Appendix C. Gauss–Newton step for scan registration

We update the pose using Eq. (A.3). The Jacobians of the point-to-point and point-to-plane cost functions are estimated as in Eq. (C.1):

$$\begin{cases} \mathbf{J}_{point} = \begin{bmatrix} \mathbf{I}_3 & -[\mathbf{R}\mathbf{p}]_{\times} \end{bmatrix} \in \mathbb{R}^{3\times 6} \\ \mathbf{J}_{plane} = \boldsymbol{\nu}^{\top} \begin{bmatrix} \mathbf{I}_3 & -[\mathbf{R}\mathbf{p}]_{\times} \end{bmatrix} \in \mathbb{R}^{1\times 6} \end{cases} \tag{C.1}$$

The Gauss–Newton matrices $\mathbf{H} \leftarrow \mathbf{0} \in \mathbb{R}^{6\times 6}$ and $\mathbf{b} \leftarrow \mathbf{0} \in \mathbb{R}^{6\times 1}$ are updated for all correspondences, as shown in Eq. (C.2):

$$\begin{cases} \mathbf{H} \mathrel{+}= \mathbf{J}_{\text{plane}_i}^{\top} \alpha \mathbf{J}_{\text{plane}_i}, \\ \mathbf{b} \mathrel{+}= \mathbf{J}_{\text{plane}_i}^{\top} \alpha \mathbf{e}_{\text{plane}_i} \quad \forall i \in \mathcal{I}_{\text{plane}} \\ \mathbf{H} \mathrel{+}= \mathbf{J}_{\text{point}_j}^{\top} \alpha \mathbf{J}_{\text{point}_j}, \\ \mathbf{b} \mathrel{+}= \mathbf{J}_{\text{point}_j}^{\top} \alpha \mathbf{e}_{\text{point}_j} \quad \forall j \in \mathcal{I}_{\text{point}} \end{cases} \tag{C.2}$$

where $\alpha = \kappa(\mathbf{e}_{\text{point}})$ or $\alpha = \kappa(\mathbf{e}_{\text{plane}})$ is the Cauchy robust kernel function to downsample the effect of outliers. One update step comes from solving Eq. (C.3):

$$\mathbf{H}\delta\boldsymbol{\xi} = -\mathbf{b}. \tag{C.3}$$

Appendix D. Initial approximate extrinsic rotation between two sensors

In this section, we present an example of how to compute a rough initial extrinsic rotation between two sensor frames. This step is essential for our map-based projection extrinsic calibration. As shown in Fig. 4 and further illustrated in Fig. D.16, we consider two sensor frames: $\mathcal{A}$, mounted at the front of the car, and $\mathcal{B}$, located at the rear. The coordinate axes of each sensor are aligned as follows:

- **Sensor frame $\mathcal{A}$ (3D MLS):**
  - $\hat{x}_{\mathcal{A}}$: right $= [1, 0, 0]^{\top}$
  - $\hat{y}_{\mathcal{A}}$: forward $= [0, 1, 0]^{\top}$
  - $\hat{z}_{\mathcal{A}}$: up $= [0, 0, 1]^{\top}$
- **Sensor frame $\mathcal{B}$ (2D LiDAR):**
  - $\hat{x}_{\mathcal{B}}$: up $\approx \hat{z}_{\mathcal{A}} = [0, 0, 1]^{\top}$
  - $\hat{y}_{\mathcal{B}}$: left $\approx -\hat{x}_{\mathcal{A}} = [-1, 0, 0]^{\top}$
  - $\hat{z}_{\mathcal{B}}$: backward $\approx -\hat{y}_{\mathcal{A}} = [0, -1, 0]^{\top}$

Given the approximate axis alignment between the two frames, we can construct the rotation matrix that transforms a 3D point from frame $\mathcal{B}$ to frame $\mathcal{A}$, denoted as ${}^{\mathcal{A}}\mathbf{R}_{\mathcal{B}} \in \mathbb{R}^{3\times 3}$. This matrix is constructed by

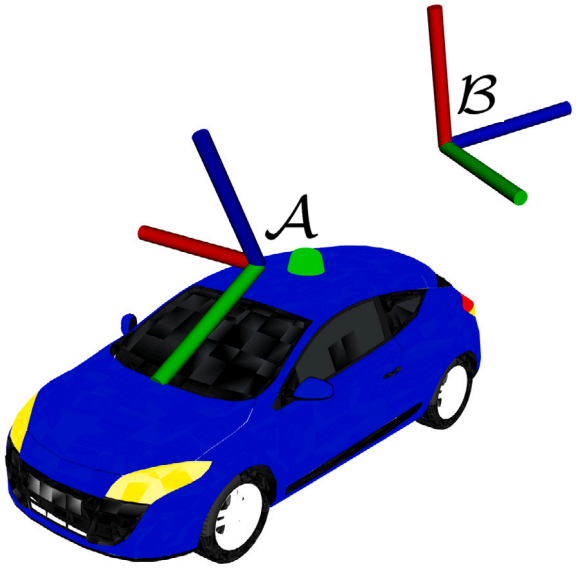


Fig. D.16. Initial extrinsic rotation from frame $\mathcal{B}$ to frame $\mathcal{A}$.

expressing the unit vectors of frame $\mathcal{B}$ in the coordinate system of frame $\mathcal{A}$:

$$ {}^{\mathcal{A}}\mathbf{R}_{\mathcal{B}} = \begin{bmatrix} \hat{x}_{\mathcal{B}} \cdot \hat{x}_{\mathcal{A}} & \hat{y}_{\mathcal{B}} \cdot \hat{x}_{\mathcal{A}} & \hat{z}_{\mathcal{B}} \cdot \hat{x}_{\mathcal{A}} \\ \hat{x}_{\mathcal{B}} \cdot \hat{y}_{\mathcal{A}} & \hat{y}_{\mathcal{B}} \cdot \hat{y}_{\mathcal{A}} & \hat{z}_{\mathcal{B}} \cdot \hat{y}_{\mathcal{A}} \\ \hat{x}_{\mathcal{B}} \cdot \hat{z}_{\mathcal{A}} & \hat{y}_{\mathcal{B}} \cdot \hat{z}_{\mathcal{A}} & \hat{z}_{\mathcal{B}} \cdot \hat{z}_{\mathcal{A}} \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \tag{D.1} $$

Thus, any point expressed in frame $\mathcal{B}$ can be rotated to frame $\mathcal{A}$ as shown in Eq. (D.2):

$$ {}^{\mathcal{A}}\mathbf{p} = {}^{\mathcal{A}}\mathbf{R}_{\mathcal{B}}\, {}^{\mathcal{B}}\mathbf{p}. \tag{D.2} $$

References

Baltsavias, Emanuel P., 1999. Airborne laser scanning: existing systems and firms and other resources. ISPRS J. Photogramm. Remote Sens. 54 (2–3), 164–198.

Bosse, Michael, Zlot, Robert, 2009. Continuous 3D scan-matching with a spinning 2D laser. In: 2009 IEEE International Conference on Robotics and Automation. pp. 4312–4319.

Chebrolu, Nived, Läbe, Thomas, Vysotska, Olga, Behley, Jens, Stachniss, Cyrill, 2021. Adaptive robust kernels for non-linear least squares problems. IEEE Robot. Autom. Lett. 6 (2), 2240–2247. http://dx.doi.org/10.1109/LRA.2021.3061331.

Chen, Mengxiao, Yang, Shaowu, Yi, Xiaodong, Wu, Dan, 2017. Real-time 3D mapping using a 2D laser scanner and IMU-aided visual SLAM. In: 2017 IEEE International Conference on Real-Time Computing and Robotics. RCAR, pp. 297–302. http://dx.doi.org/10.1109/RCAR.2017.8311877.

Cheng, Xiaolong, Cheng, Xiaojun, Li, Quan, Ma, Liwei, 2018. Automatic registration of terrestrial and airborne point clouds using building outline features. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. PP, 1–11. http://dx.doi.org/10.1109/JSTARS.2017.2788054.

Cheng, Liang, Wu, Yang, Tong, Lihua, Chen, Yanming, Li, Manchun, 2015. Hierarchical registration method for airborne and vehicle LiDAR point cloud. Remote. Sens. 7 (10), 13921–13944.

Cohen, Israel, Huang, Yiteng, Chen, Jingdong, Benesty, Jacob, Benesty, Jacob, Chen, Jingdong, Huang, Yiteng, Cohen, Israel, 2009. Pearson correlation coefficient. Noise Reduct. Speech Process. 1–4.

Dai, Wenxia, Yang, Bisheng, Liang, Xinlian, Dong, Zhen, Huang, Ronggang, Wang, Yunsheng, Li, Wuyan, 2019. Automated fusion of forest airborne and terrestrial point clouds through canopy density analysis. ISPRS J. Photogramm. Remote Sens. (ISSN: 0924-2716) 156, 94–107. http://dx.doi.org/10.1016/j.isprsjprs.2019.08.008.

DiCiccio, Thomas J., Efron, Bradley, 1996. Bootstrap confidence intervals. Statist. Sci. 11 (3), 189–228.

El Issaoui, Aimad, Feng, Ziyi, Lehtomäki, Matti, Hyyppä, Eric, Hyyppä, Hannu, Kaartinen, Harri, Kukko, Antero, Hyyppä, Juha, 2021. Feasibility of mobile laser scanning towards operational accurate road rut depth measurements. Sensors 21 (4), 1180.

Feng, Tong, Chen, Shilin, Feng, Zhongke, Shen, Chaoyong, Tian, Yi, 2021. Effects of canopy and multi-epoch observations on single-point positioning errors of a GNSS in coniferous and broadleaved forests. Remote. Sens. 13 (12).

Fernández-Moral, Eduardo, Gonzalez-Jimenez, Javier, Arévalo, Vicente, 2015. Extrinsic calibration of 2D laser rangefinders from perpendicular plane observations. Int. J. Robot. Res. 34 (11), 1401–1417.

He, Dongjiao, Xu, Wei, Zhang, Fu, 2021. Embedding manifold structures into Kalman filters. URL https://api.semanticscholar.org/CorpusID:231847324. ArXiv, abs/2102.03804.

Hesai, 2024. Pandar 32: 32-channel 360° spinning mid-range LiDAR. URL https://www.hesaitech.com. (Accessed 10 September 2024).

Horaud, Radu, Dornaika, Fadi, 1995. Hand-eye calibration. Int. J. Robot. Res. 14 (3), 195–210.

Horn, Markus, Wodtko, Thomas, Buchholz, Michael, Dietmayer, Klaus, 2021. Online extrinsic calibration based on per-sensor ego-motion using dual quaternions. IEEE Robot. Autom. Lett. 6 (2), 982–989.

Huang, Huaiyang, Sun, Yuxiang, Wu, Jin, Jiao, Jianhao, Hu, Xiangcheng, Zheng, Linwei, Wang, Lujia, Liu, Ming, 2021. On bundle adjustment for multiview point cloud registration. IEEE Robot. Autom. Lett. 6 (4), 8269–8276.

Hyyppä, Eric, Hyyppä, Juha, Hakala, Teemu, Kukko, Antero, Wulder, Michael A, White, Joanne C, Pyörälä, Jiri, Yu, Xiaowei, Wang, Yunsheng, Virtanen, Juho-Pekka, et al., 2020a. Under-canopy UAV laser scanning for accurate forest field measurements. ISPRS J. Photogramm. Remote Sens. 164, 41–60.

Hyyppä, Eric, Kukko, Antero, Kaartinen, Harri, Yu, Xiaowei, Muhojoki, Jesse, Hakala, Teemu, Hyyppä, Juha, 2022. Direct and automatic measurements of stem curve and volume using a high-resolution airborne laser scanning system. Sci. Remote. Sens. 5, 100050.

Hyyppä, Eric, Kukko, Antero, Kaijaluoto, Risto, White, Joanne C., Wulder, Michael A., Pyörälä, Jiri, Liang, Xinlian, Yu, Xiaowei, Wang, Yunsheng, Kaartinen, Harri, et al., 2020b. Accurate derivation of stem curve and volume using backpack mobile laser scanning. ISPRS J. Photogramm. Remote Sens. 161, 246–262.

Hyyppä, Eric, Muhojoki, Jesse, Yu, Xiaowei, Kukko, Antero, Kaartinen, Harri, Hyyppä, Juha, 2021. Efficient coarse registration method using translation-and rotation-invariant local descriptors towards fully automated forest inventory. ISPRS Open J. Photogramm. Remote. Sens. 2, 100007.

Kaartinen, Harri, Hyyppä, Juha, Vastaranta, Mikko, Kukko, Antero, Jaakkola, Anttoni, Yu, Xiaowei, Pyörälä, Jiri, Liang, Xinlian, Liu, Jingbin, Wang, Yungshen, Kaijaluoto, Risto, Melkas, Timo, Holopainen, Markus, Hyyppä, Hannu, 2015a. Accuracy of kinematic positioning using global satellite navigation systems under forest canopies. Forests 6 (9), 3218–3236.

Kaartinen, Harri, Hyyppä, Juha, Vastaranta, Mikko, Kukko, Antero, Jaakkola, Anttoni, Yu, Xiaowei, Pyörälä, Jiri, Liang, Xinlian, Liu, Jingbin, Wang, Yungshen, et al., 2015b. Accuracy of kinematic positioning using global satellite navigation systems under forest canopies. Forests 6 (9), 3218–3236.

Kaess, Michael, Johannsson, Hordur, Roberts, Richard, Ila, Viorela, Leonard, John, Dellaert, Frank, 2011. iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In: 2011 IEEE International Conference on Robotics and Automation. IEEE, pp. 3281–3288.

Kang, Xueyang, Yin, Shengjiong, Fen, Yinglong, 2018. 3D reconstruction & assessment framework based on affordable 2D LiDAR. In: 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. AIM, pp. 292–297. http://dx.doi.org/10.1109/AIM.2018.8452242.

Kukko, Antero, Kaartinen, Harri, Hyyppä, Juha, Chen, Yuwei, 2012. Multiplatform approach to mobile laser scanning. Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. 39, 483–488.

Kukko, Antero, Kaijaluoto, Risto, Kaartinen, Harri, Lehtola, Ville V., Jaakkola, Anttoni, Hyyppä, Juha, 2017. Graph SLAM correction for single scanner MLS forest data under boreal forest canopy. ISPRS J. Photogramm. Remote Sens. 132, 199–209.

Lee, Yongkyu, Woo, Heesung, Lee, Jung-Soo, 2022. Forest inventory assessment using integrated light detection and ranging (LiDAR) systems: Merged point cloud of airborne and mobile laser scanning systems. Sensors Mater. 34.

Li, Xiaojun, Li, Xinrui, Hu, Guiting, Niu, Qi, Xu, Luping, 2024a. A low-cost 3D mapping system for indoor scenes based on 2D LiDAR and monocular cameras. Remote. Sens. (ISSN: 2072-4292) 16 (24), http://dx.doi.org/10.3390/rs16244712, URL https://www.mdpi.com/2072-4292/16/24/4712.

Li, Ruyi, Zhang, Xuebo, Zhang, Shiyong, Yuan, Jing, Liu, Hui, Wu, Songyang, 2024b. BA-LIOM: Tightly coupled laser-inertial odometry and mapping with bundle adjustment. Robotica 42 (3), 684–700.

Liang, Xinlian, Hyyppä, Juha, Kukko, Antero, Kaartinen, Harri, Jaakkola, Anttoni, Yu, Xiaowei, 2014. The use of a mobile laser scanning system for mapping large forest plots. IEEE Geosci. Remote. Sens. Lett. 11 (9), 1504–1508. http://dx.doi.org/10.1109/LGRS.2013.2297418.

Lin, Kuen-Han, Chang, Chun-Hua, Wang, Chieh-Chih, et al., 2012. Mapping and localization in 3D environments using a 2D laser scanner and a stereo camera. J. Inf. Sci. Eng. 28 (1).

Liu, Zheng, Li, Haotian, Yuan, Chongjian, Liu, Xiyuan, Lin, Jiarong, Li, Rundong, Zheng, Chunran, Zhou, Bingyang, Liu, Wenyi, Zhang, Fu, 2024. Voxel-SLAM: A complete, accurate, and versatile LiDAR-inertial SLAM system. arXiv preprint arXiv:2410.08935.

Liu, Zheng, Zhang, Fu, 2021. Balm: Bundle adjustment for LiDAR mapping. IEEE Robot. Autom. Lett. 6 (2), 3184–3191.

Maddern, Will, Harrison, Alastair, Newman, Paul, 2012. Lost in translation (and rotation): Rapid extrinsic calibration for 2D and 3D LiDARs. In: 2012 IEEE International Conference on Robotics and Automation. IEEE, pp. 3096–3102.

Matikainen, Leena, Pandžić, Miloš, Li, Fashuai, Karila, Kirsi, Hyyppä, Juha, Litkey, Paula, Kukko, Antero, Lehtomäki, Matti, Karjalainen, Mika, Puttonen, Eetu, 2019. Toward utilizing multitemporal multispectral airborne laser scanning, sentinel-2, and mobile laser scanning in map updating. J. Appl. Remote. Sens. 13 (4), 044504–044504.

Muhojoki, Jesse, Hakala, Teemu, Kukko, Antero, Kaartinen, Harri, Hyyppä, Juha, 2024. Comparing positioning accuracy of mobile laser scanning systems under a forest canopy. Sci. Remote. Sens. 9, 100121.

Napier, Ashley, Corke, Peter, Newman, Paul, 2013. Cross-calibration of push-broom 2D LiDARs and cameras in natural scenes. In: 2013 IEEE International Conference on Robotics and Automation. IEEE, pp. 3679–3684.

Nie, Xin, Gong, Jun, Cheng, Jintao, Tang, Xiaoyu, Zhang, Yuanfang, 2023. Two-step self-calibration of LiDAR-GPS/IMU based on hand-eye method. Symmetry 15 (2), 254.

Nocedal, Jorge, Wright, Stephen J., 2006. Numerical Optimization, second ed. Springer Science & Business Media, ISBN: 978-0387303031.

NovAtel, 2024. Inertial explorer GNSS+INS post-processing. https://novatel.com/products/waypoint-post-processing-software/inertial-explorer. (Accessed 20 September 2024).

NovAtel CPT7, 2022. CPT7 installation and operation manual. URL https://docs.novatel.com. (Accessed 10 September 2024).

NovAtel Inc., 2023. Inertial Explorer GNSS+INS Post-Processing Software. NovAtel Inc., https://novatel.com/products/waypoint-software/inertial-explorer.

Pi, Jiahao, Yan, Guohang, Wang, Chengjie, Cai, Xinyu, Shi, Botian, 2024. An extrinsic calibration method between LiDAR and GNSS/INS for autonomous driving. In: 2024 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 14520–14526.

Polewski, Przemyslaw, Yao, Wei, Cao, Lin, Gao, Sha, 2019. Marker-free coregistration of UAV and backpack LiDAR point clouds in forested areas. ISPRS J. Photogramm. Remote Sens. (ISSN: 0924-2716) 147, 307–318. http://dx.doi.org/10.1016/j.isprsjprs.2018.11.020, URL https://www.sciencedirect.com/science/article/pii/S092427161830323X.

Puente, I., González-Jorge, H., Arias, P., Armesto, J., 2011. Land-based mobile laser scanning systems: A review. Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. XXXVIII-5/W12, 163–168. http://dx.doi.org/10.5194/isprsarchives-XXXVIII-5-W12-163-2011, URL https://isprs-archives.copernicus.org/articles/XXXVIII-5-W12/163/2011/.

RIEGL, 2023. RIEGL VUX series: Airborne laser scanner manual. URL https://www.riegl.com. (Accessed 10 September 2024).

Roca, David, Martínez-Sánchez, Joaquín, Lagüela, Susana, Arias, Pedro, 2016. Novel aerial 3D mapping system based on UAV platforms and 2D laser scanners. J. Sensors 2016 (1), 4158370.

Schadler, Mark, Stückler, Jörg, Behnke, Sven, 2014. Rough terrain 3D mapping and navigation using a continuously rotating 2D laser scanner. KI-Künstliche Intell. 28, 93–99.

Segal, Aleksandr, Hähnel, Dirk, Thrun, Sebastian, 2009. Generalized-ICP. http://dx.doi.org/10.15607/RSS.2009.V.021.

Shan, Tixiao, Englot, Brendan, 2018. LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, pp. 4758–4765. http://dx.doi.org/10.1109/IROS.2018.8594299.

Shan, Tixiao, Englot, Brendan, Meyers, Drew, Wang, Wei, Ratti, Carlo, Rus, Daniela, 2020. LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, pp. 5135–5142. http://dx.doi.org/10.1109/IROS45743.2020.9341176.

Sigrist, P., Coppin, P., Hermy, M., 1999. Impact of forest canopy on quality and accuracy of GPS measurements. Int. J. Remote Sens. 20 (18), 3595–3610.

Sola, Joan, Deray, Jeremie, Atchuthan, Dinesh, 2018. A micro Lie theory for state estimation in robotics. arXiv preprint arXiv:1812.01537.

Välimäki, Tuomas, Garigipati, Bharath, Ghabcheloo, Reza, 2023. Motion-based extrinsic sensor-to-sensor calibration: Effect of reference frame selection for new and existing methods. Sensors 23 (7), 3740.

Vezeteu, Eugeniu, 2020. Stereo-Camera-LiDAR Calibration for Autonomous Driving. (Master's thesis). Aalto University, Espoo, Finland, URL https://aaltodoc.aalto.fi/items/5a4d5b82-a6ab-4e43-b42c-1808909b06b5.

Vezeteu, Eugeniu, Issaoui, Aimad El, Hyyti, Heikki, Hakala, Teemu, Muhojoki, Jesse, Hyyppä, Eric, Kukko, Antero, Kaartinen, Harri, Kyrki, Ville, Hyyppä, Juha, 2025. Direct integration of ALS and MLS for real-time localization and mapping. ISPRS Open J. Photogramm. Remote. Sens. (ISSN: 2667-3932) 16, 100088. http://dx.doi.org/10.1016/j.ophoto.2025.100088, URL https://www.sciencedirect.com/science/article/pii/S2667393225000079.

Vizzo, Ignacio, Guadagnino, Tiziano, Mersch, Benedikt, Wiesmann, Louis, Behley, Jens, Stachniss, Cyrill, 2023. KISS-ICP: In defense of point-to-point ICP – simple, accurate, and robust registration if done the right way. IEEE Robot. Autom. Lett. (RA-L) 8 (2), 1029–1036. http://dx.doi.org/10.1109/LRA.2023.3236571.

Wang, H., Wang, C., Chen, C., Xie, L., 2020. F-LOAM : Fast LiDAR odometry and mapping. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS.

Wehr, Aloysius, Lohr, Uwe, 1999. Airborne laser scanning—an introduction and overview. ISPRS J. Photogramm. Remote Sens. 54 (2–3), 68–82.

Xu, Wei, Cai, Yixi, He, Dongjiao, Lin, Jiarong, Zhang, Fu, 2022. FAST-LIO2: Fast direct LiDAR-inertial odometry. IEEE Trans. Robot. 38, 1–21. http://dx.doi.org/10.1109/TRO.2022.3141876.

Xu, Wei, Zhang, Fu, 2020. FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter. IEEE Robot. Autom. Lett. 6, 3317–3324, URL https://api.semanticscholar.org/CorpusID:223953478.

Yin, Deyu, Liu, Jingbin, Wu, Teng, Liu, Keke, Hyyppä, Juha, Chen, Ruizhi, 2018. Extrinsic calibration of 2D laser rangefinders using an existing cuboid-shaped corridor as the reference. Sensors 18 (12), 4371.

Yuan, Chongjian, Xu, Wei, Liu, Xiyuan, Hong, Xiaoping, Zhang, Fu, 2022. Efficient and probabilistic adaptive voxel mapping for accurate online LiDAR odometry. IEEE Robot. Autom. Lett. 7 (3), 8518–8525.

Zhang, Zhengyou, 2014. Iterative closest point (ICP). In: Ikeuchi, Katsushi (Ed.), Computer Vision: A Reference Guide. Springer US, Boston, MA, ISBN: 978-0-387-31439-6, pp. 433–434. http://dx.doi.org/10.1007/978-0-387-31439-6_179.

Zhang, Ji, Singh, Sanjiv, 2014. LOAM: Lidar odometry and mapping in real-time. In: Robotics: Science and Systems. URL https://api.semanticscholar.org/CorpusID:18612391.

Zhou, Pengwei, Guo, Xuexun, Pei, Xiaofei, Chen, Ci, 2021. T-LOAM: Truncated least squares LiDAR-only odometry and mapping in real time. IEEE Trans. Geosci. Remote Sens. 1–13. http://dx.doi.org/10.1109/TGRS.2021.3083606.

Zhou, Tian, Ravi, Radhika, Lin, Yi-Chun, Manish, Raja, Fei, Songlin, Habib, Ayman, 2023. In situ calibration and trajectory enhancement of UAV and backpack LiDAR systems for fine-resolution forest inventory. Remote. Sens. 15 (11), 2799.