ISPRS Journal of Photogrammetry and Remote Sensing 215 (2024) 62–79

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

A flexible trajectory estimation methodology for kinematic laser scanning

Florian Pöppl [a,*], Andreas Ullrich [b], Gottfried Mandlburger [a], Norbert Pfeifer [a]

[a] Department of Geodesy and Geoinformation, TU Wien, Wiedner Hauptstraße 8/E120, 1040 Wien, Austria
[b] RIEGL Laser Measurement Systems GmbH, Riedenburgstraße 48, 3580 Horn, Austria

ARTICLE INFO

Keywords:
Georeferencing
Sensor orientation
Sensor fusion
Tightly-coupled LiDAR
Multi-platform adjustment
Integrated trajectory estimation

ABSTRACT

Kinematic laser scanning is a widely-used surveying technique based on light detection and ranging (LiDAR) that enables efficient data acquisition by mounting the laser scanner on a moving platform. In order to obtain a georeferenced point cloud, the trajectory of the moving platform must be accurately known. To this end, most commercial laser scanning systems comprise an inertial measurement unit (IMU) and a global navigation satellite system (GNSS) receiver and antenna. Trajectory estimation is then the task of determining the platform's position and orientation by integrating measurements from the IMU, GNSS, and possibly the laser scanner itself. Here, we present a comprehensive approach to trajectory estimation for kinematic laser scanning, based on batch least-squares adjustment incorporating pre-processed GNSS positions, raw IMU data and plane-based LiDAR correspondences in a single estimation procedure. In comparison to the classic workflow of Kalman filtering followed by strip adjustment, this is a holistic approach with tight coupling of IMU and LiDAR. For the latter, we extend the data-derived stochastic model for the LiDAR plane observations with prior knowledge of the LiDAR measurement process. The proposed trajectory estimation approach is flexible and allows different system configurations as well as joint registration of multiple independent kinematic datasets. This is demonstrated using as a practical example a combined dataset consisting of two independent data acquisitions from crewed aircraft and uncrewed aerial vehicle. All measurements from both datasets are jointly adjusted in order to obtain a single high-quality point cloud, without the need for ground control. The performance of this approach is evaluated in terms of point cloud consistency, precision, and accuracy. The latter is done by comparison to terrestrially surveyed reference data on the ground. The results show improved consistency, accuracy, and precision compared to a standard workflow, with the RMSE reduced from 7.43 cm to 3.85 cm w.r.t. the reference data surfaces, and the point-to-plane standard deviation on the surfaces reduced from 3.01 cm to 2.44 cm. Although a direct comparison to the state-of-the-art can only be made with caution, we can state that the suggested method performs better in terms of point cloud consistency and precision, while at the same time achieving better absolute accuracy.

1. Introduction

Kinematic laser scanning is a standard surveying technique, allowing for efficient acquisition of highly accurate 3D point clouds. Laser scanning is based on measuring the range to a target via light detection and ranging (LiDAR), while simultaneously varying the direction of the laser beam. Laser scanning measurements are thus made in polar coordinates (range and one or two angles), which are then transformed into scanner-referenced Cartesian coordinates. In kinematic laser scanning, the laser scanner is mounted on a moving carrier platform (e.g., car, drone, or airplane). The platform's motion must be accounted for in order to reference the point cloud to a well-defined earth-fixed coordinate system. This requires knowledge of the platform trajectory (position and orientation over time), which may be obtained by integrating data from auxiliary navigation systems. Almost all survey-grade kinematic mapping systems therefore comprise at least a global navigation satellite system (GNSS) receiver/antenna and an inertial measurement unit (IMU). These technologies complement each other, as GNSS provides absolute positioning and the IMU provides relative position and orientation through strap-down inertial navigation. Inertial sensors suffer from time-varying measurement errors, which result in drift in the trajectory if not properly accounted for. Some errors may be compensated in-run by fusing the IMU and GNSS measurements in a Kalman filter. Depending on the platform motion and the characteristics of the sensors involved, the resulting trajectory often still contains significant errors. These errors manifest in the point

* Corresponding author.
E-mail addresses: florian.poeppl@geo.tuwien.ac.at (F. Pöppl), aullrich@riegl.com (A. Ullrich), gottfried.mandlburger@geo.tuwien.ac.at (G. Mandlburger), norbert.pfeifer@geo.tuwien.ac.at (N. Pfeifer).

https://doi.org/10.1016/j.isprsjprs.2024.06.014
Received 21 February 2024; Received in revised form 17 June 2024; Accepted 18 June 2024
Available online 3 July 2024

clouds as discrepancies in areas which are scanned multiple times, and as discrepancies w.r.t. reference data.

1.1. Related & previous work

The industry standard for kinematic data processing in surveying is Kalman filtering of GNSS/IMU data, followed by an integrated sensor orientation (Toth and Jóźków, 2016), specifically bundle adjustment in photogrammetry or strip adjustment in laser scanning. In a bundle or strip adjustment the imaging sensor data is used, possibly together with ground control, to improve the sensor orientation. Building on stand-alone strip adjustment and bundle adjustment approaches, hybrid approaches combine LiDAR and camera measurements in a single adjustment (e.g., Glira et al., 2019; Jonassen et al., 2023). This corresponds to a tight coupling of camera and laser scanning data. Recent works have presented holistic integration methods, incorporating the laser scanning data into the trajectory estimation, thereby simultaneously estimating the trajectory and system calibration with the goal of improving point cloud quality. In such approaches, all available measurements are incorporated into a single estimation procedure. This integrated approach to sensor fusion has recently seen increasing use, especially in robotics (e.g., Chang et al., 2019; Beuchert et al., 2023), but also in remote sensing. Here, it has been successfully applied to bundle adjustment (Cucci et al., 2017b) and airborne laser scanning (Brun et al., 2022; Mouzakidou et al., 2024).

The term *trajectory estimation* refers here to all such methods that attempt to recover position and orientation from noisy measurements. Pöppl et al. (2023b) gives an overview of trajectory estimation methods and establishes a common framework for different approaches. It is instructive to differentiate trajectory estimation methods on the basis of (a) trajectory model and (b) estimation procedure. The *trajectory* is often parametrized discretely, with position and orientation provided at given points in time (*epochs*). In this case, the state needs to be either estimated for each measurement time, or suitably interpolated. Recently, continuous-time representations have gained traction due to their inherent interpolation capability (Furgale et al., 2015; Cioffi et al., 2022). As a continuous-time function, the trajectory can be evaluated at any given time therefore allowing straightforward integration of asynchronous measurements. Two main types of continuous-time trajectory representations are Gaussian process-based models and spline-based models, a comparison of which is given in Johnson et al. (2024). Here, we use spline-based continuous-time trajectories. In addition to being continuous, higher-order splines are also differentiable allowing the analytic computation of acceleration and angular velocity as required for forming the inertial measurement equations. The *estimation* itself may be performed sequentially or in batch, i.e., using filtering or optimization-based approaches. On the one hand, filters process measurements sequentially, and these measurements are used epoch-by-epoch to update the estimated parameters. Filters may be run forwards and backwards to obtain a smoothed estimate incorporating both past and future information. For non-linear problems, this is generally not optimal as the linearization is consequently also only performed sequentially. On the other hand, one may consider all epochs simultaneously. Specifically, the estimation is posed as a Bayesian maximum a-posteriori (MAP) problem (cf. Barfoot, 2017). Assuming Gaussian errors, this further simplifies to a batch non-linear least-squares (NLS) problem. In this case, linearization is performed iteratively and repeatedly for all epochs, thus reducing linearization errors (cf. Strasdat et al., 2012). The drawback is the drastically increased computational effort, as the batch approach involves repeated solving of very large systems of equations. For some applications, the size of the system can quickly become prohibitive. Performance may be improved by employing IMU pre-integration (Forster et al., 2015) to reduce the number of measurements, and by selectively marginalizing certain parameters in a sliding-window (Lv et al., 2023) or key frame approach (Leutenegger et al., 2015) to reduce the number of parameters which are optimized simultaneously. For continuous-time trajectories, Lv et al. (2021) propose a hybrid approach where the trajectory is locally parametrized by splines, but global optimization is performed only for selected keyframe poses.

These techniques are popular in simultaneous localization and mapping (SLAM), where runtime efficiency is paramount due to limited available computational resources and the inherent requirements for near real-time processing. However, both pre-integration and marginalization trade off accuracy in return for processing speed. Generally, above-mentioned SLAM approaches focus on typical robotics platforms (e.g., handheld devices, wheeled robots, cars or small multicopters and consumer- or industrial-grade sensors). Here, in contrast to SLAM techniques, we explicitly target post-processing workflows, mapping applications with cm-level accuracy requirements, and the platforms typically used in this context (e.g., larger uncrewed aerial vehicles or crewed fixed-wing aircraft together with survey-grade laser scanners and navigation sensors).

To this end, we propose a holistic methodology for GNSS/IMU/LiDAR-based trajectory estimation, which aims to be rigorous in terms of modelling, yet *flexible* in terms of application. Through formulation of all measurement equations within a common adjustment framework, it is easily adapted to various system configurations and different combinations of sensors. Our approach is based on tight coupling of LiDAR and IMU, with loose coupling of GNSS. Integration of raw GNSS code and carrier-phase measurements in a similar fusion architecture has also been demonstrated (Li et al., 2023), but this significantly increases model complexity and computational effort. While the loose coupling is less robust to signal outages and possibly less accurate than tightly coupled GNSS, it is significantly less complex and through decoupling of the GNSS processing allows usage of different existing GNSS processing software and processing strategies (RTK, PPK, PPP).

The pre-processed GNSS position measurements are incorporated in a single model, together with raw measurements from the IMU, and LiDAR-derived geometric measurements (see Fig. 1). The differences between modelled measurements and observed measurements are minimized in a MAP-derived NLS estimation. This is done globally, i.e., for the full batch of parameters with no marginalization of parameters. No IMU pre-integration is used, because we require the trajectory at the highest possible frequency in order to accurately represent all trajectory dynamics, including vibrations. Any unresolved vibrations would imply trajectory errors, and through georeferencing these errors propagate and negatively impact the quality of the point cloud.

Various applications have already been demonstrated for different use cases with different sensor configurations, from mobile (Pöppl et al., 2023d,e), to airborne (Pöppl et al., 2023a,c; Mandlburger et al., 2023a,b) laser scanning. Compared to a traditional georeferencing workflow of Kalman filtering followed by strip adjustment, the flexibility of the proposed method stems from the formulation of all observations as measurements with additive errors.[1] Together with the continuous-time trajectory model, this allows a combined adjustment of data from different sensors with possibly asynchronous measurements, including multi-GNSS, multi-IMU or multi-scanner set-ups. Also, data acquisitions from multiple independent platforms can be rigorously co-registered in one joint trajectory estimation.

1.2. Main contributions

In this article, we present the methodology itself and provide functional and stochastic models for GNSS, IMU and LiDAR measurements, including both instantaneous and integrated IMU measurements, support for dynamic GNSS lever-arms (e.g., for usage with gyro stabilization or gimbal mounts), and with improved stochastic modelling

[1] Kalman filter-based GNSS/IMU integration usually treats the IMU measurements as input in the process model. Strip adjustment (e.g., Glira et al., 2016) then considers only the LiDAR-derived observations in a separate step.

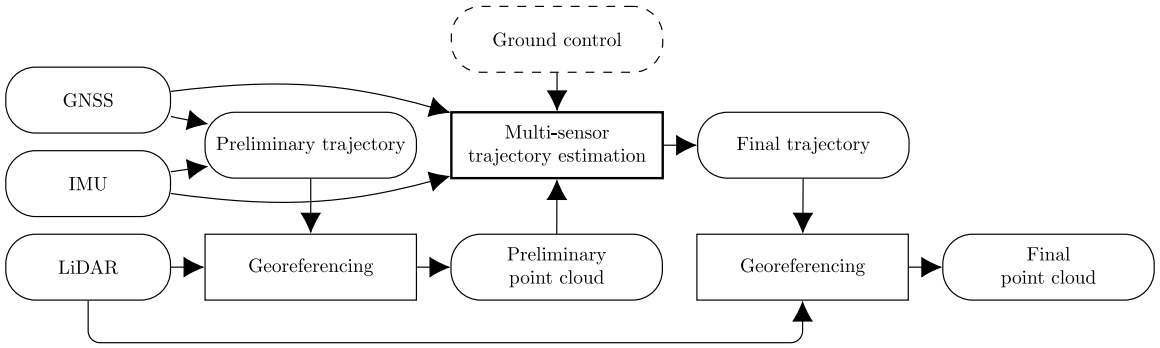


Fig. 1. Proposed trajectory estimation workflow.

of LiDAR-derived tie-plane observations. We extend the results from previous work and demonstrate a multi-platform adjustment approach for a combined ALS (*airborne laser scanning*) and ULS (*uncrewed or unmanned laser scanning*) dataset. Accuracy, precision and the impact of the LiDAR stochastic model are evaluated through analysis of strip differences and through comparison to independent terrestrially surveyed reference data on the ground.

Apart from the unified presentation of the estimation problem and measurement equations, the methodological contributions of this article are thus (1) tightly-coupled co-registration of kinematic laser scanning data from multiple platforms and without ground control, and (2) stochastic modelling of the LiDAR plane observations based on LiDAR measurement characteristics.

The rest of this paper is structured as follows: Section 2 presents the underlying methodology, including a brief mathematical background, details on the trajectory model and the measurement equations, and a discussion on numerical aspects. Section 3 then introduces the practical example, the specific structure of the multi-platform trajectory estimation and the processing workflow, and then analyses the results in terms of point cloud accuracy, precision, and consistency. Finally, a summary, concluding remarks and outlook are given in Section 4.

2. Methodology

2.1. Generic estimation methodology

The sensor fusion approach presented here was developed specifically for fusion of GNSS, IMU and LiDAR data. This is the primary focus of this work, but the estimation problem may be posed in a generic way (Pöppl et al., 2023b); similar methodology has also been used with different sensors (e.g., camera, odometer). The statistical framework is maximum a-posteriori estimation (MAP), where GNSS, IMU and LiDAR measurements are considered simultaneously, and all parameters are estimated jointly.

Consider the goal of estimating unknown parameters $\boldsymbol{x} \in \mathbb{R}^n$ from m noisy measurements $\widetilde{\boldsymbol{y}} \in \mathbb{R}^m$. The true values $\boldsymbol{y}$ are modelled as a function of the unknown parameters $\boldsymbol{x}$, so that

$$\boldsymbol{y} := f(\boldsymbol{x}), \tag{2.1}$$

and the measurements $\widetilde{\boldsymbol{y}} := \boldsymbol{y} + \boldsymbol{\epsilon}$ are *observations* of the modelled true value tainted by an additive measurement error

$$\underbrace{\widetilde{\boldsymbol{y}}}_{\text{measurement}} = \underbrace{f(\boldsymbol{x})}_{\text{model}} + \underbrace{\boldsymbol{\epsilon}}_{\text{error}}. \tag{2.2}$$

If the conditional probability density function $p(\widetilde{\boldsymbol{y}}|\boldsymbol{x})$ is known, this allows inference based on the posterior $p(\boldsymbol{x}|\widetilde{\boldsymbol{y}})$, i.e., the probability density of the parameters $\boldsymbol{x}$ given the measurements $\widetilde{\boldsymbol{y}}$. Using Bayes theorem, the posterior density is written as

$$p(\boldsymbol{x}|\widetilde{\boldsymbol{y}}) = \frac{p(\widetilde{\boldsymbol{y}}|\boldsymbol{x})p(\boldsymbol{x})}{p(\widetilde{\boldsymbol{y}})}, \tag{2.3}$$

where the a-priori density $p(\boldsymbol{x})$ contains any prior knowledge (belief) about $\boldsymbol{x}$. The *maximum a-posteriori* (MAP) estimate $\boldsymbol{x}^*_{\text{MAP}}$ is obtained by maximizing the posterior density $p(\cdot\,, \widetilde{\boldsymbol{y}})$

$$\boldsymbol{x}^*_{\text{MAP}} = \operatorname*{argmax}_{\boldsymbol{x}} p(\boldsymbol{x}|\widetilde{\boldsymbol{y}}) \tag{2.4}$$

$$= \operatorname*{argmax}_{\boldsymbol{x}} \frac{p(\widetilde{\boldsymbol{y}}|\boldsymbol{x})p(\boldsymbol{x})}{p(\widetilde{\boldsymbol{y}})} \tag{2.5}$$

$$= \operatorname*{argmax}_{\boldsymbol{x}} p(\widetilde{\boldsymbol{y}}|\boldsymbol{x})p(\boldsymbol{x}). \tag{2.6}$$

The parameters obtained by MAP estimation are thus the mode of the posterior density, in other words the parameters $\boldsymbol{x}$ *most likely* given the measurements $\widetilde{\boldsymbol{y}}$. Under the assumption of Gaussian measurement errors, the MAP estimate is identical to the non-linear least squares (NLS) estimate. Gaussian priors in the MAP framework may be included as additional observations in the NLS formulation. For Gaussian errors with zero mean and variance $\boldsymbol{\Sigma}$, the NLS estimator $\boldsymbol{x}^*_{\text{NLS}}$ is given by

$$\boldsymbol{x}^*_{\text{NLS}} = \operatorname*{argmin}_{\boldsymbol{x}} \; (\widetilde{\boldsymbol{y}} - f(\boldsymbol{x}))^T \, \boldsymbol{\Sigma}^{-1} \, (\widetilde{\boldsymbol{y}} - f(\boldsymbol{x})). \tag{2.7}$$

In our specific problem, we assume that the full measurement vector $\boldsymbol{y}$ is partitionable into k individual measurements $(\boldsymbol{y}_i)_{i=1}^k = (\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_k)$ with $\boldsymbol{y}_i \in \mathbb{R}^{m_i}$, $i \in 1, \dots, k$ and

$$\boldsymbol{y}_i = f_i(\boldsymbol{x}), \;\; i \in 1, \dots, k. \tag{2.8}$$

Further assuming the respective errors $\boldsymbol{\epsilon}_i = \widetilde{\boldsymbol{y}}_i - \boldsymbol{y}_i$ are uncorrelated with $\mathbb{V}(\boldsymbol{\epsilon}_i, \boldsymbol{\epsilon}_j) = \boldsymbol{0}$, $i \neq j$ and $\mathbb{V}(\boldsymbol{\epsilon}_i) = \boldsymbol{\Sigma}_i$, the NLS estimator $\boldsymbol{x}^*_{\text{NLS}}$ is obtained by minimizing the sum of squared residuals (SSR) over all k measurements

$$\begin{aligned} \boldsymbol{x}^*_{\text{NLS}} &= \operatorname*{argmin}_{\boldsymbol{x}} \; \text{SSR}(\boldsymbol{x}) \\ &= \operatorname*{argmin}_{\boldsymbol{x}} \; \sum_{i=1}^{k} (\widetilde{\boldsymbol{y}}_i - f_i(\boldsymbol{x}))^T \, \boldsymbol{\Sigma}_i^{-1} \, (\widetilde{\boldsymbol{y}}_i - f_i(\boldsymbol{x})). \end{aligned} \tag{2.9}$$

The relations f_i are referred to as the *functional model*, and the covariances $\boldsymbol{\Sigma}_i = \mathbb{V}(\boldsymbol{\epsilon}_i)$ of the zero-mean errors $\boldsymbol{\epsilon}_i$ as the *stochastic model*.

Based on the above abstract problem description, we now present the specific functional and stochastic models for trajectory estimation using GNSS, IMU and LiDAR data. In this estimation, all parameters are jointly estimated: the platform position and orientation, IMU biases and scale factors, GNSS antenna lever-arms, LiDAR mounting parameters, and object parameters that model the physical environment.

2.2. Coordinate systems and transformations

Following Groves (2013), a coordinate system α refers to coordinates $\boldsymbol{x}^\alpha = (\boldsymbol{B}^\alpha)^T \boldsymbol{x}$ of a vector $\boldsymbol{x} \in \mathbb{R}^3$ w.r.t. the orthonormal, right-handed basis $\boldsymbol{B}^\alpha = \left(b_1^\alpha, b_2^\alpha, b_3^\alpha\right) \in \text{SO}(3)$ of a corresponding three-dimensional Cartesian frame with origin $\boldsymbol{o}^\alpha \in \mathbb{R}^3$, where $\mathbb{R}^3$ is itself

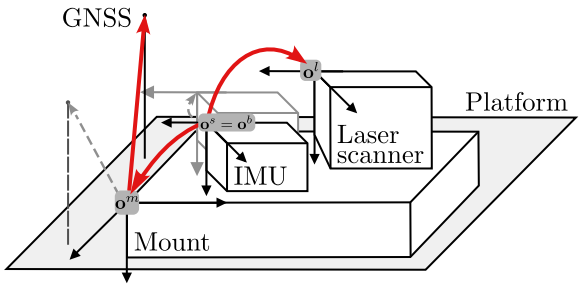


Fig. 2. The coordinate systems of the sensor platform components. Here, the body coordinate system is identical to the IMU coordinate system. This simplification is not necessary, but any transformation between b- and s-system is required to be rigid.

interpreted as Cartesian frame with canonical basis and origin $o = 0$. The frames α, β and γ are related by

$\boldsymbol{x}^{\gamma}_{\beta\alpha} := (\boldsymbol{B}^{\gamma})^T(\boldsymbol{o}^{\alpha} - \boldsymbol{o}^{\beta})$ the position of frame or point α w.r.t. frame β, resolved in frame γ, and

$\boldsymbol{R}^{\beta}_{\alpha} := \boldsymbol{B}^{\beta}(\boldsymbol{B}^{\alpha})^T$ the rotation from α-frame to β-frame.

The coordinate systems used are:

- i … earth-centered inertial coordinate system,
- e … earth-centered, earth-fixed coordinate system,
- m … mount coordinate system,
- b … body coordinate system,
- s … inertial sensor coordinate system,
- l … laser scanner coordinate system.

The body coordinate system is in our case in reference to the laser scanner system *body*, not the platform itself (see Fig. 2). A mount coordinate system is explicitly introduced here, rigid w.r.t. the platform, in order to allow modelling a gimbal or gyro stabilization mount. When the laser scanner system (including the IMU) is mounted on a stabilization mount, the platform's rotating motion is partially compensated for, but the m-system's orientation changes w.r.t. the b-, s- and l- systems and as a consequence the GNSS antenna is subject to dynamic motion relative to the body coordinate system. Note that the vector from body origin to mount origin $\boldsymbol{x}^{b}_{bm}$ is constant, as the origin of the m-system is exactly the centre of rotation for $\boldsymbol{R}^{b}_{m}$.

For ease of presentation, we use only one earth-fixed system. In practical application, this system would be replaced by a local Cartesian coordinate system to keep the coordinate values reasonably small. Unless otherwise specified, vectors $\boldsymbol{x}$ without a superscript will refer to geo-referenced coordinates in the e-system.

2.3. Trajectory model

For the purpose of processing and georeferencing the imaging sensor measurements, the platform trajectory is required at least for the time of each LiDAR measurement or camera exposure. Since trajectory data is commonly available as timestamped discrete samples, it is accordingly interpolated to obtain position and orientation for all measurements. Here, instead of such a discrete representation, the trajectory of the sensor platform is modelled using standard Euclidean B-splines (de Boor, 1980) in $\mathbb{R}^3$ for position, and rotation B-splines in $\mathrm{SO}(3) \subseteq \mathbb{R}^{3\times3}$ for orientation (Kim et al., 1995).

The use of rotation B-splines instead of, e.g., Euler-angle B-splines is advantageous as they are gimbal-lock- and singularity-free (cf. Haarbach et al., 2018), and invariant to coordinate system changes (Sommer et al., 2016). For a time interval $[t^x_1, t^x_{n_x}]$ containing n_x nodes ordered by time $t^x_1 < t^x_2 < \cdots < t^x_{n_x}$, the Euclidean position B-spline $\boldsymbol{x}^e_{eb}(t)$ of degree k_x is defined by

$$\begin{aligned} \boldsymbol{x}^e_{eb}(t) &: [t^x_1, t^x_{n_x}] \to \mathbb{R}^3 \\ t &\mapsto \sum_{i=1}^{n_x+k_x-1} \boldsymbol{x}_i B^x_i(t), \end{aligned} \tag{2.10}$$

with coefficients $\boldsymbol{x}_i \in \mathbb{R}^3$ and basis functions B^x_i, $1 \le i \le n_x + k_x - 1$. Similarly, for a time interval $[t^R_1, t^R_{n_R}]$ containing n_R nodes with $t^R_1 < t^R_2 < \cdots < t^R_{n_R}$, the rotation B-spline $\boldsymbol{R}^b_e$ of degree k_R is defined by

$$\begin{aligned} \boldsymbol{R}^b_e(t) &: [t^R_1, t^R_{n_R}] \to \mathrm{SO}(3) \\ t &\mapsto \boldsymbol{R}_0 \prod_{i=2}^{n_R+k_R-1} \exp(\log(\boldsymbol{R}^T_{i-1}\boldsymbol{R}_i)\tilde{B}^R_i(t)), \end{aligned} \tag{2.11}$$

with coefficients $\boldsymbol{R}_i \in \mathrm{SO}(3)$, $1 \le i \le n_R + k_R - 1$. The *cumulative basis functions* $\tilde{B}^R_i$ are obtained by summing up each basis function B^R_i and all following basis functions $(B^R_j)_{j>i}$

$$\tilde{B}^R_i(t) := \sum_{j=i}^{n_R+k_R-1} B^R_j(t), \quad 1 \le i \le n_R + k_R - 1. \tag{2.12}$$

The first cumulative basis function does not appear in Eq. (2.11), because the B-splines form a partition of unity on $[t^R_1, t^R_{n_R}]$ and therefore $\tilde{B}^R_1 = \sum_{i=1}^{n_R+k_R-1} B^R_i \equiv 1$.

The functions exp and log in Eq. (2.11) are the exponential and logarithmic maps between $\mathbb{R}^3$ and SO(3). The exponential $\exp : \mathbb{R}^3 \to \mathrm{SO}(3) \subset \mathbb{R}^{3\times3}$ maps a vector $\boldsymbol{\omega}$ to the corresponding rotation matrix, which represents a rotation of angle $\|\boldsymbol{\omega}\|$ around the axis $\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$. It is a composition of the hat operator $[\,\cdot\,]_\wedge : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$

$$[\boldsymbol{\omega}]_\wedge = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}. \tag{2.13}$$

and the matrix exponential. The inverse of the hat operator $[\,\cdot\,]_\wedge$ is denoted by $[\,\cdot\,]_\vee : \left[\mathbb{R}^3\right]_\wedge \subseteq \mathbb{R}^{3\times3} \to \mathbb{R}^3$. While the above describes abstract rotation splines in SO(3), in practice unit quaternions are used to compactly and efficiently represent the rotations $\boldsymbol{R}_i$.

The locality of classical B-splines carries over to the cumulative formulation: when B-splines of degree k are chosen as basis functions B^x_i and B^R_i, all but $k+1$ terms in the sum and product above vanish. The B-splines provide a continuous-time representation of the trajectory, which may be evaluated at any measurement time. For a given spline degree k, the spline trajectory is $k-1$ times continuously differentiable. Derivatives of position and orientation are required to compute the inertial measurement predictions. Thus, the spline degree must be high enough to allow computing the required derivatives. Velocity and acceleration are obtained simply by linear combination of the derivatives of the respective basis functions

$$\begin{aligned} \dot{\boldsymbol{x}}^e_{eb}(t) &= \sum_{i=1}^{n_x+k_x-1} \boldsymbol{x}_i \dot{B}^x_i(t), \\ \ddot{\boldsymbol{x}}^e_{eb}(t) &= \sum_{i=1}^{n_x+k_x-1} \boldsymbol{x}_i \ddot{B}^x_i(t). \end{aligned} \tag{2.14}$$

For computing the B-spline basis functions B^x_i, B^R_i and their derivatives, the standard recursively defined Cox-de-Boor formulas (de Boor, 1980) are used. In the case of uniformly spaced spline nodes, more efficient matrix-based formulas exist, but we do not make this assumption. For the orientation splines, recursive algorithms for the time-derivatives are given in Sommer et al. (2020), enabling the efficient computation of angular velocity and angular acceleration

$$\boldsymbol{\omega}^b_{eb}(t) = \left[\boldsymbol{R}^b_e(t)\dot{\boldsymbol{R}}^e_b(t)\right]_\vee, \tag{2.15}$$

$$\begin{aligned} \dot{\boldsymbol{\omega}}^b_{eb}(t) &= \left[\dot{\boldsymbol{R}}^b_e(t)\dot{\boldsymbol{R}}^e_b(t) + \boldsymbol{R}^b_e(t)\ddot{\boldsymbol{R}}^e_b(t)\right]_\vee \\ &= \left[-\left[\boldsymbol{\omega}^b_{eb}(t)\right]^2_\wedge + \boldsymbol{R}^b_e(t)\ddot{\boldsymbol{R}}^e_b(t)\right]_\vee. \end{aligned} \tag{2.16}$$

The spline-based trajectory theoretically allows for a representation with a reduced number of parameters compared to a discrete state representation as it automatically provides interpolation. However, for the purpose of LiDAR georeferencing, it is imperative to retain the full frequency spectrum of inertial measurements. Even small but high frequency vibrations, which are not represented properly, will lead to high frequency (*wave-like*) errors in the georeferenced point cloud. For this reason, we place spline nodes at the IMU sample times, in order to resolve trajectory dynamics at the full bandwidth provided by the IMU. For the results presented in Section 3, the spline degree is fixed to $k_x = 3$ and $k_R = 2$, i.e., cubic splines for position and quadratic splines for orientation. This is, respectively, the minimum degree necessary for a continuous linear acceleration and angular velocity. Since spline nodes are already placed at all measurement times of the IMU (the navigation sensor with the most frequent observations), we expect no benefit in using higher spline degrees. Even with less frequent spline nodes, the simulation analysis performed in Johnson et al. (2024) indicates the use of higher spline degrees does not generally improve trajectory accuracy but does increase time required for solving the equation systems.

Given the spline coefficients for position and orientation, together with calibration and object parameters, model predictions can be derived for all measurements: GNSS, IMU, LiDAR. The differences between model predictions and actual measurements are minimized in a least-squares optimization to obtain a set of best-fitting parameters.

2.4. Measurement models: Overview & priors

Due to the assumption of Gaussian errors with no correlation between different measurement types, the MAP problem reduces to a non-linear least-squares (NLS) problem. The general NLS formulation Eq. (2.7) is split up into the respective contributions to the overall SSR, which is made up of prior assumptions, GNSS position observations, IMU accelerometer and gyroscope observations, and LiDAR plane observations

$$\begin{aligned}\boldsymbol{x}^* \overset{(2.9)}{=} \underset{\boldsymbol{x}}{\operatorname{argmin}}\ & \mathrm{SSR}_{\mathrm{Priors}}(\boldsymbol{x}) \\ & + \mathrm{SSR}_{\mathrm{GNSS}}(\boldsymbol{x}) \\ & + \mathrm{SSR}_{\mathrm{IMU}}(\boldsymbol{x}) \\ & + \mathrm{SSR}_{\mathrm{LiDAR}}(\boldsymbol{x}).\end{aligned} \tag{2.17}$$

Each residual is the discrepancy between functional model and measurement, weighted according to the stochastic model. Note that for each residual, only a subset of all parameters (see Table 1) is relevant. In the following subsections, we provide functional and stochastic models for the four types of residuals.

Following the Bayesian approach, existing *prior* knowledge of the stochastic characteristics of a parameter may be introduced into the estimation. This is not strictly done for all parameters; some parameters are introduced without prior (or, rather, with a prior that is constant over the parameter's range). The introduction of priors allows accounting for, e.g., lever arms previously measured with known accuracy, or time-varying sensor errors with known stochastic properties.

For *time-constant parameters* $\boldsymbol{x}$, fictional observations may be introduced observing $\boldsymbol{x}$ with a certain value $\widetilde{\boldsymbol{x}}$, and an error $\boldsymbol{\epsilon}$ following a Gaussian distribution with (known) covariance matrix $\boldsymbol{\Sigma}$

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}). \tag{2.18}$$

Thus, for each constant parameter with associated prior, a fictional observation with functional model $f(\boldsymbol{x}) = \boldsymbol{x}$ and stochastic model $\boldsymbol{\Sigma}$ is added as part of Eq. (2.17).

For *time-varying parameters*, the fictional observations for parameters are derived by assuming the parameter follows a certain stochastic process model (see also Cucci et al., 2017a). We consider only first-order processes of the form

$$\dot{\boldsymbol{x}}(t) = -\rho \boldsymbol{x}(t) + \boldsymbol{w}(t), \tag{2.19}$$

where $\rho \in [0, \infty)$ is the reciprocal of the process correlation time, and $\boldsymbol{w}(t)$ is a white noise process with known power spectral density $\boldsymbol{Q}_w = \mathrm{diag}(Q_1, \dots)$. This may be written in discrete form as

$$\begin{aligned}\boldsymbol{x}(t_{i+1}) &= \beta \boldsymbol{x}(t_i) + \boldsymbol{w}_i, \\ \text{where } \beta &:= e^{-\rho(t_{i+1}-t_i)}, \\ \boldsymbol{w}_i &\sim N\left(\boldsymbol{0}, \boldsymbol{\Sigma}_i\right), \text{ and}\end{aligned} \tag{2.20}$$

$$\boldsymbol{\Sigma}_i := \begin{cases} \boldsymbol{Q}_w (t_{i+1} - t_i), & \rho = 0, \\ \dfrac{\boldsymbol{Q}_w}{2\rho}\left(1 - e^{-2\rho(t_{i+1}-t_i)}\right), & \rho > 0. \end{cases}$$

For $\rho = 0$, this is a random walk process, for $\rho \in (0, \infty)$ a stationary first-order Gauss–Markov process. The random processes are modelled here in their discrete form and represented as linear splines $\boldsymbol{x}(t)$, with fictional observations

$$\boldsymbol{0} = \beta \boldsymbol{x}(t_i) - \boldsymbol{x}(t_{i+1}) + \boldsymbol{w}_i, \tag{2.21}$$

added for every spline node t_i. Note that for these priors, the functional model also encodes a stochastic assumption, namely the correlation time of the process. Conversely, the stochastic model $\boldsymbol{\Sigma}_i = \mathbb{V}(\boldsymbol{w}_i)$ depends on the choice of spline nodes t_i of the stochastic process model (cf. Eq. (2.20)). Due to the Markovian property of the processes considered here, each term of the prior only depends on epoch $i+1$ and epoch i. Other types of Gaussian processes may be considered for modelling certain errors, but possibly destroy the sparsity of the equation system (cf. Dong et al., 2018). As an extension to the first-order Gauss–Markov process model, one may consider higher-order Gauss–Markov processes or specifically constrain higher-order derivatives of certain parameters. The latter is often used to stochastically model the motion of the platform if no IMU data is available, e.g., by assuming a zero-acceleration or zero-jerk motion model. Inclusion of such motion priors is analogous to the first-order Gauss–Markov model described above. However, in contrast to a Kalman filter, where inertial measurements are interpreted as an input to the motion model (Farrell et al., 2022), our formulation does not necessarily need motion priors in any form.

The above refers to *priors* in the stochastic sense. Often, prior information about parameters is also encoded deterministically, through the choice of parametrization of a given quantity. Specifically, any time-continuous quantity needs to be parametrized discretely. The decision how to perform this discretization is a sort of *functional* prior. In a classical Kalman filter, parameters may have different values at any given time as their estimate is updated and becomes more accurate over time as additional measurements arrive, even if the underlying physical quantity is constant. Here, we model all parameters as splines of arbitrary degree with arbitrary nodes, which allows both for discretization at any epoch (linear spline with nodes at all epochs) and fully constant (constant spline with only one node). But more importantly, it allows reducing the parameter count of the estimation by suitably choosing spline degree and spline node based on prior knowledge of the relevant physical quantities and measurement processes and considerations with respect to their estimability.

2.5. GNSS measurement model

The GNSS measurement model uses pre-processed GNSS positions, i.e., a loose coupling of GNSS. The modelled position of the GNSS antenna may be derived from the platform position $\boldsymbol{x}^e_{eb}(t)$, the platform orientation $\boldsymbol{R}^b_e(t)$, and the GNSS antenna lever arm $\boldsymbol{x}^m_{ma}$ together with the mount orientation $\boldsymbol{R}^b_m(t)$. In case multiple GNSS antennas/receivers are used, the GNSS position measurements of the different receivers may not arrive simultaneously, depending on the internal receiver clock sampling time and how receiver time offsets are applied to the output measurement timestamps. Apart from possibly different timing and the respective antenna lever arms, the measurement equation itself remains the same, regardless of whether one, two, or even more antennas are

used. At time t, the modelled position of the GNSS antenna is derived from platform trajectory, antenna lever arm, and mount orientation by

$$\boldsymbol{p}(t) = \boldsymbol{x}^e_{eb}(t) + \boldsymbol{R}^e_b(t)\,(\boldsymbol{x}^b_{bm} + \boldsymbol{R}^b_m(t)\boldsymbol{x}^m_{ma}). \tag{2.22}$$

The rotation $\boldsymbol{R}^b_m(t)$ from the stabilization mount (m-system) to the body frame (b-system) is considered known; practically, it is obtained from angular encoders in the stabilization mount (or gimbal), which are assumed to provide measurements of high enough frequency and accuracy that they may be used directly, possibly with linear interpolation, but without needing to explicitly model interpolation errors or measurement noise. Note also that the accuracy requirements for this orientation are lower than for the trajectory itself, since it is only used to correctly reference the GNSS antenna position and the GNSS lever arm is generally orders of magnitude smaller than the laser ranging distances. Thus, for a GNSS position measurement $\widetilde{\boldsymbol{p}}$ at time t, the observation equation is simply

$$\underbrace{\widetilde{\boldsymbol{p}}}_{\text{measurement}} = \underbrace{\boldsymbol{p}(t)}_{\text{model}} + \underbrace{\boldsymbol{\epsilon}_p}_{\text{error}}. \tag{2.23}$$

The position measurement errors $\boldsymbol{\epsilon}_p$ are assumed zero-mean and normally distributed with covariance $\boldsymbol{\Sigma}_p$. Standard GNSS processing provides an estimate of $\boldsymbol{\Sigma}_p$ along with each position measurement. Measurements of different epochs are assumed uncorrelated, but this assumption is not always appropriate. In such a case, the error model should be adapted to account for time-correlation (see also Crespillo et al., 2020), e.g., by including an additional error term modelled as a Gauss–Markov process. In our formulation, the error term in Eq. (2.23) is in this case split up into the explicitly modelled time-correlated bias and uncorrelated additive white noise (Pöppl et al., 2023c)

$$\underbrace{\boldsymbol{\epsilon}_p}_{\text{error}} := \underbrace{\boldsymbol{b}_p}_{\substack{\text{time-correlated error}\\ (=\text{ 'bias'})}} + \underbrace{\boldsymbol{w}_p}_{\substack{\text{uncorrelated error}\\ (=\text{ white noise})}}. \tag{2.24}$$

Conceptually, both the time-correlated bias and the uncorrelated white noise are stochastic assumptions. However, the time-correlated bias $\boldsymbol{b}_p$ is modelled explicitly as described in Section 2.4 and its stochastic properties are considered as part of the prior observations (Eq. (2.21)). Practically, $\boldsymbol{b}_p$ becomes part of the functional relation for the GNSS measurement model

$$\underbrace{\widetilde{\boldsymbol{p}}}_{\text{measurement}} = \underbrace{\boldsymbol{p}(t) + \boldsymbol{b}_p(t)}_{\text{model}} + \underbrace{\boldsymbol{w}_p}_{\text{uncorrelated error}}, \tag{2.25}$$

and $\boldsymbol{\Sigma}_p = \mathbb{V}(\boldsymbol{w}_p)$ is used for weighting the residuals. Whether modelling of the time-correlation is necessary may be decided either based on expectations regarding the quality of the GNSS measurements (environment, satellite visibility, etc.) and the specifics of the GNSS processing algorithm, or by statistical analysis of the GNSS residuals. Alternatively, or additionally, one may use robust loss functions such as the Huber loss (cf. Crespillo et al., 2018) to account for outliers in the GNSS measurements, which consequently also reduces the impact of large time-correlated errors. Most GNSS processing software also outputs velocity estimates. Because the GNSS velocity estimates are computed as part of the internal GNSS processing's Kalman filter, they will be highly correlated with the GNSS positions. Therefore, while velocity measurements could be integrated in the same way as position measurements, this is not done here.

2.6. Inertial measurement model

The measurement model uses standard strap-down inertial navigation equations. However, in this approach the inertial measurements are not numerically integrated to obtain position and orientation; rather the measurement model is derived directly from the spline trajectory (i.e., forward model). This allows treating the IMU measurements in a statistically appropriate way, and does not require propagating the measurement uncertainties as in the Kalman filter prediction step.

The IMU, made up of an accelerometer and a gyroscope, provides measurements $\widetilde{\boldsymbol{f}}$ and $\widetilde{\boldsymbol{\omega}}$ of specific force $\boldsymbol{f}$ and angular velocity $\boldsymbol{\omega}$. Specific force and angular velocity depend on platform position and orientation, and may be computed using the strap-down inertial navigation equations (cf. Groves, 2013). For an ideal IMU, the measured angular velocity and specific force is calculated from platform position and orientation as

$$\begin{aligned}\boldsymbol{\omega}^b_{ib}(t) &= \boldsymbol{\omega}^b_{ie}(t) + \boldsymbol{\omega}^b_{eb}(t)\\ &= \boldsymbol{R}^b_e(t)\,\boldsymbol{\omega}^e_{ie} + \left[\boldsymbol{R}^b_e(t)\dot{\boldsymbol{R}}^e_b(t)\right]_\vee,\end{aligned} \tag{2.26}$$

$$\begin{aligned}\boldsymbol{f}^b_{ib}(t) &= \boldsymbol{a}^b_{ib}(t) - \boldsymbol{\gamma}^b_{ib}(t)\\ &= \boldsymbol{R}^b_i(t)\ddot{\boldsymbol{x}}^i_{ib}(t) - \boldsymbol{R}^b_e(t)\boldsymbol{\gamma}^e_{ib}(t)\\ &= \boldsymbol{R}^b_e(t)\Big(\left[\boldsymbol{\omega}^e_{ie}\right]^2_\wedge \boldsymbol{x}^e_{eb}(t) + 2\left[\boldsymbol{\omega}^e_{ie}\right]_\wedge \dot{\boldsymbol{x}}^e_{eb}(t)\\ &\qquad + \ddot{\boldsymbol{x}}^e_{eb}(t) - \boldsymbol{\gamma}^e_{ib}(t)\Big),\end{aligned} \tag{2.27}$$

where $\boldsymbol{\omega}^e_{ie}$ is the angular velocity of the e-frame with respect to the inertial frame (i.e., the Earth's rotation), and $\boldsymbol{\gamma}^e_{ib}$ is the acceleration due to gravitational force.

The measurements of angular velocity and acceleration are with respect to the inertial sensor frame, the s-frame. In most applications, the trajectory will be referenced to the axes and centre of the IMU, in which case the b-frame and s-frame coincide and the above equations are used directly. In case multiple IMUs are used, this is of course not possible and measurements for at least one of the IMUs have to be transformed from the b-frame, to which the trajectory is referenced, to the inertial sensor's s-frame. This is easily done since b- and s-frame are related by a constant rotation and translation; angular velocity and acceleration are then given by

$$\boldsymbol{\omega}^s_{is} = \boldsymbol{R}^s_b\boldsymbol{\omega}^b_{is} = \boldsymbol{R}^s_b\big(\boldsymbol{\omega}^b_{ib} + \underbrace{\boldsymbol{\omega}^b_{bs}}_{=0}\big) = \boldsymbol{R}^s_b\boldsymbol{\omega}^b_{ib}, \tag{2.28}$$

$$\boldsymbol{f}^s_{is} = \boldsymbol{R}^s_e\left(\left[\boldsymbol{\omega}^e_{ie}\right]^2_\wedge \boldsymbol{x}^e_{es} + 2\left[\boldsymbol{\omega}^e_{ie}\right]_\wedge \dot{\boldsymbol{x}}^e_{es} + \ddot{\boldsymbol{x}}^e_{es} - \boldsymbol{\gamma}^e_{is}\right), \tag{2.29}$$

$$\begin{aligned}\text{with}\quad \boldsymbol{x}^e_{es} &= \boldsymbol{x}^e_{eb} + \boldsymbol{R}^e_b\boldsymbol{x}^b_{bs},\\ \dot{\boldsymbol{x}}^e_{es} &= \dot{\boldsymbol{x}}^e_{eb} + \dot{\boldsymbol{R}}^e_b\boldsymbol{x}^b_{bs},\\ \ddot{\boldsymbol{x}}^e_{es} &= \ddot{\boldsymbol{x}}^e_{eb} + \ddot{\boldsymbol{R}}^e_b\boldsymbol{x}^b_{bs}.\end{aligned} \tag{2.30}$$

Most IMUs do not output measurements of angular velocity and specific force at certain time instants (i.e., *instantaneous* measurements), but rather output averaged or integrated increments (i.e., *delta* measurements). For higher-end IMUs, these integrated measurements are typically internally corrected for coning and sculling errors (cf. Groves, 2013; Al-Jlailaty and Mansour, 2021). In this case, or if using pre-integrated IMU measurements, equations Eq. (2.26) and Eq. (2.27) are replaced with the corresponding integrated *delta-υ* and *delta-θ* measurements. The exact method with which the integrated measurements are computed is often not known to the user of the IMU, but we assume standard coning and sculling corrections (Savage, 1998a,b) are applied. For a measurement at time t_i, integrated from t_{i-1} to $t_i = t_{i-1} + \Delta t$, the corresponding integrated measurement equations are given by

$$\Delta\boldsymbol{\theta}^s_{is}(t_i) = \log\Big(\boldsymbol{R}^s_e(t_{i-1})\exp\left(\boldsymbol{\omega}^e_{ie}\Delta t_i\right)\boldsymbol{R}^e_s(t_i)\Big), \tag{2.31}$$

$$\begin{aligned}\Delta\boldsymbol{v}^s_{is}(t_i) = \quad & \boldsymbol{R}^s_e(t_{i-1})\Big(\left(\dot{\boldsymbol{x}}^e_{es}(t_i) - \dot{\boldsymbol{x}}^e_{es}(t_{i-1})\right)\\ &+ \Delta t_i\Big(\left[\boldsymbol{\omega}^e_{ie}\right]^2_\wedge \boldsymbol{x}^e_{es}(t_{i-1})\\ &+ 2\left[\boldsymbol{\omega}^e_{ie}\right]_\wedge \dot{\boldsymbol{x}}^e_{es}(t_{i-1})\\ &- \boldsymbol{\gamma}^e_{is}(t_{i-1})\Big)\Big).\end{aligned} \tag{2.32}$$

Eqs. (2.31)–(2.32) are still *ideal* measurements without errors, albeit in integrated form. In practice, the measurements are disturbed by a number of stochastic and deterministic errors (Groves, 2013; Farrell

et al., 2022). We use a standard error model, where the measured values are tainted by additive white noise $\epsilon_\star$, biases $\boldsymbol{b}_\star$ and scale factors $\boldsymbol{S}_\star = \mathrm{diag}(s_{x\star}, s_{y\star}, s_{z\star})$. The measurement model for specific force and angular velocity at a time t is thus given by

$$\underbrace{\begin{matrix}\widetilde{\Delta \boldsymbol{v}}^s_{is}\,\frac{1}{\Delta t} \\ \widetilde{\Delta \boldsymbol{\theta}}^s_{is}\,\frac{1}{\Delta t}\end{matrix}}_{\text{measurement}} = \underbrace{\begin{matrix}(\boldsymbol{I}+\boldsymbol{S}_v)\,\Delta \boldsymbol{v}^s_{is}(t)\,\frac{1}{\Delta t} + \boldsymbol{b}_v(t) \\ (\boldsymbol{I}+\boldsymbol{S}_\theta)\,\Delta \boldsymbol{\theta}^s_{is}(t)\,\frac{1}{\Delta t} + \boldsymbol{b}_\theta(t)\end{matrix}}_{\text{model}} + \underbrace{\begin{matrix}\boldsymbol{\epsilon}_v \\ \boldsymbol{\epsilon}_\theta\end{matrix}}_{\text{noise}} . \tag{2.33}$$

The delta measurements are scaled by their time interval, so that the magnitude of bias and noise is the same as when using instantaneous measurements. The biases and scale factors cause errors in the trajectory which compound in time, but may be calibrated in-run if sufficiently redundant information is available from GNSS or LiDAR. Biases and scale factors are made up of a time-constant and a time-varying component. The time-varying components are modelled as stochastic processes with known characteristics (Farrell et al., 2022). While it is possible to experimentally derive the accelerometer and gyroscope bias' process parameters from statically acquired data via Allan Variance Analysis (AVA), such a procedure is not readily available for the scale factors errors. Additionally, scale factor errors are only determinable if sufficient input signal is present for the respective sensor. In situations with limited platform dynamics, this is not the case and the time-varying biases and scale factors are hard to distinguish. Thus, although scale factors can theoretically also vary in time, we find it beneficial to model them as constant within each data acquisition. The error processes of accelerometer and gyroscope biases are represented using linear splines, with spline nodes set at GNSS measurement times, and interpolated in between. Stochastically, the biases are modelled using first-order Gauss–Markov process priors, as described in Section 2.4. Note that albeit being a stochastic quantity, the time-varying biases become part of the functional model in Eq. (2.33). Their stochastic properties are considered through prior observations as in Eq. (2.21) and the noise covariances $\boldsymbol{\Sigma}_v = \mathbb{V}(\boldsymbol{\epsilon}_v)$ and $\boldsymbol{\Sigma}_\theta = \mathbb{V}(\boldsymbol{\epsilon}_\theta)$, derived from AVA, are used to weight the residuals.

2.7. LiDAR measurement model

The integration of LiDAR measurements of some form in a trajectory estimation may be done in many ways. In a loosely coupled integration, position and orientation may be derived from LiDAR odometry (e.g., Zhang and Singh, 2014), which is then integrated with the other sensor data. The approach presented here however employs tight coupling of LiDAR and IMU. Note that the laser scanners considered here are single-beam single-channel time-of-flight LiDAR systems commonly used in surveying applications. Thus, each LiDAR point possesses its own timestamp, and measurements are acquired with very high frequency (often >1 MHz). To obtain useful spatial information for the trajectory measurements, a number of individual points are aggregated and assigned a single timestamp. Such an aggregation of LiDAR points will be termed *feature*, with multiple data-derived *features* corresponding to one physical *object*. This object may be modelled explicitly, with the features forming observations of that object, in which case the modelled planes are analogous to tie points in bundle adjustment. Or, the feature-object correspondences may be used implicitly to form pairwise constraints between features corresponding to the same object, as is done in ICP-based approaches (iterative closest point, cf. Glira et al., 2015). As no pre-existing knowledge of the environment is assumed and no ground control required, the LiDAR measurements must be associated with other LiDAR measurements to form the observations or constraints, based on the assumption that the environment largely remains static during data acquisition.

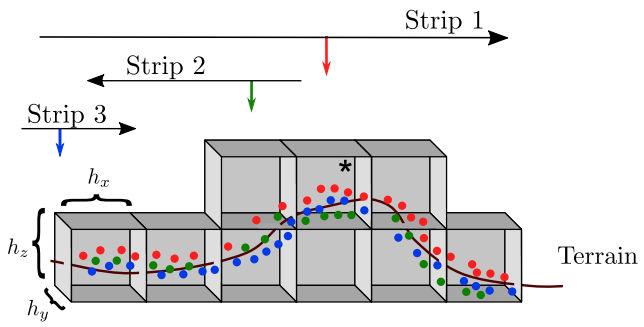


Fig. 3. Points are rasterized and clustered by time. In standard airborne acquisitions, time clusters usually contain points from a single flight strip, but generally there is no requirement for the acquisition pattern.

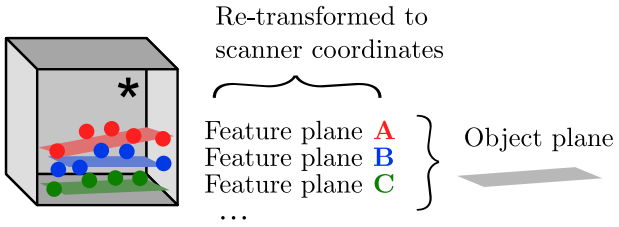


Fig. 4. For each cell and time-cluster, a PCA is computed. All feature planes belonging to a single cell are combined into a corresponding object plane.

2.7.1. Feature extraction and matching

Cheng et al. (2018) classify LiDAR registration techniques into point-based, line-based and surface-based methods. Here, we focus on the most basic type of surfaces, namely planes, which have proven versatile and effective in various scenarios, especially airborne acquisitions and have been widely used for a long time (Kager, 2004; Skaloud and Lichti, 2006).

We divide the processing into two parts, (1) feature extraction and (2) feature matching, and outline a simple and fast procedure for extracting and matching planar features from point clouds. The tasks of feature extraction and feature matching are independent of the general trajectory estimation problem. The feature extraction and matching procedure was presented previously in Pöppl et al. (2023c), and is based on spatial rasterization and temporal clustering of the LiDAR points together with standard principal component analysis (PCA). This approach is somewhat similar to the normal distributions transform (NDT, Biber and Strasser, 2003) often used for point cloud registration, a variant of which has also been used recently for hybrid bundle/strip adjustment (Jonassen et al., 2023). Instead of a purely data-driven stochastic model obtained from the 3D point covariances, our approach uses a stochastic model which augments the data-driven uncertainty estimates derived from the geometric properties of the plane fit with prior knowledge of the laser scanning measurement process.

For the *feature extraction*, all LiDAR points are placed into their respective raster cells of a rasterization with cell size (h_x, h_y, h_z). The cell size should be chosen according to (average) point density and expected size of the planar features. The points in each cell are greedily clustered together according to their acquisition time, so that points in each cluster are never further than $t_{\max}$ seconds apart. The maximum cluster duration $t_{\max}$ is set to approximately the time it takes the platform to move two cells further. For example, a velocity of 50 m/s and a cell size of 2 m result in $t_{\max} = 0.08$ s. Clustering is done separately for each laser scanner, so that points from different scanners are never aggregated into the same feature. This means that the data acquisition does not necessarily need to be split into separate strips beforehand, as the feature extraction relies only on inherent properties of the points themselves. As points are added to each cluster, the cluster's sample

covariance is incrementally built up, and a PCA is then computed to obtain a best-fit plane (see Fig. 3 and Fig. 4). Cells with a bad fit (i.e., smallest eigenvalue is above threshold, larger two eigenvalues are below given threshold) are discarded. A feature plane serves as measurement in the course of the least-squares adjustment, but is actually an aggregate of a number of individual LiDAR point measurements. To tie the feature plane to the trajectory, it needs to be associated with a timestamp. Here, each feature plane is assigned the interval midpoint of the timestamps $t_1, \dots, t_n$ of all its contributing points $\widetilde{\boldsymbol{x}}_1, \dots, \widetilde{\boldsymbol{x}}_n \in \mathbb{R}^3$

$$t := \frac{\max(t_1, \dots, t_n) + \min(t_1, \dots, t_n)}{2}. \tag{2.34}$$

The *feature matching* procedure is then very simple: All feature planes associated with a cell are combined into an object plane, which represents the actual physical object. This requires that the cell size is chosen so that it exceeds the georeferencing errors, as otherwise no correspondences can be established with this scheme. In extreme cases with exceptionally large initial georeferencing errors (>2 m), the feature extraction, matching, adjustment and georeferencing (cf. Fig. 1) may be repeated iteratively with successively smaller cell sizes.

In previous work (Pöppl et al., 2023d,e), we have also used a different extraction method based on hierarchically traversing an octree representation of the point cloud to perform robust plane-fit similar to Nurunnabi et al. (2015). While this is computationally more intensive, it has some advantages in mobile laser scanning applications, as it is capable of extracting both small-scale (e.g., curbs) and large-scale features (e.g., facades). As long as the model geometry (in our case: planar) remains the same, different feature extraction and matching algorithms and implementations may be employed depending on the given application requirements.

2.7.2. Plane measurement model

From now on, we simply assume that some procedure is available for planar feature extraction and matching. Specifically, we consider a feature plane made up of point measurements $\widetilde{\boldsymbol{x}}_1, \dots, \widetilde{\boldsymbol{x}}_n \in \mathbb{R}^3$. The matrix $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{n\times 3}$ is obtained by transposing and stacking all points $\widetilde{\boldsymbol{x}}_i$ from which the planar feature was derived. Then, the de-meaned data matrix $\boldsymbol{M}$ is computed by subtracting the data mean $\widetilde{\boldsymbol{\mu}}$ from $\widetilde{\boldsymbol{X}}$

$$\widetilde{\boldsymbol{\mu}} := \frac{1}{n}\sum_{i=1}^{n} \widetilde{\boldsymbol{x}}_i, \quad \boldsymbol{M} = \begin{pmatrix} \boldsymbol{m}_1^T \\ \vdots \\ \boldsymbol{m}_n^T \end{pmatrix} := \begin{pmatrix} \widetilde{\boldsymbol{x}}_1^T - \widetilde{\boldsymbol{\mu}}^T \\ \vdots \\ \widetilde{\boldsymbol{x}}_n^T - \widetilde{\boldsymbol{\mu}}^T \end{pmatrix} \tag{2.35}$$

The eigenvalue decomposition of the unscaled covariance matrix $\boldsymbol{C} = \boldsymbol{M}^T\boldsymbol{M}$ is given by $\boldsymbol{C} = \boldsymbol{V}\,\mathrm{diag}(\lambda_1, \lambda_2, \lambda_3)\boldsymbol{V}^T$, where the eigenvalues are assumed to be sorted in descending order, i.e., $\lambda_1 \geq \lambda_2 \geq \lambda_3$. The matrix $\boldsymbol{V}$ contains the eigenvectors of $\boldsymbol{C}$ in its columns. The feature plane's coordinate system is then defined by $\boldsymbol{V}$, specifically by plane axes $\widetilde{\boldsymbol{k}}_1, \widetilde{\boldsymbol{k}}_2$, and plane normal $\widetilde{\boldsymbol{n}}$, so that

$$\boldsymbol{V} = (\widetilde{\boldsymbol{k}}_1\ \widetilde{\boldsymbol{k}}_2\ \widetilde{\boldsymbol{n}}), \text{ and}$$
$$\boldsymbol{V}^T\boldsymbol{M}^T\boldsymbol{M}\,\boldsymbol{V} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}. \tag{2.36}$$

Note that while the above computations are carried out in the earth-referenced e-system, the centroid, axes, and normal are stored in laser scanner coordinates $\widetilde{\boldsymbol{\mu}}^l, \widetilde{\boldsymbol{k}}_1^l, \widetilde{\boldsymbol{k}}_2^l, \widetilde{\boldsymbol{n}}^l$ and transformed into the earth-referenced coordinate system using the georeferencing equations

$$\begin{aligned} \widetilde{\boldsymbol{\mu}} &:= \widetilde{\boldsymbol{\mu}}^e = \boldsymbol{x}_{eb}^e(t) + \boldsymbol{R}_b^e(t)(\boldsymbol{x}_{bl}^b + \boldsymbol{R}_l^b\widetilde{\boldsymbol{\mu}}^l) \\ \widetilde{\boldsymbol{k}}_1 &:= \widetilde{\boldsymbol{k}}_1^e = \boldsymbol{R}_b^e(t)\boldsymbol{R}_l^b\,\widetilde{\boldsymbol{k}}_1^l, \\ \widetilde{\boldsymbol{k}}_2 &:= \widetilde{\boldsymbol{k}}_2^e = \boldsymbol{R}_b^e(t)\boldsymbol{R}_l^b\,\widetilde{\boldsymbol{k}}_2^l, \\ \widetilde{\boldsymbol{n}} &:= \widetilde{\boldsymbol{n}}^e = \boldsymbol{R}_b^e(t)\boldsymbol{R}_l^b\,\widetilde{\boldsymbol{n}}^l, \end{aligned} \tag{2.37}$$

where $\boldsymbol{R}_l^b$ and $\boldsymbol{x}_{bl}^b$ are the laser scanner boresight and lever arm. Position $\boldsymbol{x}_{eb}^e(t)$ and orientation $\boldsymbol{R}_b^e(t) = (\boldsymbol{R}_e^b(t))^T$ are evaluated at the feature's associated time t.

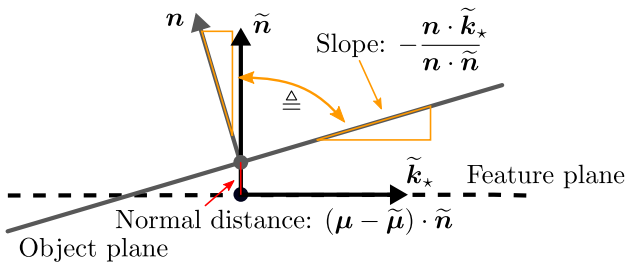


Fig. 5. Illustration of the feature-object plane observations.

An object plane is defined by centroid $\boldsymbol{\mu}$ and normal $\boldsymbol{n}$. The initial centroid $\boldsymbol{\mu}^0$, normal $\boldsymbol{n}^0$ and (orthonormal but otherwise arbitrary) axes $\boldsymbol{k}_1^0, \boldsymbol{k}_2^0$ are computed by combining the means and covariances of all corresponding feature planes. During the adjustment, the plane's position and orientation is optimized with three degrees of freedom, an offset o along the normal vector and slopes z_1, z_2 so that the effective centroid and (scaled) normal is given by

$$\boldsymbol{\mu} := \boldsymbol{\mu}^0 + o\boldsymbol{n}^0, \quad \boldsymbol{n} := \boldsymbol{n}^0 + z_1\boldsymbol{k}_1^0 + z_2\boldsymbol{k}_2^0. \tag{2.38}$$

The feature-object measurement equations are made up of normal distance between the planes and the slopes of object plane w.r.t. the feature plane axes (Fig. 5)

$$\tilde{d} = \underbrace{(\boldsymbol{\mu} - \widetilde{\boldsymbol{\mu}})\cdot\widetilde{\boldsymbol{n}}}_{\text{normal distance } d} + \epsilon_d, \tag{2.39}$$

$$\tilde{s}_1 = \underbrace{\frac{\boldsymbol{n}\cdot\widetilde{\boldsymbol{k}}_1}{\boldsymbol{n}\cdot\widetilde{\boldsymbol{n}}}}_{\text{slope } s_1} + \epsilon_{s_1}, \tag{2.40}$$

$$\tilde{s}_2 = \underbrace{\frac{\boldsymbol{n}\cdot\widetilde{\boldsymbol{k}}_2}{\boldsymbol{n}\cdot\widetilde{\boldsymbol{n}}}}_{\text{slope } s_2} + \epsilon_{s_2}. \tag{2.41}$$

Note that we do not model the errors of plane mean $\widetilde{\boldsymbol{\mu}}$ and normal $\widetilde{\boldsymbol{n}}$, but rather specify the measurement equations in each feature plane's coordinate systems. Instead of performing linearized error propagation, the stochastic model for the above errors is derived directly from the PCA results: Following Quinn and Ehlmann (2019), standard deviations for $\epsilon_d, \epsilon_{s_1}, \epsilon_{s_2}$ are computed by interpreting the PCA as an ordinary least squares (OLS) fit. In the coordinate system defined by $\boldsymbol{V}$, the normal distance $\tilde{d}$ and slopes $\tilde{s}_1, \tilde{s}_2$ correspond to the least-squares solution of

$$(\boldsymbol{m}_i\cdot\widetilde{\boldsymbol{k}}_1)s_1 + (\boldsymbol{m}_i\cdot\widetilde{\boldsymbol{k}}_2)s_2 + d = (\boldsymbol{m}_i\cdot\widetilde{\boldsymbol{n}}) + \epsilon_i, \quad 1 \leq i \leq n, \tag{2.42}$$

where $\mathbb{V}(\epsilon_i) := \sigma$. Defining $\boldsymbol{A} := (\boldsymbol{1}, \boldsymbol{M}\widetilde{\boldsymbol{k}}_1, \boldsymbol{M}\widetilde{\boldsymbol{k}}_2)$ and $\boldsymbol{b} := \boldsymbol{M}\widetilde{\boldsymbol{n}}$, the solution $\boldsymbol{x} := (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$ to the OLS problem is simply $\tilde{d} = \tilde{s}_1 = \tilde{s}_2 = 0$, because

$$\boldsymbol{A}^T\boldsymbol{b} = \begin{pmatrix} \sum_{i=1}^n \boldsymbol{m}_i^T\widetilde{\boldsymbol{n}} \\ \widetilde{\boldsymbol{k}}_1^T\boldsymbol{M}^T\boldsymbol{M}\widetilde{\boldsymbol{n}} \\ \widetilde{\boldsymbol{k}}_2^T\boldsymbol{M}^T\boldsymbol{M}\widetilde{\boldsymbol{n}} \end{pmatrix} \overset{(2.35)}{\underset{(2.36)}{=}} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{2.43}$$

Similarly, the sample covariance for $\tilde{d}, \tilde{s}_1, \tilde{s}_2$ may be stated in terms of eigenvalues of $\boldsymbol{C}$

$$\mathrm{diag}(\sigma_d^2, \sigma_{s_1}^2, \sigma_{s_2}^2) = \sigma^2(\boldsymbol{A}^T\boldsymbol{A})^{-1} \tag{2.44}$$

$$= \sigma^2\mathrm{diag}\left(\frac{1}{n}, \frac{1}{\lambda_1}, \frac{1}{\lambda_2}\right). \tag{2.45}$$

The standard OLS estimate s^2 of σ^2 is

$$\sigma^2 \mathrel{\widehat{=}} s^2 = \frac{\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2}{n-3} = \frac{\widetilde{\boldsymbol{n}}^T\boldsymbol{M}^T\boldsymbol{M}\,\widetilde{\boldsymbol{n}}}{n-3} \overset{(2.36)}{=} \frac{\lambda_3}{n-3}, \tag{2.46}$$

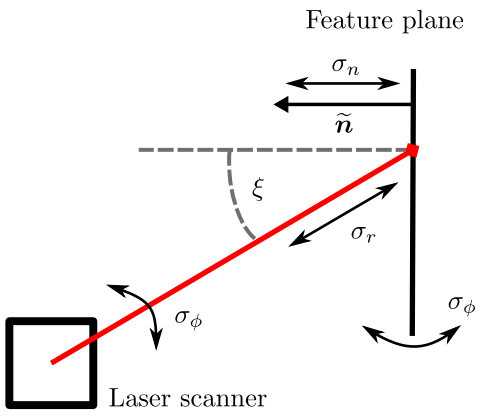


Fig. 6. Impact of ranging and angular accuracy on plane measurement accuracies.

from which the variances of the measurement errors in Eqs. (2.39)–(2.41) are derived as

$$\begin{aligned}\mathbb{V}(\epsilon_d) &= \sigma_d^2 = \frac{s^2}{n},\\ \mathbb{V}(\epsilon_{s_1}) &= \sigma_{s_1}^2 = \frac{s^2}{\lambda_1} = \sigma_d^2\frac{n}{\lambda_1},\\ \mathbb{V}(\epsilon_{s_2}) &= \sigma_{s_2}^2 = \frac{s^2}{\lambda_2} = \sigma_d^2\frac{n}{\lambda_2}.\end{aligned} \tag{2.47}$$

This is a purely geometric interpretation which assumes statistical independence of the individual points $\boldsymbol{x}_i$ and thus disregards any correlated or systematic measurement errors. Growing point densities, especially in mobile laser scanning, and high point measurement precision cause the uncertainty of feature plane measurements to become unrealistically small. As an example, a perfectly planar surface made up of 100 points measured with a precision of 5 mm results in a standard deviation of approximately $\sigma_d \approx 0.5$ mm. However, this is purely an uncertainty estimate based on local precision under the assumption of uncorrelated measurements, and thus does not take into account any effects which similarly impact all points of the plane.

2.7.3. Extended stochastic model

To account for (1) scanner ranging accuracy σ_r, angular accuracy σ_ϕ and (2) the effect of angle of incidence (AOI) ξ and beam footprint diameter f on the measurement error, we extend the geometrically derived stochastic model and replace Eq. (2.47) with

$$\begin{aligned}\mathbb{V}(\epsilon_d) &= \bar{\sigma}_d^2 := \sigma_d^2 + \sigma_n(\xi, r)^2,\\ \mathbb{V}(\epsilon_{s_1}) &= \bar{\sigma}_{s_1}^2 := \bar{\sigma}_d^2\frac{n}{\lambda_1} + \sigma_\phi^2,\\ \mathbb{V}(\epsilon_{s_2}) &= \bar{\sigma}_{s_2}^2 := \bar{\sigma}_d^2\frac{n}{\lambda_2} + \sigma_\phi^2.\end{aligned} \tag{2.48}$$

1. *Scanner ranging and angular accuracy*
   The laser scanner performs ranging and angular measurements, with an accuracy of σ_r and σ_ϕ, respectively. Making a small angle approximation and assuming angular uncertainty is isotropic, this may be interpreted as the accuracy of a point in a scanner-referenced coordinate system aligned to the laser beam direction $\Sigma_{\text{beam}} \mathrel{\widehat{=}} \text{diag}(\sigma_r^2, (r\sigma_\phi)^2, (r\sigma_\phi)^2)$. This covariance is transformed into the feature plane coordinate system, to obtain the standard deviation along the normal direction σ_n (see Fig. 6)

   $$\sigma_n(\xi, r) = \sqrt{\left(\sigma_r \cos(\xi)\right)^2 + \left(\sigma_\phi r \sin(\xi)\right)^2}. \tag{2.49}$$

   Additionally, the angular accuracy causes an uncertainty in the slopes of $\tan(\sigma_\phi) \approx \sigma_\phi$.
2. *Angle of incidence and beam footprint*
   At low range, the normal distance standard deviation is now predominantly determined by σ_r, even though the ranging measurement may be expected to significantly degrade for high angles of incidence ξ. In the extreme case of the laser beam being parallel to the plane, no sensible measurement is possible. Additionally, as ξ increases, a nominally circular laser footprint with diameter f becomes elliptical with increasingly large diameter $f_\xi = \frac{f}{\cos\xi}$. This may lead to differing measurements when the same plane is observed from different angles.
   Based on these two observations, we introduce empirical scale factors of the LiDAR ranging and angular accuracies. Assuming that the ranging accuracy depends on the ability to correctly time the return signal, the ranging accuracy deteriorates with increasing return pulse duration and thus with incidence angle (Pfeifer and Briese, 2007). Here, this effect is modelled by scaling the nominal ranging accuracy σ_r, assumed to be valid for orthogonal incidence, with the ratio of the return pulse duration w_ξ at incidence angle ξ to the nominal pulse duration w_0. Under a Gaussian beam assumption, the return pulse duration w_ξ may be approximated by convolving the signal Gaussian with the sloped-plane Gaussian, yielding

   $$w_\xi \approx \sqrt{{w_0}^2 + \left(\frac{f_\xi \sin(\xi)}{c}\right)^2}, \tag{2.50}$$

   where c is the speed of light. Furthermore, the power of the return signal for a Lambertian surface reduces with the cosine of the angle of incidence (Soudarissanane et al., 2009), leading to a corresponding deterioration in signal-to-noise ratio. This effect is considered by scaling the ranging accuracy with $1/\cos(\xi)$. The final standard deviation of the ranging measurements is thus given as a function of the incidence angle ξ and the nominal ranging accuracy σ_r

   $$\sigma_r(\xi) := \frac{1}{\cos(\xi)}\frac{w_\xi}{w_0}\sigma_r. \tag{2.51}$$

   Note that the adjusted accuracies correspond to the nominal accuracies at orthogonal incidence, but are asymptotically infinite for small angles of incidence (Fig. 7), as

   $$\begin{aligned}&\sigma_n(\xi, r) \geq \sigma_r, \quad \forall \xi \in [0, \pi/2],\\ &\lim_{\xi\to 0}\sigma_n(\xi, r) = \sigma_r(0) \quad \text{and}\\ &\lim_{\xi\to\pi/2}\sigma_n = \infty.\end{aligned} \tag{2.52}$$

   At smaller ranges, the ranging uncertainty σ_r dominates, but at larger ranges r, the angular accuracy σ_ϕ is responsible for most of the uncertainty in the normal direction.

It should be emphasized that these are not corrections, which would require knowledge of the exact measurement mechanisms and methods of echo detection, but rather an attempt to stochastically account for such effects in a generic but still model-based manner. Notably, the AOI-adapted ranging accuracy $\sigma_r(\xi)$ does not depend on measurement range, and it is assumed the nominal ranging accuracy σ_r is approximately valid throughout the operating range of the LiDAR system.

The complete LiDAR observation equation is obtained by plugging Eq. (2.37) and Eq. (2.38) into Eqs. (2.39)–(2.41). For a LiDAR object plane described by $\boldsymbol{\mu}^0, \boldsymbol{k}_1^0, \boldsymbol{k}_2^0, \boldsymbol{n}^0$ and modelled by (o, z_1, z_2), observed by a feature plane described by $\tilde{\boldsymbol{\mu}}^l, \tilde{\boldsymbol{k}}_1^l, \tilde{\boldsymbol{k}}_2^l, \tilde{\boldsymbol{n}}^l$ at time t, it is given by Eqs. (2.53)–(2.55) (see Box I). The observations are weighted according to the (adjusted) error variance $\boldsymbol{\Sigma}_l := \text{diag}(\bar{\sigma}_d^2, \bar{\sigma}_{s_1}^2, \bar{\sigma}_{s_2}^2)$. This stochastic model still relies on the assumptions that the associated feature planes actually represent the same object plane. As robust data association is a challenging problem (Cadena et al., 2016) and we cannot rule out invalid correspondences, we rely instead on robust statistics by replacing the squared loss function for the LiDAR measurements in Eq. (2.17) with the Huber loss function (Huber, 1964).

$$\underbrace{\tilde{d}}_{=0} = \underbrace{\left(\left(\boldsymbol{\mu}^0 + o\boldsymbol{n}^0\right) - \left(\boldsymbol{x}^e_{eb}(t) + \boldsymbol{R}^e_b(t)\left(\boldsymbol{x}^b_{bl} + \boldsymbol{R}^b_l\widetilde{\boldsymbol{\mu}}^l\right)\right)\right) \cdot \left(\boldsymbol{R}^e_b(t)\boldsymbol{R}^b_l\,\widetilde{\boldsymbol{n}}^l\right)}_{\text{normal distance}} + \epsilon_d \tag{2.53}$$

$$\underbrace{\tilde{s}_1}_{=0} = \underbrace{\frac{\left(\boldsymbol{n}^0 + z_1\boldsymbol{k}^0_1 + z_2\boldsymbol{k}^0_2\right) \cdot \left(\boldsymbol{R}^e_b(t)\boldsymbol{R}^b_l\,\widetilde{\boldsymbol{k}}^l_1\right)}{\left(\boldsymbol{n}^0 + z_1\boldsymbol{k}^0_1 + z_2\boldsymbol{k}^0_2\right) \cdot \left(\boldsymbol{R}^e_b(t)\boldsymbol{R}^b_l\,\widetilde{\boldsymbol{n}}^l\right)}}_{\text{slope for axis 1}} + \epsilon_{s_1} \tag{2.54}$$

$$\underbrace{\tilde{s}_2}_{=0} = \underbrace{\frac{\left(\boldsymbol{n}^0 + z_1\boldsymbol{k}^0_1 + z_2\boldsymbol{k}^0_2\right) \cdot \left(\boldsymbol{R}^e_b(t)\boldsymbol{R}^b_l\,\widetilde{\boldsymbol{k}}^l_2\right)}{\left(\boldsymbol{n}^0 + z_1\boldsymbol{k}^0_1 + z_2\boldsymbol{k}^0_2\right) \cdot \left(\boldsymbol{R}^e_b(t)\boldsymbol{R}^b_l\,\widetilde{\boldsymbol{n}}^l\right)}}_{\text{slope for axis 2}} + \epsilon_{s_1} \tag{2.55}$$

Box I.

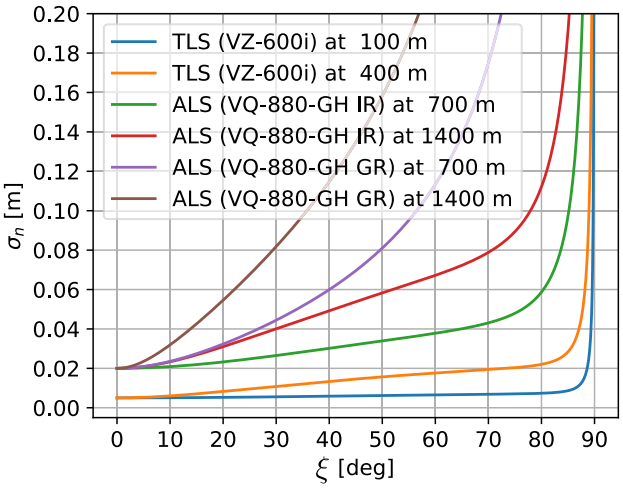


Fig. 7. Standard deviation in plane normal direction $\sigma_n(\xi, r)$ for the extended stochastic model Eq. (2.48). Shown for parameter combinations of different scanners, namely a terrestrial laser scanner (TLS, *RIEGL* VZ-600i) and for the infrared (IR) and green (GR) laser channels of the topo-bathymetric dual-wavelength airborne laser scanner used in Section 3 (ALS, *RIEGL* VQ-880-GH).

Control information

Point cloud data from other sources may be introduced into the adjustment as control information. In this case, planes are extracted from the control point cloud as described above, but the planar object parameters are not optimized and stay fixed to their initial value. Control planes derived by other means, e.g. by fitting of checkerboard targets, may be integrated in the same way by matching them to data-derived object planes. The use of control information is of course useful for referencing to a given datum defined by the ground control, but also for boresight calibration with challenging scan geometries (Pöppl et al., 2023e).

2.8. Solving the non-linear optimization

The trajectory and calibration parameter estimates are obtained by solving the non-linear least squares estimation posed by Eq. (2.9), i.e., minimizing the SSR consisting of prior observations (2.18), (2.21), GNSS observations (2.25), IMU observations (2.33) and LiDAR observations (2.53)–(2.55). Table 1 recaps the full set of parameters which occur in these equations. Note that this is a non-linear optimization, which requires suitable initial values and may not converge to the sought minimum otherwise. In previous work (Pöppl et al., 2023a,e), we used a Kalman filter for obtaining an initial trajectory estimate. Although the filter itself is computationally efficient, the initial trajectory has to be converted to the corresponding spline coefficients, as spline coefficients are not identical to the function values for spline degrees $k > 1$. To avoid this, we employ the GNSS/IMU adjustment as described above in a sliding-window fashion, where only parameters in a certain time window (here: 60 s) are optimized, and the trajectory is incrementally built up while the optimization window advances in time. Although slower in terms of runtime compared to the filter, this approach provides more suitable initial values for the batch adjustment as it re-uses the exact same observation models and implementation thereof.

The full optimization involves a very large number of parameters, as the trajectory is modelled at the frequency of the IMU measurements in order to preserve high frequency components. Thus, the resulting Jacobian and Hessian matrices occurring in the NLS optimization are also large, with up to multiple millions of rows and columns. However, both Jacobian and Hessian are sparse in nature, as all observation equations depend only on a small subset of parameters due to the local nature of the splines. As discussed in Cucci and Skaloud (2019), the GNSS/IMU-derived linear least-squares sub-problem is ill-conditioned for some combinations of IMU and GNSS measurement frequencies and noise characteristics, and iterative sparse solvers will converge only slowly if at all. This is especially the case when starting with a bad initial estimate for some or all parameters, and in the presence of large outliers, which cause correspondingly large entries in the Jacobian matrix. The latter part is partially mitigated by using a robust loss function (e.g., Huber loss), as is done for LiDAR and GNSS measurements. Though possibly ill-conditioned, these sparse problems can still be solved with sufficient precision using direct sparse methods, specifically Cholesky decomposition. The effort required for direct decomposition depends on the variable ordering. This may be done manually by time-sorting the trajectory variables to obtain a band-diagonal or arrow-shaped Hessian matrix, or automatically by using a suitable sorting algorithm. In our experience, the nested dissection (NESDIS) sorting in SuiteSparse's Cholesky factorization (CHOLMOD, Chen et al., 2008) performs best. The CERES solver (Agarwal et al., 2022) is used to solve the NLS optimization using the Levenberg–Marquardt algorithm, relying on CHOLMOD internally to solve the least-squares sub-problem.

3. Applications & results

To practically demonstrate the proposed trajectory estimation methodology, we apply the adjustment procedure to a real-world laser scanning dataset. This dataset consists of two airborne data acquisitions, one with a crewed fixed-wing aircraft (airborne laser scanning, ALS) and one with an uncrewed multicopter (uncrewed laser scanning, ULS). The two datasets, obtained from different platforms, are co-registered in a joint adjustment of all available data from both

Table 1
Parameters occurring in the trajectory estimation. Not all calibration parameters are always estimated; in case some parameters are precisely known (e.g., lever arms previously measured using geodetic techniques), in which case they are held constant in the adjustment. In case no stabilization mount is used, $\boldsymbol{x}^b_{bm} = \boldsymbol{0}$, $\boldsymbol{R}^m_b = \boldsymbol{I}_{3\times3}$ and $\boldsymbol{x}^m_{ma} = \boldsymbol{x}^b_{ba}$.

Trajectory	Position Orientation	$\boldsymbol{x}^e_{eb}$ $\boldsymbol{R}^b_e$
Mount	Origin	$\boldsymbol{x}^b_{bm}$
GNSS	Lever arm Position bias	$\boldsymbol{x}^m_{ma}$ $\boldsymbol{b}_p$
IMU	Accelerometer bias Accelerometer scale factor Gyroscope bias Gyroscope scale factor	$\boldsymbol{b}_v$ $\boldsymbol{s}_v$ $\boldsymbol{b}_\theta$ $\boldsymbol{s}_\theta$
LiDAR	Lever arm Boresight	$\boldsymbol{x}^b_{bl}$ $\boldsymbol{R}^b_l$
Objects	Plane parameters	o, z_1, z_2

platforms: GNSS positions, IMU inertial measurements and LiDAR plane observations. The joint adjustment ensures a common datum for the ALS and ULS point clouds, while still accounting for the underlying measurements and their stochastic properties. This is done without use of any ground control information (i.e., direct georeferencing), and the absolute accuracy is verified by comparison to independently acquired reference data on the ground. This reference data consists of 106 planar reference surfaces, with 58 horizontal surfaces on the ground and 48 sloped surfaces on building roofs. Comparison of the ALS point cloud to the reference data is done by fitting a plane to all points within 1 m of a reference surface centre and computing the average distance from point cloud to surface centre w.r.t. the estimated surface normal direction. The sloped surfaces are oriented in sufficiently different directions, and a possible 3D datum shift vector is determined by simple least-squares adjustment. Only two reference targets are situated in the area covered by the ULS data acquisition. These are not large surfaces but rather small photogrammetric checkerboard targets. Thus, they are suitable only for evaluation of the ULS point cloud and not of the ALS point cloud, due to insufficient point density and too large beam footprint of the latter in comparison to the size of the target.

Both kinematic datasets were acquired almost concurrently on the same day in March 2021 in the area of the Pielach river in Lower Austria (Mandlburger et al., 2023b). The ALS dataset is an airborne laser bathymetry dataset acquired with a *RIEGL* VQ-880-GH laser scanner, featuring both an infrared (1064 nm) and a green (532 nm) laser channel. The laser scanner is integrated with an *Applanix* AV 510 navigation system with a type 46 IMU. Point cloud and trajectory are shown in Fig. 8 together with the locations of the reference surfaces. The ALS dataset contains in total 770 million points acquired in 8 flight strips, with simultaneous acquisition in each strip of infrared and green wavelength LiDAR channels. At the same time, a UAV-borne data acquisition was done with a *RIEGL* VQ-840-G bathymetric laser scanner with a green (532 nm) LiDAR unit, integrated with an *Applanix* APX-20 navigation system with a type 82 IMU. The ULS dataset, shown in Fig. 9 together with the two reference targets, contains approximately 340 million points, acquired in 27 flight segments flown in irregular pattern along the Pielach river. Although this is a topo-bathymetric data acquisition, we filter out non-ground points by using only last echos with sufficiently high reflectance, in order to avoid dealing with refraction effects and to focus on evaluating the georeferencing quality.

The full processing pipeline is shown in Fig. 10 and includes separate ALS processing, ULS processing, and the joint processing. The general steps for the separate processing are identical: After pre-processing the raw GNSS data to obtain a position solution, a GNSS/IMU integration is performed to obtain an initial trajectory. The nodes for the cubic

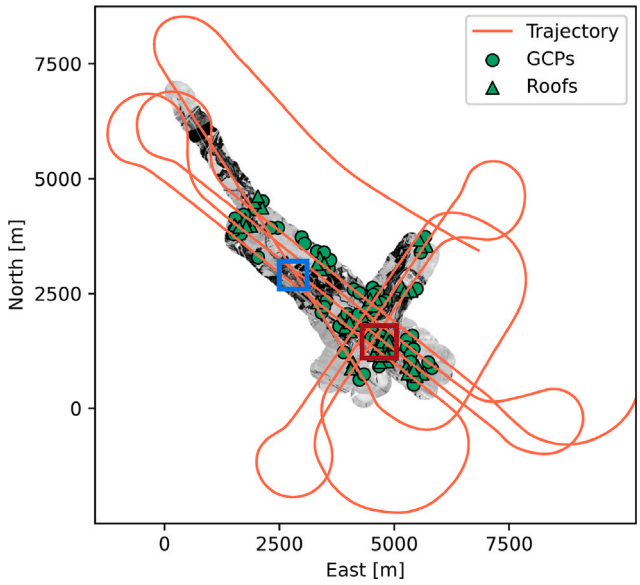


Fig. 8. ALS point cloud coloured by reflectance (dark/bright = low/high reflectance), with the trajectory overlaid in red. The red rectangle marks the subset of the ALS point cloud selected for visualization of the results below, which covers the Loosdorf village centre. The blue rectangle marks the ULS study area (Fig. 9). The green circles indicate horizontal reference points, the green triangles indicate sloped reference surfaces (i.e., rooftops). The reference points and surfaces were measured in a standard RTK survey, directly in the first case and via total station for the latter. (Reference surfaces are slightly offset from their true location for better visibility).

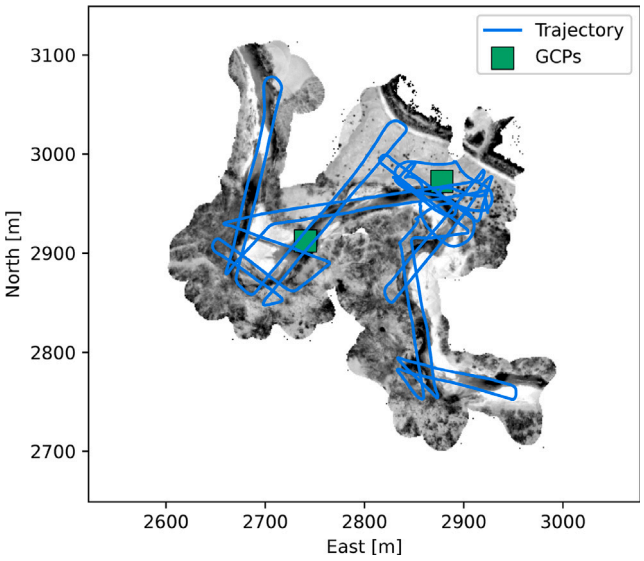


Fig. 9. ULS point cloud coloured by reflectance, with the trajectory overlaid in blue. The ULS study area is located along the Pielach river, as indicated with a blue rectangle in Fig. 8. The green squares represent two photogrammetric targets, square 2 × 2 checkerboards with side length 30 cm, used for evaluation of the ULS point cloud.

position splines and quadratic orientation splines are placed at the sample times of the respective IMUs, which have a measurement frequency of 200 Hz. The initial trajectory together with the nominal LiDAR mounting calibration is used to preliminarily georeference the point cloud. Based on this initial point cloud, planar features are extracted and matched to obtain the LiDAR plane observations, using cells of size $h_x = h_y = h_z = 2$ m. Note that the same rasterization (and thus cell size) is used for both the ALS and the ULS point clouds. For georeferencing and adjustment of the ALS data, all sensors, specifically both LiDAR channels, are referenced to the same trajectory. To reduce the number

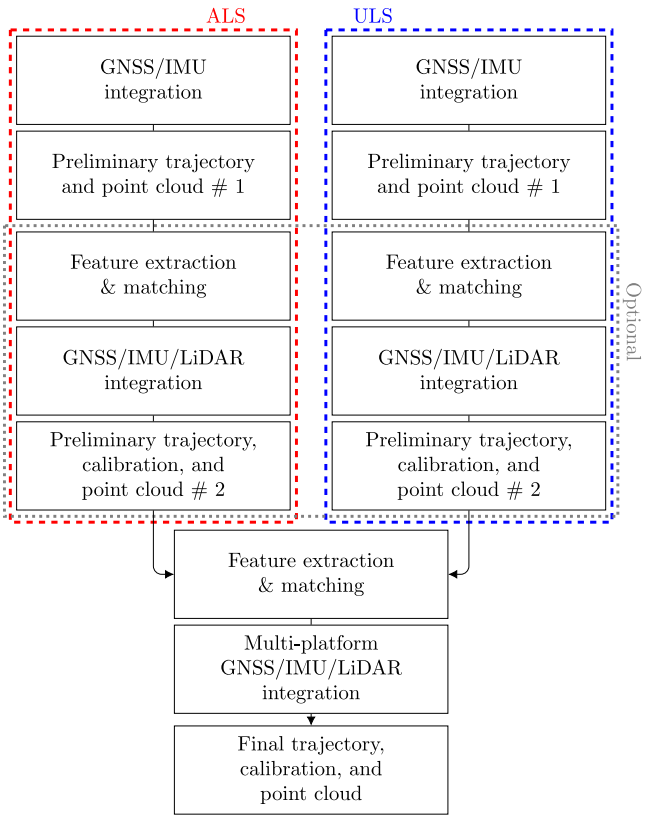


Fig. 10. Workflow for joint boresight calibration and trajectory estimation of ALS and ULS data acquisitions. Note that the boresight for the ALS system is initially unknown, causing discrepancies of up to 1 m in the initial georeferencing. For this reason, the feature matching and extraction process is first done separately for the two datasets, but this step is otherwise optional. It is not necessary to further iterate the feature extraction and matching process, and the same cell size of 2 m is used for all three plane extractions.

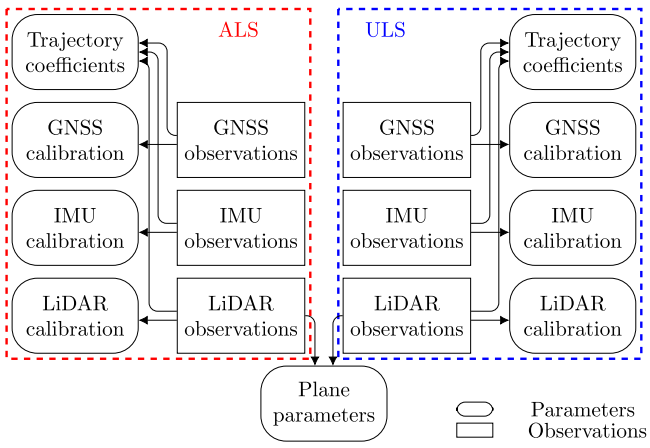


Fig. 11. Graphical representation of the parameters and measurements occurring in the joint adjustment. Specifically, each type of observation is connected to the parameters occurring in the respective measurement equation. The fictional observations derived from the stochastic priors are not shown here.

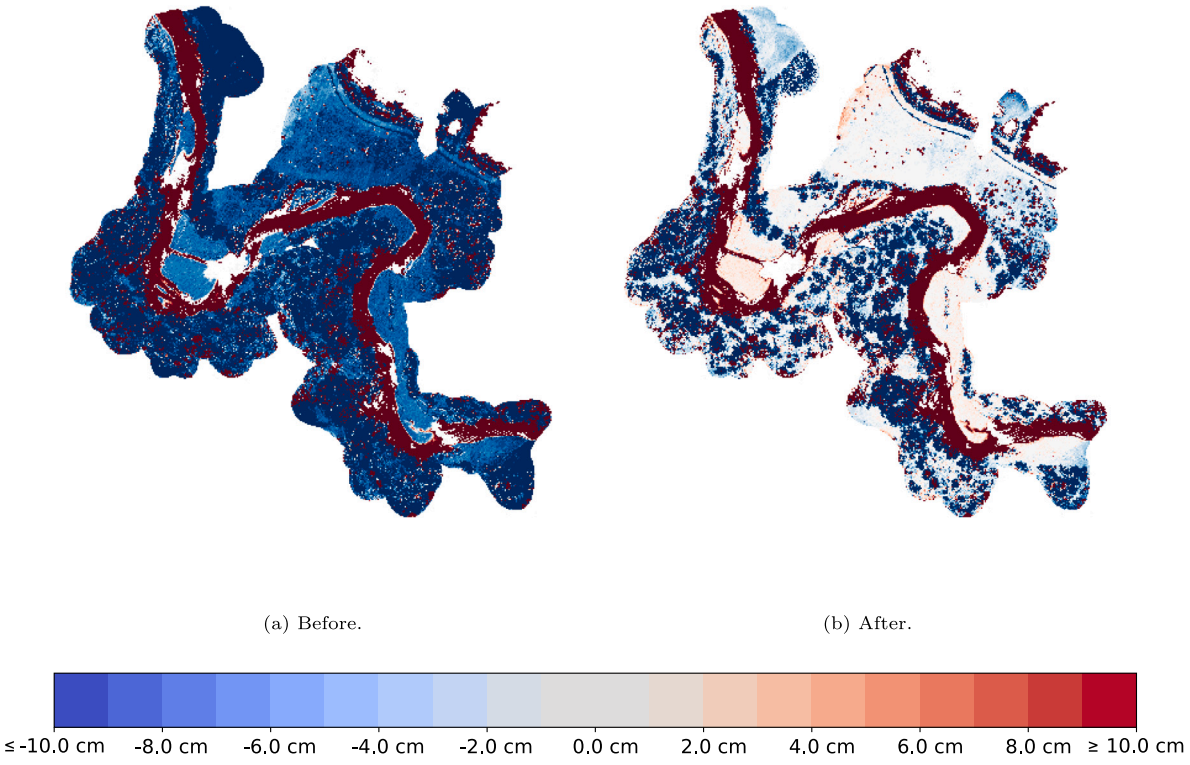
of LiDAR plane observations and parameters, we spatially subsample the ALS raster, keeping only those planes with the best fit. Specifically, we first subsample all planes by $5 \times 5 \times 5$, then the horizontal planes by $20 \times 20 \times 20$. This is done to ensure a large number of non-horizontally oriented planes, which are essential in constraining lateral shifts. The ALS dataset was acquired with an uncalibrated test system (i.e., fully unknown boresight angles). To empirically account for a non-constant boresight between the two laser scanners, the boresight angles of the green laser channel had to be modelled as a linear function. The LiDAR observations are now also considered in the adjustments, allowing boresight calibration within the trajectory estimation procedure. The resulting trajectory and boresight calibration is used to re-georeference the point cloud. The GNSS/IMU/LiDAR adjustment may be done (a) completely separately, or (b) in a joint adjustment. To understand the differences and possible benefits, we first compare the individual point clouds resulting from (a), i.e., fully decoupled processing of both datasets, and then evaluate the results of (b), the joint processing.

For the GNSS processing of both datasets, the GNSS positions are computed in post-processing kinematic (PPK) mode using the RTKLIB software (Takasu and Yasuda, 2009). The GNSS base station is located approximately 40 km away in Amstetten, Lower Austria, resulting in a relatively long baseline. In previous work (Pöppl et al., 2023a), we have observed difficulties with ambiguity resolution with RTKLIB for long baselines and expect this to significantly impact the quality of the GNSS solution, especially for the UAV platform, which has a lower-grade GNSS receiver and antenna in comparison to the ALS system. This becomes immediately apparent in the separate processing, as there is a large discrepancy between the ALS and the ULS point cloud. Fig. 12(a) shows the height differences between the ALS block and the ULS block, which indicate at least a vertical offset of about 7 cm. This is confirmed by comparison with the reference data: The ALS data achieves an RMSE of 3.85 cm with an estimated offset below 1.5 cm horizontally and 1 cm vertically. On the other hand, comparison of the two available reference targets to the ULS data again shows errors of 7 cm in the vertical component, which is notably larger than the expected accuracy of the reference data.[2]

The relative fit of both datasets and the – so far unsatisfactory – absolute accuracy of the ULS datasets is expected to improve in a joint processing of both datasets, without needing to introduce any ground control. To this end, not only do we consider the navigation data and LiDAR data together, but also both datasets simultaneously, which further improves the redundancy and allows the ULS dataset to be non-rigidly co-registered with the ALS dataset. Fig. 11 illustrates the interdependency of parameters and observations in this combined processing approach, where both platforms are included in a single adjustment, effectively coupled to each other through the object space parameters. With this approach, the ULS point cloud's datum is defined through the ALS data, or more specifically through the object space plane parameters. Through the joint adjustment, the discrepancy between the ALS and ULS block is reduced to a maximum of a few centimeters (Fig. 12(b)). The height discrepancy between ULS point cloud and the two reference targets is now 0.56 cm and 1.67 cm, well within the specified accuracy of the reference targets. Fig. 13 shows the maximum strip height differences for this block, which are below 3 cm for areas which are approximately planar (i.e., specifically excluding vegetation), with standard deviations largely below 2 cm for those areas. The residual errors in the height differences are caused by high-frequency errors in the orientation, which are especially visible here as data recording was active throughout the flight, including in turns where the UAV turns and de/accelerates very quickly. The errors are likely either due to inaccurate time synchronization between IMU and LiDAR, or simply due to the high-frequency dynamics not being resolved correctly by the IMU's 200 Hz sampling rate.

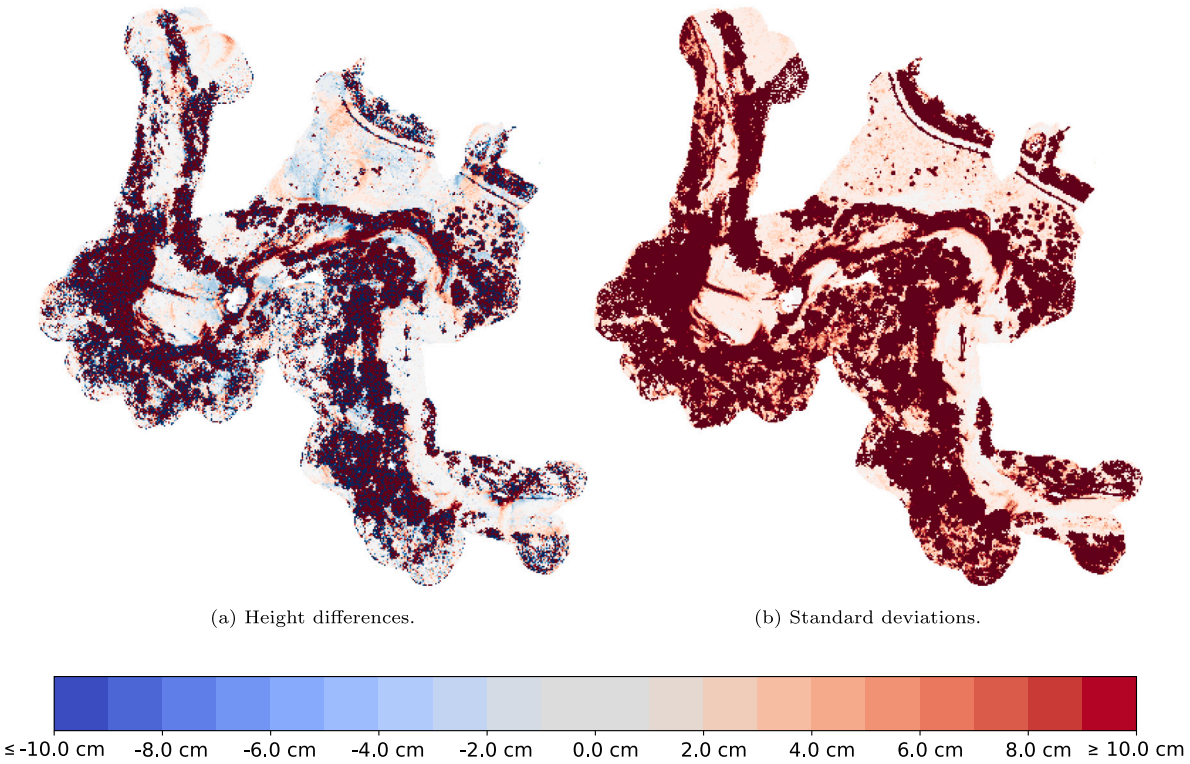
For the ALS dataset, strip differences for all strips and a selected study area in the Loosdorf village centre are given in Fig. 14. The strip differences show good consistency between the infrared point clouds from all strips, with discrepancies below 3 cm. However, slightly larger discrepancies are visible in the strip differences of the green laser

[2] RTK measurements with 4 cm accuracy according to the RTK service provider.



(a) Before. (b) After.

Fig. 12. Height difference between the ALS and ULS blocks before and after joint adjustment. Computed with a grid size of 1 m.



(a) Height differences. (b) Standard deviations.

Fig. 13. Maximum residual height differences of all strips to the height average, and standard deviations of the ULS strips. Computed with a grid size of 1 m.

channel, and also in the combined infrared and green strip differences. These differences mostly occur at smaller scales, such as for building roofs (especially edges) and in vegetated areas, due to the different beam characteristics between the infrared channel (smaller beam footprint, longer pulse duration) and the green channel (larger beam footprint, shorter pulse duration). In addition, some higher frequency trajectory errors are present, again likely caused by insufficient IMU sampling frequency compared to the frequency of platform vibrations. This causes errors in individual scan lines, while neighbouring scan lines fit well.

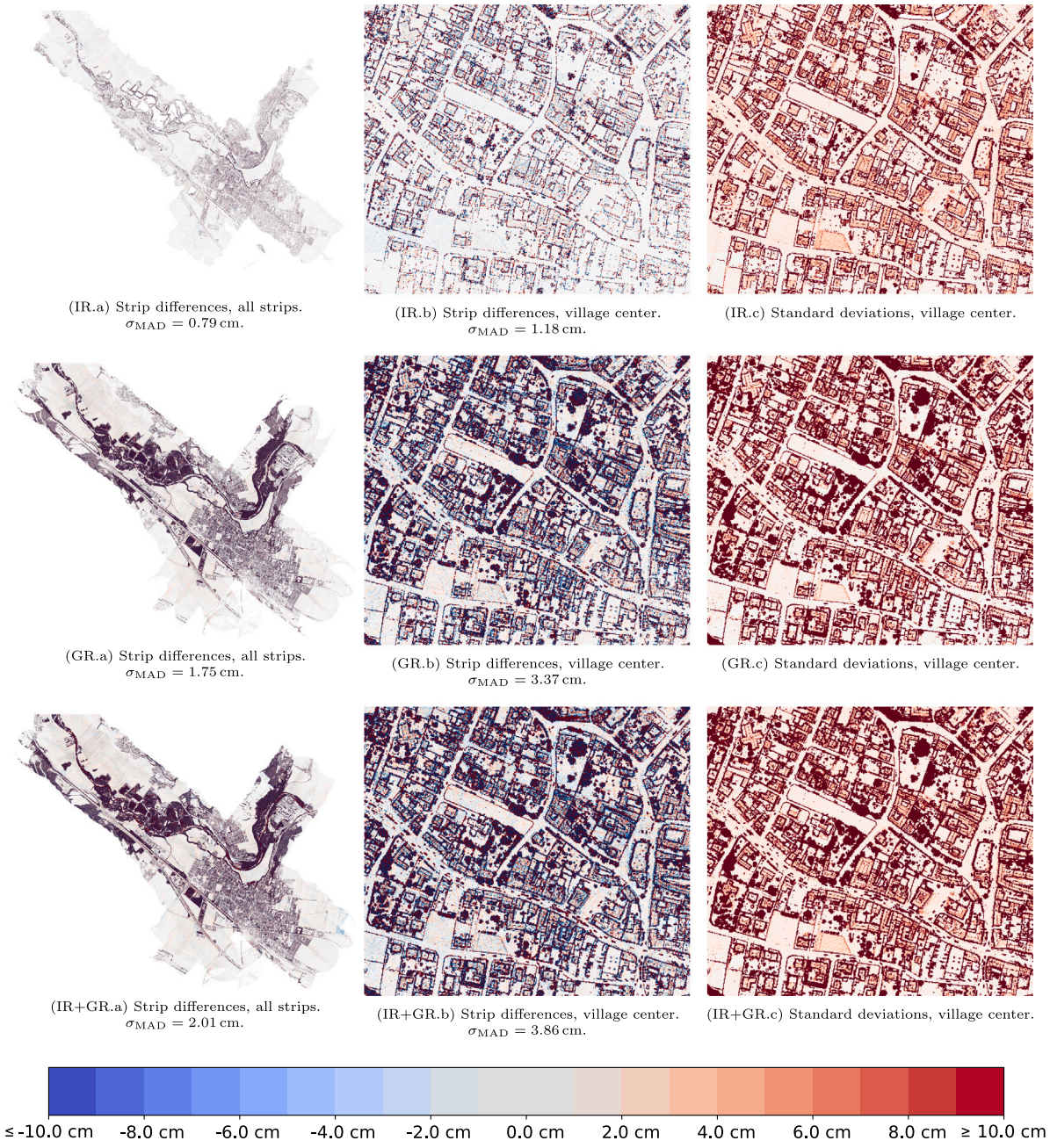
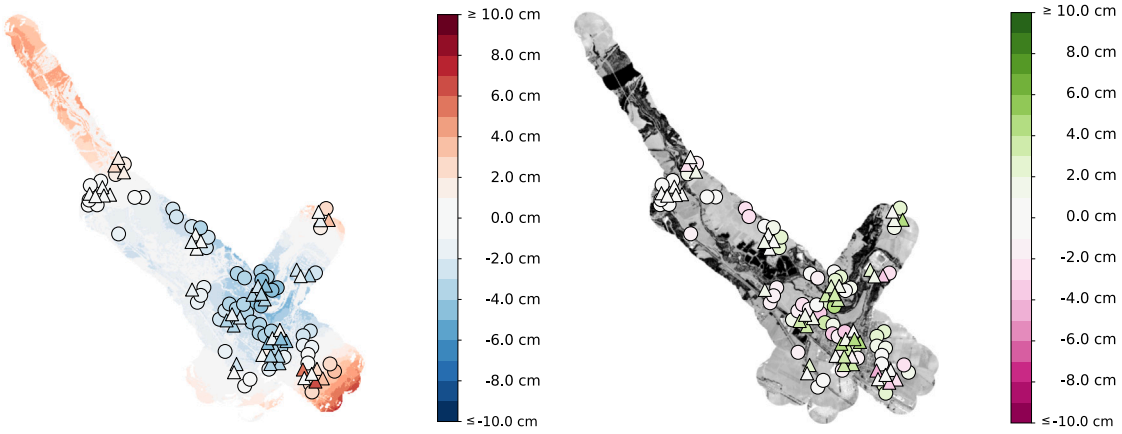


Fig. 14. Maximum normal differences (left, middle) and standard deviations (right) of the ALS strips, infrared (IR) and green (GR) channel separately and combined (IR+GR). Normal differences are distances of each strip's average to the combined average w.r.t. the combined surface normal. Only last echos are used, and points with low reflectance or high pulse shape deviation are filtered out beforehand. To account for e.g. vegetation, the overall precision is reported using the robust median absolute deviations of the IR, GR and IR+GR strip differences. Computed with a grid size of 1 m.

To evaluate absolute accuracy, the estimated root mean square error (RMSE), and standard deviation (SD) of the normal distances of the ALS point cloud w.r.t. the 106 reference surfaces is computed and given in Table 2, together with the corresponding robust statistics.

As a baseline, we also provide the respective value of an industry-standard georeferencing solution, computed using the *Applanix* POSPac software for GNSS/IMU integration via Kalman filter and *RIEGL* RiPRECISION for strip adjustment. Our solution achieves a notably lower RMSE of 3.85 cm compared to 7.43 cm, which is largely due to a constant datum error. The reason for this shift is unknown and may include either software or user error at some point during processing. As in our solution we have full control over the whole georeferencing pipeline, great care was taken to ensure the correct datum is used throughout. In conjunction with the redundancy introduced by the overlapping flight strips, constant positioning errors are avoided and time-dependent positioning errors may be corrected for in the combined adjustment. As a measure of precision, we also include the standard deviation (point spread) of all points-to-target normal distances in Table 2. Again, the proposed solution performs slightly better, with a standard deviation of 2.44 cm, which is consistent with the standard deviations of approximately planar surfaces shown in Fig. 14 (IR+GR.c).

The LiDAR stochastic model introduced in Section 2.7.3 is essential in avoiding block deformation while still allowing enough flexibility in the models to correct moderate time-varying GNSS errors and ensure good consistency between different flight strips. Fig. 15(a) shows the height difference between the point clouds from the two processing



(a) Block height difference between point clouds resulting from joint adjustment using (1) the purely data-driven stochastic model Eq. (2.47) and (2) the adapted stochastic model Eq. (2.48). The circles (triangles) represent the horizontal (sloped) reference surfaces and are colored by the average normal distance between points from (1) and from (2) belonging to the same target. The background is colored by height difference between the respective point clouds in a 1 m by 1 m grid.

(b) For each reference surface $i \in \{1, \ldots, 106\}$ and block $l \in \{1, 2\}$ the average normal distance d_i^l from point cloud to surface center is computed. The circles (triangles) representing the horizontal (sloped) surfaces are colored by the differences of the absolute normal distances between point cloud and target $|d_i^1| - |d_i^2|$. The RMSE is higher for the data-driven stochastic model (1), with 4.42 cm compared to 3.85 cm for the adapted stochastic model (2). The background is colored by reflectance (dark/bright = low/high reflectance).

Fig. 15. Differences between (1) the purely data-driven stochastic model Eq. (2.47) and (2) the adapted stochastic model Eq. (2.48) of the two point clouds (a), and of the respective distances between point cloud and reference surface (b). (Reference surfaces are slightly offset from their true location for better visibility.)

Table 2
Computed datum shift, root mean square error (RMSE), median absolute error (MAE), standard deviation (SD), median absolute deviation (MAD), and point spread w.r.t. to the reference surfaces. These values are provided both for the baseline solution, processed with the manufacturer software, and our proposed solution. Shift, RMSE, MAD and SD are derived by fitting the point clouds to the reference targets. Spread is the combined standard deviation of all points' residuals w.r.t. their corresponding plane fit.

[cm]		Baseline	Proposed
Shift	East	−3.26	1.12
	North	3.25	−1.17
	Up	−6.80	−0.63
RMSE		7.43	3.85
MAE		6.22	2.46
SD		4.54	3.83
MAD		3.06	2.36
Spread (1σ)		3.01	2.44

runs: (1) with the purely data-driven LiDAR stochastic model Eq. (2.47) and (2) with the proposed LiDAR stochastic model Eq. (2.48) that accounts for measurement errors which are correlated for points belonging to the same feature plane. Although only the LiDAR stochastic model differs in the processing of (1) and (2), the resulting point clouds exhibit systematic deformation compared to each other. The differences are most visible in the centre, where the strips overlap and thus the most LiDAR correspondences are present, and at the beginning and end of the strips, where the opposite is the case. Fig. 15(b) shows the difference of the absolute distances between point cloud and the reference surfaces: A green marker indicates that block (1) has a larger absolute error w.r.t. the reference surface, and a pink marker indicates that block (2) has a larger absolute error. The systematic behaviour seen in Fig. 15(a) is not as obvious in Fig. 15(b), where we compare both blocks to the reference surfaces, which are themselves also only accurate at the cm-level. Also, only few or no horizontal reference surfaces are available at the northernmost and southernmost ends of the strips — where the difference between the two blocks reaches 10 cm. Nonetheless, the use of the adapted stochastic model in block (2) results in an overall reduced RMSE of 3.85 cm compared to 4.42 cm for block (1). In case (1), the LiDAR correspondences are weighted much higher compared to case (2), and also weighted higher in comparison to the navigation data. This different weighting leads to deformation of the whole trajectory and thus point cloud. Although absolute accuracy of the laser scanning data can only be judged where reference data is available and to the level of accuracy of this reference data, our results show the LiDAR stochastic model has a notable impact on the final point cloud, and strongly indicate that the adapted stochastic model does in fact improve georeferencing accuracy.

In summary, the result of the joint adjustment is a combined point cloud which is consistent both within the acquisition type (i.e., strip differences for ALS and ULS separately), with respect to each other (block difference between the ALS and ULS datasets) and accurate in comparison to reference data. The initial discrepancy between both datasets is corrected by the co-registration as part of the GNSS observation error budget. Note again that the adjustments use no ground control, and the reported absolute accuracy is verified using completely independent terrestrially surveyed data.

The multi-platform adjustment is of course more complex and requires more computational effort compared to stand-alone adjustments followed by rigid co-registration. Table 3 reports the runtimes for the different stages of processing. Even though the various adjustments are computationally expensive, the total runtime is dominated by the LiDAR georeferencing, simply due to the large amount of points involved. Similarly, while Kalman filtering might be a faster way of obtaining an initial solution, the runtime of the sliding-window adjustment approach is still negligible compared to the combined runtime of the full adjustments and georeferencing. Nevertheless, adjustment runtime and also memory requirements are expected to increase at the very least linearly with acquisition time, and further work is both needed and planned in order to allow efficient processing of large datasets.

4. Conclusion

In this contribution, we have presented a comprehensive methodology for integrating navigation sensors with the payload LiDAR sensor in kinematic mapping applications. As a practical demonstration, the methodology is applied to direct georeferencing of airborne crewed and uncrewed data acquisitions without ground control. The result of this joint ALS and ULS adjustment procedure is a combined point cloud that is both consistent and accurate: Through analysis of strip differences, both within each dataset and compared to each other, we show considerable improvements (a) in absolute accuracy and relative precision when comparing the ALS results from our methodology to a standard

Table 3
Runtimes in seconds and memory usage in GiB for the different stages of processing, as shown in Fig. 10. The reported memory usage is the maximum of each step. LiDAR georeferencing is done using *RIEGL* software. The Kalman filtering solution is processed using a quaternion unscented Kalman filter with forward–backward smoothing; the Kalman filter runtime is mentioned here only to provide additional context, the resulting smoothed GNSS/IMU trajectory is not used further. Processing was done on a Linux workstation with an *AMD Ryzen 7 2700X* CPU, 64 GiB RAM and an NVMe drive.

	ALS		ULS
GNSS/IMU smoother	0:48 min		0:14 min
GNSS/IMU initialization	2:39 min 0.90 GiB		1:09 min 2.04 GiB
GNSS/IMU adjustment	1:41 min 7.15 GiB		1:00 min 4.45 GiB
Georeferencing	8:13 min 6.41 GiB		5:05 min 4.68 GiB
Plane extraction and matching	3:49 min 21.23 GiB		1:19 min 11.77 GiB
GNSS/IMU/LiDAR adjustment	4:01 min 11.04 GiB		0:53 min 4.90 GiB
Georeferencing	8:04 min 6.30 GiB		4:58 min 4.44 GiB
Plane extraction and matching		5:40 min 21.23 GiB	
GNSS/IMU/LiDAR ALS/ULS adjustment		8:12 min 15.09 GiB	
Georeferencing	8:08 min 6.63 GiB		4:40 min 4.41 GiB

processing workflow and (b) in absolute accuracy when comparing the jointly processed ULS dataset to the individually processed ULS dataset. For the ALS dataset, an absolute accuracy with an RMSE of below 4 cm is achieved. In terms of precision, the point spread on planar surfaces shows a standard deviation also below 4 cm for the combined infrared and green dataset, both for the reference surfaces themselves and for other planar surfaces such as roads and building roofs.

The proposed framework is highly flexible and allows for various sensor configurations, e.g., multiple laser scanners with different wavelengths (in this article), a laser scanner on a gyro stabilization mount (Mandlburger et al., 2023b), multiple GNSS antennas (Pöppl et al., 2023e) and non-standard kinematic scanning platforms such as a terrestrial laser scanner (Pöppl et al., 2023d). Using appropriate stochastic models, both consumer-grade navigation sensors (Pöppl et al., 2023e) and survey-grade navigation sensors (in this article, Pöppl et al., 2023a) are processed with the same functional model and implementation, achieving high quality trajectories even for low-cost navigation components but without sacrificing accuracy for high-end systems. The proposed method can be seen as an extension to standard strip adjustment and used as such, i.e., without IMU and GNSS measurements. While there is little advantage to do so in cases where the navigation data is available, missing GNSS and IMU measurements may be replaced with stochastic priors, which function as fictional time-correlated measurements and constrain the corrections to position and orientation (Mandlburger et al., 2024).

In any case, the main focus is on the LiDAR data itself, with the LiDAR measurements being in this context not just an additional input for navigation purposes but rather the final product. The LiDAR measurement model is based on planar surfaces as geometric primitives for constructing LiDAR correspondences. The stochastic model is extended here to account for measurement effects dependent on sensor characteristics such as ranging and angular accuracy as well as pulse duration, footprint size and incidence angle on the target. So far, we have relied exclusively on extracting and matching planar surfaces. However, the general methodology could be adapted for different observation types (point-to-point, or higher-order surfaces) in the future, with no impact on the rest of the estimation procedure. Different feature types, for example for forest environments, may be straightforwardly integrated as the feature extraction and matching procedures are conceptually and practically decoupled from the rest of the processing pipeline.

While the results shown in Section 3 are promising, the computational effort required for the full adjustment, especially when simultaneously processing multiple datasets, is high. This is especially the case since our method relies on processing inertial measurement data at full output rate, in order to capture any motion which may impact the LiDAR measurements. Reducing the computational effort becomes even more relevant as newer IMU generations output measurements at higher rates. In the future, IMU pre-integration could be used together with a trajectory reconstruction scheme which ensures the full fidelity is restored after adjustment. Additionally, parallelization schemes such as domain decomposition, or split-optimization methods such as the alternating direction method of multipliers (ADMM) are candidates for improving runtime performance as for many applications the problem is relatively easily decomposable into smaller problems, e.g., different platforms or strips, coupled together through the object space parameters.

CRediT authorship contribution statement

Florian Pöppl: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Andreas Ullrich:** Writing – review & editing, Software, Resources, Project administration, Methodology, Data curation, Conceptualization. **Gottfried Mandlburger:** Writing – review & editing, Supervision, Methodology, Data curation, Conceptualization. **Norbert Pfeifer:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: This work was carried out as part of the project ZAP-ALS (883660), a joint project between the Technische Universität Wien and the company RIEGL, which is funded by the Austrian Research Promotion Agency (FFG). Dr. Andreas Ullrich is the Technical Director of RIEGL Laser Measurement Systems GmbH. Florian Pöppl consults RIEGL on selected technical issues related to satellite and inertial navigation but outside the scope of this work.

Acknowledgements

This work was carried out as part of the project ZAP-ALS (883660), a joint project between the *Technische Universität Wien* and the company *RIEGL*, which is funded by the Austrian Research Promotion Agency (FFG[3]). The authors also acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Program.

References

Agarwal, S., Mierle, K., Team, T.C.S., 2022. Ceres solver.
Al-Jlailaty, H., Mansour, M.M., 2021. Efficient attitude estimators: a tutorial and survey. J. Signal Process. Syst. http://dx.doi.org/10.1007/s11265-020-01620-4.
Barfoot, T.D., 2017. State Estimation for Robotics. Cambridge University Press, Cambridge, http://dx.doi.org/10.1017/9781316671528.
Beuchert, J., Camurri, M., Fallon, M., 2023. Factor graph fusion of raw GNSS sensing with IMU and LiDAR for precise robot localization without a base station. In: 2023 IEEE International Conference on Robotics and Automation. ICRA, pp. 8415–8421. http://dx.doi.org/10.1109/ICRA48891.2023.10161522.

[3] Österreichische Forschungsförderungsgesellschaft (FFG), Wien, Austria, www.ffg.at.

Biber, P., Strasser, W., 2003. The normal distributions transform: A new approach to laser scan matching. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453). Vol. 3, pp. 2743–2748. http://dx.doi.org/10.1109/IROS.2003.1249285, vol.3.

Brun, A., Cucci, D.A., Skaloud, J., 2022. LiDAR point–to–point correspondences for rigorous registration of kinematic scanning in dynamic networks. ISPRS J. Photogramm. Remote Sens. 189, 185–200. http://dx.doi.org/10.1016/j.isprsjprs.2022.04.027.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J., 2016. Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. IEEE Trans. Robot. 32 (6), 1309–1332. http://dx.doi.org/10.1109/TRO.2016.2624754.

Chang, L., Niu, X., Liu, T., Tang, J., Qian, C., 2019. GNSS/INS/LiDAR-SLAM Integrated Navigation System Based on graph optimization. Remote Sens. 11 (9), 1009. http://dx.doi.org/10.3390/rs11091009.

Chen, Y., Davis, T.A., Hager, W.W., Rajamanickam, S., 2008. Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/downdate. ACM Trans. Math. Software 35 (3), 1–14. http://dx.doi.org/10.1145/1391989.1391995.

Cheng, L., Chen, S., Liu, X., Xu, H., Wu, Y., Li, M., Chen, Y., 2018. Registration of laser scanning point clouds: A review. Sensors 18 (5), 1641. http://dx.doi.org/10.3390/s18051641.

Cioffi, G., Cieslewski, T., Scaramuzza, D., 2022. Continuous-time vs. discrete-time vision-based SLAM: A comparative study. IEEE Robot. Autom. Lett. 7 (2), 2399–2406. http://dx.doi.org/10.1109/LRA.2022.3143303.

Crespillo, O.G., Joerger, M., Langel, S., 2020. Overbounding GNSS/INS Integration with uncertain GNSS Gauss–Markov error parameters. In: 2020 IEEE/ION Position, Location and Navigation Symposium. PLANS, IEEE, Portland, OR, USA, pp. 481–489. http://dx.doi.org/10.1109/PLANS46316.2020.9109874.

Crespillo, O.G., Medina, D., Skaloud, J., Meurer, M., 2018. Tightly coupled GNSS/INS Integration Based on robust M-estimators. In: 2018 IEEE/ION Position, Location and Navigation Symposium. PLANS, IEEE, Monterey, CA, pp. 1554–1561. http://dx.doi.org/10.1109/PLANS.2018.8373551.

Cucci, D.A., Clausen, P., Skaloud, J., Matteucci, M., 2017a. A general approach to time-varying parameters in pose-graph optimization. In: 2017 European Navigation Conference. ENC, IEEE, Lausanne, Switzerland, pp. 265–271. http://dx.doi.org/10.1109/EURONAV.2017.7954217.

Cucci, D.A., Rehak, M., Skaloud, J., 2017b. Bundle adjustment with raw inertial observations in UAV applications. ISPRS J. Photogramm. Remote Sens. 130, 1–12. http://dx.doi.org/10.1016/j.isprsjprs.2017.05.008.

Cucci, D.A., Skaloud, J., 2019. On raw inertial measurements in dynamic networks. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. IV-2/W5, 549–557. http://dx.doi.org/10.5194/isprs-annals-IV-2-W5-549-2019.

de Boor, C., 1980. A practical guide to splines. Math. Comp. 34 (149), 325. http://dx.doi.org/10.2307/2006241, arXiv:2006241.

Dong, J., Mukadam, M., Boots, B., Dellaert, F., 2018. Sparse Gaussian processes on matrix Lie groups: A unified framework for optimizing continuous-time trajectories. In: 2018 IEEE International Conference on Robotics and Automation. ICRA, IEEE, Brisbane, QLD, pp. 6497–6504. http://dx.doi.org/10.1109/ICRA.2018.8461077.

Farrell, J.A., Silva, F.O., Rahman, F., Wendel, J., 2022. Inertial measurement unit error modeling tutorial: Inertial navigation system state estimation with real-time sensor calibration. IEEE Control Syst. 42 (6), 40–66. http://dx.doi.org/10.1109/MCS.2022.3209059.

Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D., 2015. IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In: Robotics: Science and Systems XI. Robotics: Science and Systems Foundation, http://dx.doi.org/10.15607/RSS.2015.XI.006.

Furgale, P., Tong, C.H., Barfoot, T.D., Sibley, G., 2015. Continuous-time batch trajectory estimation using temporal basis functions. Int. J. Robot. Res. 34 (14), 1688–1710. http://dx.doi.org/10.1177/0278364915585860.

Glira, P., Pfeifer, N., Briese, C., Ressl, C., 2015. A correspondence framework for ALS strip adjustments based on variants of the ICP algorithm. Photogramm. Fernerkundung Geoinform. 2015 (4), 275–289. http://dx.doi.org/10.1127/pfg/2015/0270.

Glira, P., Pfeifer, N., Mandlburger, G., 2016. Rigorous strip adjustment of UAV-based laserscanning data including time-dependent correction of trajectory errors. Photogramm. Eng. Remote Sens. 82 (12), 945–954. http://dx.doi.org/10.14358/PERS.82.12.945.

Glira, P., Pfeifer, N., Mandlburger, G., 2019. Hybrid orientation of airborne lidar point clouds and aerial images. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. IV-2/W5, 567–574. http://dx.doi.org/10.5194/isprs-annals-IV-2-W5-567-2019.

Groves, P.D., 2013. Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, second ed. In: GNSS Technology and Application Series, Artech House, Boston.

Haarbach, A., Birdal, T., Ilic, S., 2018. Survey of higher order rigid body motion interpolation methods for keyframe animation and continuous-time trajectory estimation. In: 2018 International Conference on 3D Vision. 3DV, IEEE, Verona, pp. 381–389. http://dx.doi.org/10.1109/3DV.2018.00051.

Huber, P.J., 1964. Robust estimation of a location parameter. Ann. Math. Stat. 35 (1), 73–101. http://dx.doi.org/10.1214/aoms/1177703732.

Johnson, J., Mangelson, J., Barfoot, T., Beard, R., 2024. Continuous-time trajectory estimation: A comparative study between Gaussian process and spline-based approaches. arXiv:2402.00399.

Jonassen, V.O., Kjørsvik, N.S., Gjevestad, J.G.O., 2023. Scalable hybrid adjustment of images and LiDAR point clouds. ISPRS J. Photogramm. Remote Sens. 202, 652–662. http://dx.doi.org/10.1016/j.isprsjprs.2023.07.007.

Kager, H., 2004. Discrepancies between overlapping laser scanner strips-simultaneous fitting of aerial laser scanner strips. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 35, 555–560.

Kim, M.-J., Kim, M.-S., Shin, S.Y., 1995. A general construction scheme for unit quaternion curves with simple high order derivatives. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '95. ACM Press, pp. 369–376. http://dx.doi.org/10.1145/218380.218486.

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P., 2015. Keyframe-based visual–inertial odometry using nonlinear optimization. Int. J. Robot. Res. 34 (3), 314–334. http://dx.doi.org/10.1177/0278364914554813.

Li, X., Yu, H., Wang, X., Li, S., Zhou, Y., Chang, H., 2023. FGO-gil: factor graph optimization-based GNSS RTK/INS/LiDAR Tightly Coupled Integration for precise and continuous navigation. IEEE Sens. J. http://dx.doi.org/10.1109/JSEN.2023.3278723.

Lv, J., Hu, K., Xu, J., Liu, Y., Ma, X., Zuo, X., 2021. CLINS: Continuous-time trajectory estimation for LiDAR-inertial system. arXiv:2109.04687 [cs].

Lv, J., Lang, X., Xu, J., Wang, M., Liu, Y., Zuo, X., 2023. Continuous-time fixed-lag smoothing for LiDAR-inertial-camera SLAM. IEEE/ASME Trans. Mechatronics 28 (4), 2259–2270. http://dx.doi.org/10.1109/TMECH.2023.3241398.

Mandlburger, G., Kölle, M., Pöppl, F., Cramer, M., 2023a. Evaluation of consumer-grade and survey-grade UAV-LiDAR. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLVIII-1/W3-2023, 99–106. http://dx.doi.org/10.5194/isprs-archives-XLVIII-1-W3-2023-99-2023.

Mandlburger, G., Kölle, M., Pöppl, F., Cramer, M., 2024. Potential and limitations of consumer-grade UAV-LiDAR for 3D topographic mapping. AVN - Allgemein. Vermess. Nachr. (2), 77–90. http://dx.doi.org/10.14627/avn.2024.2.2.

Mandlburger, G., Pfennigbauer, M., Schwarz, R., Pöppl, F., 2023b. A decade of progress in topo-bathymetric laser scanning exemplified by the pielach river dataset. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. X-1-W1-2023, 1123–1130. http://dx.doi.org/10.5194/isprs-annals-X-1-W1-2023-1123-2023.

Mouzakidou, K., Brun, A., Cucci, D.A., Skaloud, J., 2024. Airborne sensor fusion: Expected accuracy and behavior of a concurrent adjustment. ISPRS Open J. Photogramm. Remote Sens. 100057. http://dx.doi.org/10.1016/j.ophoto.2023.100057.

Nurunnabi, A., West, G., Belton, D., 2015. Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. Pattern Recognit. 48 (4), 1404–1419. http://dx.doi.org/10.1016/j.patcog.2014.10.014.

Pfeifer, N., Briese, C., 2007. Geometrical aspects of airborne laser scanning and terrestrial laser scanning. In: Proceedings of the ISPRS Workshop Laser Scanning 2007: IAPRS Volume XXXVI, Part 3 / W52. Espoo, Finland.

Pöppl, F., Mandlburger, G., Pfeifer, N., 2023a. Evaluation of a GNSS/IMU/LiDAR-Integration for airborne laser scanning using RTKLIB PPK and PPP GNSS solutions. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLVIII-1/W3-2023, 161–166. http://dx.doi.org/10.5194/isprs-archives-XLVIII-1-W3-2023-161-2023.

Pöppl, F., Neuner, H., Mandlburger, G., Pfeifer, N., 2023b. Integrated trajectory estimation for 3D kinematic mapping with GNSS, INS and imaging sensors: A framework and review. ISPRS J. Photogramm. Remote Sens. 196, 287–305. http://dx.doi.org/10.1016/j.isprsjprs.2022.12.022.

Pöppl, F., Pfennigbauer, M., Ullrich, A., Mandlburger, G., Neuner, H., Pfeifer, N., 2023c. Modelling of GNSS positioning errors in a GNSS/INS/LiDAR-integrated Georeferencing. In: Publikationen Der Deutschen Gesellschaft FÜR Photogrammetrie, Fernerkundung Und Geoinformation E.V.. pp. 183–196.

Pöppl, F., Pfennigbauer, M., Ullrich, A., Pfeifer, N., 2023d. Trajectory estimation with GNSS, IMU, and LiDAR for terrestrial/Kinematic Laser Scanning. In: Turner, M.D., Kamerman, G.W., Magruder, L.A. (Eds.), Laser Radar Technology and Applications XXVIII. SPIE, Orlando, United States, p. 8. http://dx.doi.org/10.1117/12.2663454.

Pöppl, F., Teufelsbauer, H., Ullrich, A., Pfeifer, N., 2023e. Mobile laser scanning with low-cost navigation sensors: compensating for low-grade IMU with dual-GNSS and tightly-coupled LiDAR. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLVIII-1/W1-2023, 403–410. http://dx.doi.org/10.5194/isprs-archives-XLVIII-1-W1-2023-403-2023.

Quinn, D.P., Ehlmann, B.L., 2019. A PCA-based framework for determining remotely sensed geological surface orientations and their statistical quality. Earth Space Sci. (Hoboken, NJ) 6 (8), 1378–1408. http://dx.doi.org/10.1029/2018EA000416.

Savage, P.G., 1998a. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. J. Guid. Control Dyn. 21 (1), 19–28. http://dx.doi.org/10.2514/2.4228.

Savage, P.G., 1998b. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. J. Guid. Control Dyn. 21 (2), 208–221. http://dx.doi.org/10.2514/2.4242.

Skaloud, J., Lichti, D., 2006. Rigorous approach to bore-sight self-calibration in airborne laser scanning. ISPRS J. Photogramm. Remote Sens. 61 (1), 47–59. http://dx.doi.org/10.1016/j.isprsjprs.2006.07.003.

Sommer, H., Forbes, J.R., Siegwart, R., Furgale, P., 2016. Continuous-time estimation of attitude using B-splines on Lie groups. J. Guid. Control Dyn. 39 (2), 242–261. http://dx.doi.org/10.2514/1.G001149.

Sommer, C., Usenko, V., Schubert, D., Demmel, N., Cremers, D., 2020. Efficient derivative computation for cumulative B-splines on Lie groups. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 11145–11153. http://dx.doi.org/10.1109/CVPR42600.2020.01116.

Soudarissanane, S., Lindenbergh, R., Menenti, M., Teunissen, P., 2009. Incidence angle influence on the quality of terrestrial laser scanning points. In: Proceedings of the ISPRS Workshop Laserscanning 2009: ISPRS Vol. 38 Part 3/W8.

Strasdat, H., Montiel, J., Davison, A.J., 2012. Visual SLAM: why filter? Image Vis. Comput. 30 (2), 65–77. http://dx.doi.org/10.1016/j.imavis.2012.02.009.

Takasu, T., Yasuda, A., 2009. Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. In: International Symposium on GPS/GNSS. Jeju, Korea, p. 7.

Toth, C., Jóźków, G., 2016. Remote sensing platforms and sensors: A survey. ISPRS J. Photogramm. Remote Sens. 115, 22–36. http://dx.doi.org/10.1016/j.isprsjprs.2015.10.004.

Zhang, J., Singh, S., 2014. LOAM: lidar odometry and mapping in real-time. In: Robotics: Science and Systems X. Robotics: Science and Systems Foundation, p. 9. http://dx.doi.org/10.15607/RSS.2014.X.007.