ISPRS Journal of Photogrammetry and Remote Sensing 196 (2023) 287–305

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

Integrated trajectory estimation for 3D kinematic mapping with GNSS, INS and imaging sensors: A framework and review

Florian Pöppl *, Hans Neuner, Gottfried Mandlburger, Norbert Pfeifer

Department of Geodesy and Geoinformation, TU Wien, Wiedner Hauptstraße 8/E120, 1040 Wien, Austria

ARTICLE INFO

Keywords:
Sensor orientation
Sensor fusion
Georeferencing
LiDAR
Camera

ABSTRACT

Trajectory estimation refers to the task of obtaining position and orientation estimates by fusing various sensor inputs. In kinematic mapping, global navigation satellite systems (GNSS) and inertial navigation systems (INS) are traditionally used to compute a trajectory which then serves as basis for direct or integrated orientation of the imaging sensors. As an inherently interdisciplinary problem, literature on trajectory estimation is broad. Apart from remote sensing itself, many recent advances come from autonomous navigation and robotics. This paper aims to provide a unified view of trajectory estimation with a focus on its role in kinematic mapping, specifically on the integration of GNSS, INS, laser scanners and cameras, as well as a survey of the related literature. Recent trends and challenges in trajectory estimation are identified and discussed.

1. Introduction

Laser scanning and photogrammetric imaging are widely used for remote sensing tasks such as mapping and surveying (Toth and Jóźków, 2016). These techniques rely on optical imaging sensors, specifically laser scanners and frame cameras, to obtain georeferenced 3D point clouds and other 3D models of the environment. *Kinematic mapping* refers to mapping with moving sensor platforms. It includes mobile mapping (e.g., car- or train-based) and airborne mapping (e.g., drone-, helicopter-, or plane-based). Modern kinematic mapping systems are multi-sensor systems (MSS), where all sensors are mounted together on a moving platform. In addition to the imaging sensors, the MSS typically includes a global navigation satellite system (GNSS) and an inertial navigation system (INS) to facilitate the determination of the platform *trajectory* (position and orientation over time).

The aim of this contribution is to provide an as-yet missing unified view of trajectory estimation, with focus on mapping and surveying applications and the accompanying requirements with respect to accuracy and sensors used. *Trajectory estimation* refers here to the task of estimating a trajectory based on various sensor inputs with respect to a given georeferenced coordinate system. In contrast to navigation, trajectory estimation emphasizes the recovery of position and orientation not just for the current moment but over a period of interest. More generally, *estimation* is the problem of recovering a systems' internal state from noisy measurements (Jazwinski, 1970).

In the context of surveying, the industry standard for trajectory estimation is GNSS/INS integration through Kalman filtering, exploiting the synergy between inertial sensors and satellite navigation (Groves, 2013; Toth and Jóźków, 2016). The resulting trajectory may still exhibit significant errors, especially in the case of challenging GNSS conditions, making the quality of the trajectory a limiting factor for the quality of the 3D model. These trajectory errors can be partially mitigated at the trajectory level, e.g., via subsequent trajectory correction (Glira et al., 2019; Zhou et al., 2021) by exploiting redundancy in the imaging sensor measurements. Alternatively, other additional sensors may be used to improve trajectory accuracy: In mobile mapping, distance measuring instruments commonly provide wheel odometry information (Meng et al., 2017). Magnetic field sensors provide heading information (Sabatini, 2006), but are hard to calibrate due to systematic distortions of the magnetic field. Range cameras, event cameras, 2D laser scanners and low-cost variants thereof are popular in robotics (Cadena et al., 2016; Chen et al., 2018b). However, this work focuses on survey-grade 3D laser scanners and frame cameras as imaging sensors, as used in high-accuracy mapping applications.

The ubiquity of sensors and the associated wealth of (possibly unsynchronized) measurements require versatile estimation methods capable of fusing the various types of sensor input. Many such methods fall under the umbrella of *simultaneous localization and mapping* (SLAM, cf. Cadena et al., 2016). The focus is on real-time capability, often for the purpose of autonomous navigation (Kolar et al., 2020). While the SLAM map is in many cases only of interest insofar as it provides a means for reliable and globally consistent localization,

* Corresponding author.
E-mail addresses: florian.poeppl@geo.tuwien.ac.at (F. Pöppl), hans.neuner@geo.tuwien.ac.at (H. Neuner), gottfried.mandlburger@geo.tuwien.ac.at (G. Mandlburger), norbert.pfeifer@geo.tuwien.ac.at (N. Pfeifer).

https://doi.org/10.1016/j.isprsjprs.2022.12.022
Received 2 July 2022; Received in revised form 4 November 2022; Accepted 22 December 2022
Available online 12 January 2023

autonomous methods also show promise for surveying applications (Nex et al., 2022). Kinematic mapping generally does not have the same real-time constraint, but has high accuracy requirements for the *map*, the georeferenced point cloud. Recently, novel methods have been proposed for kinematic mapping that directly integrate the measurements of the payload sensors with GNSS and INS for simultaneous determination of the sensor platforms' trajectory and sensor calibration (Rouzaud and Skaloud, 2011; Cucci et al., 2017; Brun et al., 2022). A common aspect in both robotics and remote sensing is the trend towards such *holistic* methods that incorporate as much available information as computationally feasible for the given application. This is in contrast to the standard two-step processing pipeline (e.g., Glira et al., 2019), based on GNSS/INS integration with subsequent adjustment, which was previously dominant in remote sensing (Toth and Jóźków, 2016).

Trajectory estimation is a task spanning many fields including navigation, remote sensing and robotics. On the navigation side, Farrell (2008) and Groves (2013) offer a comprehensive treatment of navigation solutions with focus on GNSS, INS and their integration via classical filtering and smoothing methods. Al-Jlailaty and Mansour (2021) give a detailed overview of the low-level theory of inertial navigation and GNSS/INS integration, and El-Sheimy and Youssef (2020) provide a review of modern inertial sensor technology. The use of these methods for remote sensing is reviewed in Colomina (2015). Recent innovations, especially concerning the use of additional sensors and multi-sensor fusion, often come from the fields of autonomous systems and robotics. Yuan et al. (2021) discuss localization from the perspective of unmanned aerial vehicles. Similarly, Qingqing et al. (2020) and Wang et al. (2020b) review autonomous multi-sensor fusion methods for vehicle applications. An in-depth treatment of classical SLAM is found in the tutorials of Durrant-Whyte and Bailey (2006) and Bailey and Durrant-Whyte (2006). Cadena et al. (2016) provide a survey of modern SLAM, including graph-based formalisms. While they discuss general SLAM formulations, Chen et al. (2018a) give an overview and comparison of filtering and optimization approaches for visual-inertial SLAM, and Debeunne and Vivet (2020) review visual-LiDAR[1] SLAM methods. A comparatively new development are continuous-time trajectory estimation methods, of which Furgale et al. (2015) give a short survey and exposition.

As this is a problem lying at the intersection of various fields, there exists a large body of literature on trajectory estimation and related topics. However, approaches to trajectory estimation differ in methodology and objectives, depending on field and application. Our contribution attempts a more unified treatment of trajectory estimation: While existing work usually approaches the problem from a field-specific perspective, we provide a general framework for trajectory estimation and a broad review of the relevant literature, although with a particular focus on remote sensing, mapping and surveying applications.

Specifically, this paper formulates a methodological framework for modelling of multi-sensor systems and estimation of their trajectory and calibration parameters. Due to the breadth of the topic, this is not an exhaustive survey of integrated navigation or data fusion literature in general, but rather an attempt to present the underlying theory and terminology in a coherent manner and highlight methods and trends relevant to remote sensing applications. Emphasis is put on new trends such as continuous-time trajectory modelling and tight coupling of the imaging sensors. The basic measurement equations for the most common sensor types are discussed together with their use in different estimation methodologies.

The rest of this paper is structured as follows: Section 2 gives an overview of the problem of trajectory estimation for kinematic mapping. Section 3 formulates trajectory estimation as a generic parameter estimation problem and discusses common estimation methodologies, such as Kalman filtering and smoothing as well as batch nonlinear least squares. Section 4 presents the underlying models for the trajectory as well as the different sensors' measurement processes. Section 5 reviews the trajectory estimation literature and examines existing methodologies according to the characteristics established in the previous sections. Section 6 provides a discussion on recent trends as well as future challenges.

[1] Laser scanners use *light detection and ranging* (LiDAR) together with a 2D or 3D scanning mechanism, and are therefore also referred to as *LiDAR systems*.

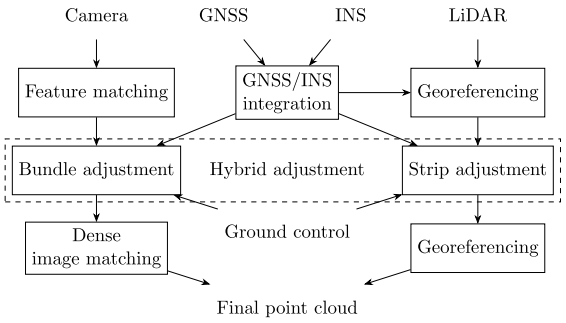

Fig. 1. Standard kinematic mapping processing pipeline with trajectory-level error modelling (see Sections 2.1 and 2.2).

2. Overview of trajectory estimation for kinematic mapping

The most common mapping products are 3D point clouds. The generation of these point clouds and their georeferencing (registration to a reference coordinate system) requires knowledge of the sensors' exterior orientation. Since the platform is mobile, the exterior orientation changes between measurements, but may be derived indirectly via measurement of known control points or directly from the platform trajectory. This is referred to respectively as *indirect georeferencing* and *direct georeferencing* (Toth and Jóźków, 2016).

The inclusion of GNSS and INS allows for direct determination of platform trajectory and thus sensor orientation without the immediate need for ground control. The default processing pipeline in modern kinematic mapping is shown in Fig. 1 (cf. Kager, 2004, or more recently Glira et al., 2016). This is generally a multi-step procedure: First, the trajectory is determined by fusion of GNSS and INS via Kalman filtering (Colomina, 2015; Fengguang et al., 2017). If the trajectory is sufficiently accurate, it can be used directly to obtain 3D point clouds from the measurements of the imaging sensors. Alternatively, the GNSS/INS integration may be followed by an adjustment step: Redundancy in the imaging sensor observations is used together with the trajectory information in an *integrated sensor orientation* to obtain (i) corrections to the trajectory, (ii) sensor orientation and calibration, and (iii) the desired 3D models, e.g., point clouds. In contrast to this two-step approach, recent methods combine imaging sensor-based adjustment with GNSS/INS integration in a single estimation procedure (see Fig. 2).

2.1. Trajectory estimation with GNSS and INS

The fundamental technologies in modern navigation applications are satellite navigation and inertial navigation, which synergize well due to GNSS providing low-frequency absolute measurements and INS providing high-frequency relative measurements (cf. Farrell, 2008). However, redundancy in GNSS/INS integration is low resulting in reduced observability (Tang et al., 2009) and overly optimistic error measures.

The inertial sensors (gyroscope and accelerometer) are usually combined into a single measurement device, the inertial measurement unit (IMU). In some cases, additional sensors such as a magnetometer are included. The INS consists of the inertial sensors, as well as

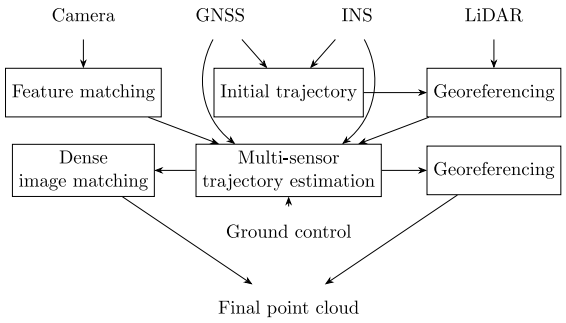

Fig. 2. Holistic kinematic mapping processing pipeline with sensor-level error modelling (see Section 2.3).

hardware and algorithms for computing a dead reckoning navigation solution (Groves, 2013). The inertial sensors provide high frequency measurements of proper acceleration and angular velocity, typically with 100–400 Hz (see also Section 4.2, Eq. (4.11)). From these, relative position and orientation may be derived through integration, but errors (e.g., noise and bias) in the inertial measurements cause large drifts over time. Information from other sensors, most commonly GNSS, mitigates this by providing lower frequency yet more stable information, allowing for in-run calibration of the inertial sensor errors. Due to its high measurement frequency and ability to track both position and orientation, the IMU is often seen as the primary navigation sensor with other sensors serving as a way to eliminate or reduce its drift. Vehicle dynamics and GNSS reception determine whether INS errors are observable and can be corrected for. As an extreme example, without any movement or with constant velocity, bias in the accelerometer cannot be completely distinguished from gravity. Generally, trajectory quality is limited by the INS quality as well as satellite visibility (Skaloud et al., 2010; Lesjak et al., 2015), and thus may exhibit large drifts during GNSS outages. Other sensors such as distance measuring instruments, or exterior orientations from an indirect georeferencing, can provide additional redundancy and improve the trajectory estimate.

2.2. Sensor orientation and trajectory adjustment

The accuracy of the acquired point cloud depends strongly on the quality of both GNSS/INS trajectory and sensor calibration parameters (Glennie, 2007; Vallet et al., 2020). For frame cameras, indirect sensor orientation is possible and trajectory information is not strictly needed but is in practice commonly used for aerotriangulation and as initial solution or constraint in a *bundle adjustment* (Triggs et al., 2000). Moderate trajectory errors can thereby be compensated, although only at the image acquisition times. On the other hand, survey-grade laser scanners provide high frequency point measurements (>10 kHz, typically even >100 kHz, cf. Pirotti, 2013; Vallet et al., 2020). Due to the high frequency and sequential nature of these point measurements, indirect georeferencing is not applicable and direct or integrated georeferencing of the laser points requires highly accurate trajectory information. Trajectory errors become visible as discrepancies in the point clouds where multiple scans observe the same physical area. A subsequent *strip adjustment* uses overlaps to reduce trajectory errors and for in-run self-calibration. In order to correct errors on the trajectory level, certain assumptions about the trajectory must be made: Trajectory errors are usually assumed to be due to slowly varying GNSS errors or uncompensated inertial sensor drift and thus of low frequency (<1 Hz) while the higher frequency components of the trajectory are comparatively accurate. Practically, corrections are either tied to the flight geometry, as fixed offsets per strip, or modelled as slowly time-varying (Kager, 2004; Glira et al., 2016). Since trajectory errors can vary strongly also within strips, such adjustment in practice often requires highly flexible correction models and is therefore at risk of overfitting, resulting in a global deformation of the point clouds. This motivates the modelling of errors at the sensor level by including laser scanner and camera measurements directly in the trajectory estimation, together with GNSS, INS and possibly other sensors.

2.3. Trajectory estimation with GNSS, INS and imaging sensors

This shift from error modelling at the trajectory level to error modelling at the sensor level is not new (Colomina, 2015), although it has accelerated recently due to a convergence between the robotics, autonomous navigation and remote sensing communities. This has lead to significant advances especially in multi-sensor integration.

It is worth noting that there are some differences in the requirements on the platform trajectory depending on the final application. For navigation purposes, e.g., car navigation or machine guidance, one is usually interested in the smoothed vehicle trajectory. For surveying, i.e., georeferencing of sensor measurements, the trajectory of interest is that of the imaging sensor itself, which may or may not be rigid with respect to the vehicle frame. For this purpose, any vibrations are not to be considered as noise and therefore are not to be filtered out, as these vibrations do effect the measurements. Similarly, the accuracy requirements for georeferencing are high. Nominal laser ranging accuracy is around 2 cm (Glennie, 2007), but for sufficiently smooth terrain deviations in the millimeter range have been reported (Mandlburger et al., 2020). Thus, trajectory accuracy is often the limiting factor. Due to the high measurement range in airborne remote sensing, errors in the orientation (be it trajectory errors or misalignment between scanner and INS), dominate.

In the following, we will discuss the sensors of specific interest in a mapping context, i.e., those that are classically included in a kinematic mapping system: GNSS, INS, laser scanner and camera. Apart from GNSS, we consider only autonomous sensors which are wholly located on the platform and do not rely on data from external sources (such as satellites or radio beacons). *Proprioceptive sensors*, such as inertial sensors or wheel odometry, measure an internal state. *Exteroceptive sensors*, such as cameras or laser scanners, measure certain properties of the environment. Each of these specific sensors is capable of standalone localization: GNSS provides position, while strap-down inertial navigation provides both relative position and orientation. Standard photogrammetric techniques allow for recovery of relative transformation between images, known in robotics as visual odometry. Similarly, LiDAR odometry is the determination of position and orientation from successively captured point clouds. In contrast to trajectory estimation, *odometry* generally focuses on an incremental computation of position and orientation using successively acquired data.

The question remains how to best integrate these measurements in order to obtain optimal estimates for the trajectory. The sensors' *coupling* describes how the measurements are used and interact with each other. Loose coupling refers to error modelling at the trajectory level: position and/or orientation estimates from different sensors are combined into an integrated trajectory estimate. In tight coupling, errors are modelled at the sensor level; i.e., measurements are combined in a raw form and only one trajectory estimate is produced. This terminology is widespread and in longstanding use (Greenspan, 1996), but is not fully standardized. For the most commonly used definitions, which are also adopted here, see Groves (2013).

For integration of more than two systems, each component can be coupled loosely or tightly to another. Coupling can also be a hierarchical process, with two systems being tightly coupled to produce an intermediate result, which is then further improved with measurements from a different system. Generally, looser coupling allows for modularization of the individual systems which makes implementation less complex and possibly computationally more efficient. This comes at the cost of reducing the information available to the trajectory estimation algorithm. The pre-processing step may also produce synthetic measurements with properties that are incompatible with the assumptions

Table 1
Multi-sensor mounting parameters.

Trajectory	Position	$\boldsymbol{x}^e_{eb}(t) \in \mathbb{R}^3$
	Orientation	$\boldsymbol{R}^b_e(t) \in \mathrm{SO}(3)$
GNSS	Lever arm	$\boldsymbol{x}^b_{bg}$
Laser scanner	Lever arm	$\boldsymbol{x}^b_{bs} \in \mathbb{R}^3$
	Boresight alignment	$\boldsymbol{R}^s_b \in \mathrm{SO}(3)$
Camera	Lever arm	$\boldsymbol{x}^b_{bc} \in \mathbb{R}^3$
	Boresight alignment	$\boldsymbol{R}^c_b \in \mathrm{SO}(3)$

of the integration method, e.g., different noise distributions. In contrast, tight coupling takes all measurements into account, thereby enabling more accurate – but also more complex – modelling, both with respect to the functional and the stochastic model. Sensors may also simply aid in the estimation process without directly impacting the trajectory estimate. This is often done to improve the GNSS solution (Groves, 2013), e.g., by using measurements from camera or laser scanner to identify non-line-of-sight effects (Wen et al., 2019a; Wen, 2020).

2.4. Multi-sensor platform

The considered MSS consists of multiple sensors which are mounted on a carrier platform (Fig. 3). Trajectory position and orientation are generally referenced to a coordinate system which is rigid with respect to the imaging and inertial sensors. Stabilized mounts can be used to stabilize the sensor system with respect to the platform, leading to a dynamic (but known) rotation between the sensors and the platform. Here, we assume a rigid mounting to simplify the presentation and interpret the trajectory as position and orientation of the inertial sensor frame. In other words, the inertial sensor frame is assumed coincident with the body frame.

The geometric relation of the imaging sensors' frames to the inertial sensors' frame (the *sensor mounting*) is required to obtain the sensors *exterior orientation*. Additionally, depending on the sensor, various internal calibration parameters (i.e., the *interior orientation*) are of interest. Table 1 recaps the geometric relations and mounting parameters commonly occurring in such a multi-sensor system setup.

Definition of coordinate systems

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 $\boldsymbol{B}^\alpha = \left(b^\alpha_1, b^\alpha_2, b^\alpha_3\right) \in \mathrm{SO}(3)$ of a corresponding three-dimensional Euclidean frame with origin $\boldsymbol{o}^\alpha \in \mathbb{R}^3$, where $\mathbb{R}^3$ is itself interpreted as Euclidean frame with canonical basis and origin $\boldsymbol{o} = \boldsymbol{0}$. 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 α w.r.t. frame β, resolved in frame γ,

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

The following coordinate systems are used:

- e … earth-centered, earth-fixed rotational coordinate system,
- i … earth-centered inertial coordinate system,
- b … body coordinate system, assumed identical to inertial sensors' coordinate system,
- s … laser scanner coordinate system,
- c … camera coordinate system,
- g … GNSS antenna coordinate system.

The platform trajectory is defined as position and orientation of the body coordinate system with respect to an earth-fixed coordinate system

$$\boldsymbol{x}^e_{eb}(t) \in \mathbb{R}^3, \quad \boldsymbol{R}^b_e(t) \in \mathrm{SO}(3) \tag{2.1}$$

in a specific time interval $[t_a, t_b] \ni t$.

Rotations in 3D are fundamentally linear maps and can be represented as matrices $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ where $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$. This is impractical in an estimation setting, as it involves nine parameters with an orthogonality constraint. Thus, a suitable parametrization for the rotational motion (cf. Trainelli and Croce, 2004) has to be chosen.

The classical and still widely used option for parametrizing rotations is to use Euler angles, where a rotation is described by three rotations around fixed axes (L'Afflitto, 2017). The axis-angle representation, another three-parameter representation, describes a rotation by a normalized axis of rotation and an angle of rotation. However, all three-parameter rotation parametrizations are subject to singularities (Stuelpnagel, 2006). Quaternions are a singularity-free 4-parameter representation with the additional constraint of unit norm and are frequently used in modern filtering and optimization methods (Schmidt et al., 2001; Al-Jlailaty and Mansour, 2021). In many robotics or computer vision applications, 4D *homogeneous coordinates* (Hartley and Zisserman, 2003) are used for describing 3D points and their transformations, due to computational and practical advantages (Barfoot, 2017).

3. Estimation framework

3.1. Problem setting and mathematical model

The general setting for trajectory estimation is that of a time-dependent parameter estimation problem (Jazwinski, 1970; van den Bos, 2007) for an underlying dynamical[2] system model.

In this section, the general probabilistic trajectory estimation model is described. This formulation is popular especially in SLAM (Cadena et al., 2016), but also describes classic GNSS/INS integration as well as strip and bundle adjustment. As customary in the robotics literature, we formulate the probabilistic problem of trajectory estimation as a maximum a-posteriori problem (Cadena et al., 2016; Barfoot, 2017).

The trajectory evolution and sensor measurements are described by a state-space model, composed of two parts. The *process model*

$$\dot{\boldsymbol{x}}(t) = f(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{v}(t), \quad t \in [0, T], \tag{3.1}$$

is a stochastic differential equation which models the evolution of the system state $\boldsymbol{x}(t)$. The state includes position, attitude and other time-varying quantities and is influenced by an extrinsic input $\boldsymbol{u}(t)$. The dynamic model for the system state is disturbed by *process noise* $\boldsymbol{v}(t)$. Often, this does not mean the state $\boldsymbol{x}(t)$ is random, but rather the process noise is used to describe uncertainties in the model itself.

The various sensors make measurements $\boldsymbol{z}_k$ which are distorted by random noise $\boldsymbol{w}_k$. This measurement process is described by *the measurement model*

$$\boldsymbol{z}_k = h_k(\boldsymbol{x}(t_k)) + \boldsymbol{w}_k, \quad t_k \in [0, T], \quad 1 \le k \le n. \tag{3.2}$$

Strictly speaking, the noise $\boldsymbol{w}_k$ describes the actual measurement uncertainty but may in practice also serve to probabilistically represent systematic uncertainty associated with the modelling of the measurement process.

Since the measurements are available at certain times t_k, a straightforward way of discretizing Eq. (3.1) is to consider the discrete subset of states $\boldsymbol{x}_k := \boldsymbol{x}(t_k)$. This is the classical approach employed in the Kalman filter and related methods, replacing Eq. (3.1) with a set of discrete difference equations

$$\boldsymbol{x}_k = F_k(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \boldsymbol{v}_k, \tag{3.3}$$

where $\boldsymbol{v}_k$ is (now discrete) random noise. The state transition F_k is in general only an approximation to the exact solution of the initial value problem

$$\dot{\boldsymbol{x}}(t) = f(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \boldsymbol{x}(t_{k-1}) = \boldsymbol{x}_{k-1}. \tag{3.4}$$

[2] The term *dynamic* refers to time-dependent behaviour, whereas *kinematic* refers specifically to the motion of one or multiple objects.

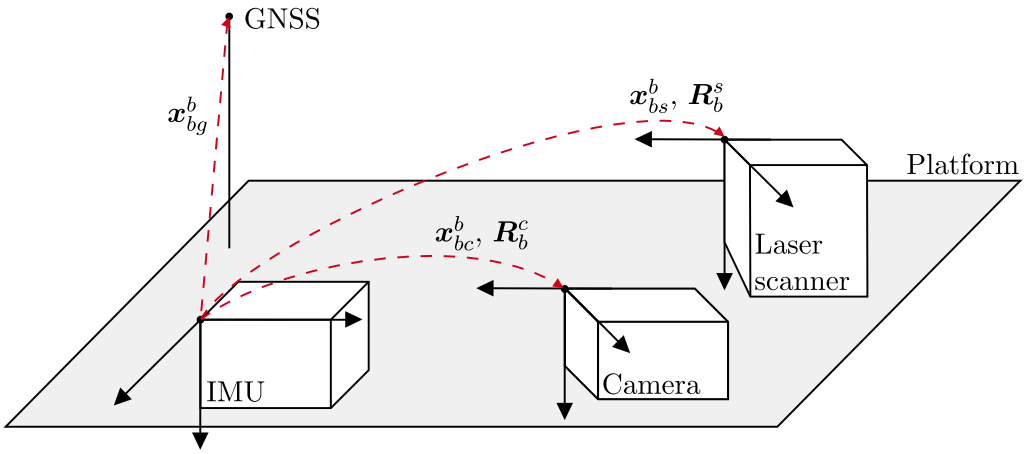

Fig. 3. Multi-sensor platform featuring GNSS, INS, laser scanner & camera. The inertial and imaging sensors are depicted as boxes, while the GNSS is represented as an antenna.

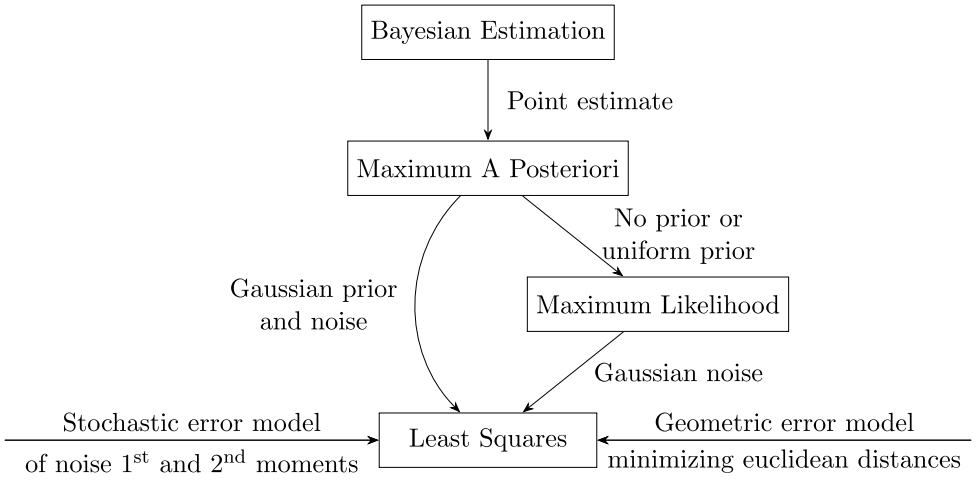

Fig. 4. Taxonomy of parameter estimation methods.

3.2. Parameter estimation

The various estimation methods differ in their assumptions on the stochastic properties of the process and measurement noises. Below follows a short recap of parameter estimation methods in a general setting and their application to the trajectory estimation problem.

Consider the goal of estimating unknown parameters $\boldsymbol{x}$ from n noisy measurements $\boldsymbol{z}$. In a *full Bayesian* approach, the conditional probability density function $p(\boldsymbol{z}|\boldsymbol{x})$ is assumed to be known. This allows inference based on the posterior $p(\boldsymbol{x}|\boldsymbol{z})$, i.e., the probability distribution of the state given the measurements. Using Bayes theorem, the posterior density is written as

$$p(\boldsymbol{x}|\boldsymbol{z}) = \frac{p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{z})}, \tag{3.5}$$

where the a priori density $p(\boldsymbol{x})$ contains any prior knowledge (belief) about $\boldsymbol{x}$.

Computing the whole posterior density (i.e., a *fully* Bayesian approach) is often not feasible for complex problems. The *maximum a-posteriori* (MAP) estimate maximizes the posterior

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, p(\boldsymbol{x}|\boldsymbol{z}) = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \frac{p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{z})} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x}). \tag{3.6}$$

MAP estimation produces the parameters *most likely* given the measurements $\boldsymbol{z}$, i.e., the mode of the posterior distribution, which in general differs from the mean. In that case, the MAP estimator is biased. If the prior density $p(\boldsymbol{x})$ is constant, MAP reduces to classical *maximum likelihood* (ML) estimation. Under the additional assumption of Gaussian noise, MAP and ML estimates are identical to the nonlinear least squares (NLS) estimator. For independent Gaussian noise $\boldsymbol{w}_k, \boldsymbol{v}_k$ with zero mean and variances $\mathbb{V}(\boldsymbol{w}_k), \mathbb{V}(\boldsymbol{v}_k)$, the NLS estimator for Eqs. (3.2)–(3.3) is given by

$$\begin{aligned}\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} &\sum_{k=1}^{n} (\boldsymbol{z}_k - h_k(\boldsymbol{x}_k))^T\, \mathbb{V}(\boldsymbol{w}_k)^{-1}\, (\boldsymbol{z}_k - h_k(\boldsymbol{x}_k)) \\ &+ \sum_{k=1}^{n} (\boldsymbol{x}_k - F_k(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k))^T\, \mathbb{V}(\boldsymbol{v}_k)^{-1}\, (\boldsymbol{x}_k - F_k(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k)).\end{aligned} \tag{3.7}$$

The least squares estimator is a linearized version of the best linear unbiased estimator (BLUE), which requires only knowledge of the noises' first and second moments (see Fig. 4). Even without probabilistic assumptions, NLS can be interpreted in terms of minimizing a geometric distance. Note that this approach does not require an explicit distinction between process and measurement model, as all equations contribute to the estimation in the same way.

In SLAM, the batch MAP problem is often described using *factor-graphs* (Kschischang et al., 2001; Grisetti et al., 2010), a graphical formalism for modelling *factored* functions, such as probability densities (Dellaert and Kaess, 2017). The factor-graph (cf. Fig. 5) can be used to derive efficient solvers for the linear system occurring in the NLS optimization.

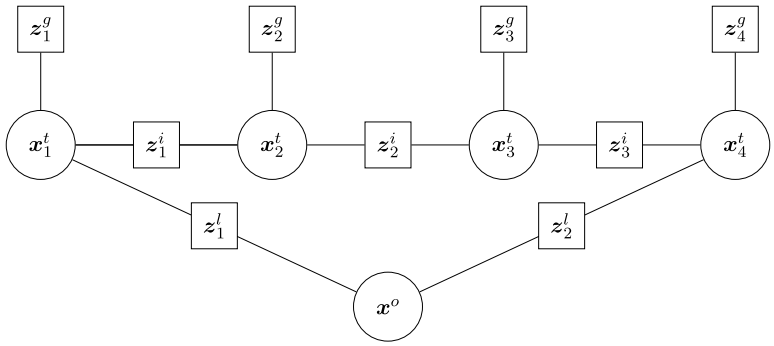

Fig. 5. Example of a factor graph for GNSS/INS/LiDAR integration. The trajectory and object parameters $\boldsymbol{x}_\star^t, \boldsymbol{x}^o$ are represented by circles, the GNSS, INS and LiDAR measurements $\boldsymbol{z}_\star^g, \boldsymbol{z}_\star^i, \boldsymbol{z}_\star^l$ by rectangles. A measurement (or *factor*) is conditionally independent of all other measurements given the parameters it is connected to.

3.3. Filtering and smoothing

Estimation theory distinguishes between *filtering*, where the state is estimated given information at the current and previous points in time and *smoothing*, which makes use of both past and future measurements. For applications where trajectory information is required in real-time, filtering methods are the obvious choice. Otherwise, the smoothing solution is superior, although it comes with a higher computational cost.

Randomness enters the system in the form of process noise $\boldsymbol{v}_k$, and the measurement noise $\boldsymbol{w}_k$. If the noise terms $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ are independent in the discrete model

$$\begin{aligned} \boldsymbol{x}_k &= F_k(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \boldsymbol{v}_k, \\ \boldsymbol{z}_k &= h_k(\boldsymbol{x}_k) + \boldsymbol{w}_k, \end{aligned} \tag{3.8}$$

then the probability density functions of the states and measurements factorize as

$$p(\boldsymbol{z}|\boldsymbol{x}) = \prod_{k=1}^{n} p(\boldsymbol{z}_k|\boldsymbol{x}_k), \quad p(\boldsymbol{x}) = \prod_{k=1}^{n} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})\ p(\boldsymbol{x}_0),$$

and their posterior density is given by

$$p(\boldsymbol{x}|\boldsymbol{z}) \propto p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x}) = \prod_{k=1}^{n} p(\boldsymbol{z}_k|\boldsymbol{x}_k) \prod_{k=1}^{n} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})\ p(\boldsymbol{x}_0). \tag{3.9}$$

The model is *Markovian*, which means the current state is conditionally independent of past measurements given the prior state

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_{1:k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}),$$

and the current measurement is conditionally independent of past measurements given the current state

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{z}_{1:k-1}) = p(\boldsymbol{z}_k|\boldsymbol{x}_k).$$

The MAP solution to this problem is usually called the *smoother*. In contrast, if one is interested only in the most recent state $\boldsymbol{x}_k$ given all previous and current measurements $\boldsymbol{z}_{1:k}$, a recursive *filtering* solution can be derived.

This *Bayesian filter* (see e.g., Särkkä, 2013) consists of a *prediction* and an *update* step. First, the predictive density $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})$ is computed from the state transition $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ and previous state $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$ as

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})\, p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})\, \mathrm{d}\boldsymbol{x}_{k-1}, \tag{3.10}$$

and the updated posterior density $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$, derived using Bayes rule, is then given by

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \propto p(\boldsymbol{z}_k|\boldsymbol{x}_k)\, p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}). \tag{3.11}$$

The *particle filter* (Del Moral, 1996) is a sequential Monte Carlo method where random samples (the *particles*) are used to represent the probability densities occurring above. Assuming a linear model and uncorrelated Gaussian noise, all densities are Gaussian and thus fully described by their mean and covariance. This is the basis of the classic Kalman filter (Kalman, 1960), which follows directly from Eqs. (3.10)–(3.11) under the Gaussian assumptions. In comparison, the *information filter* (cf. Anderson and Moore, 2005, Ch. 6) does not use the covariance matrix to describe uncertainty but rather its inverse, the information matrix. While algebraically equivalent, the information filter is computationally more efficient if the number of measurements is large compared to the number of state variables.

In the case of a linear model with Gaussian noise, the smoothing solution is equivalent to a classic *batch*[3] least-squares solution. The resulting linear system is a block-tridiagonal system, which can be efficiently solved using sparse methods. Conceptually, sparse tridiagonal solvers rely on a forward and backward pass, similar to forward–backward smoothers (cf. Aravkin et al., 2017). The *extended Kalman filter* (EKF) is a generalization of the linear Kalman filter to nonlinear models, which uses a first-order approximation of the process and measurement model for covariance propagation. Extensions to the EKF are the multi-state constraint Kalman filter (Mourikis and Roumeliotis, 2007) as well as the unscented Kalman filter, which uses deterministically chosen sample points and can be more accurate especially for highly nonlinear functions (Wan and Van Der Merwe, 2000).

Multiple Kalman filters are sometimes used in a chained (or *cascaded*) fashion. An example of this is classical loosely coupled GNSS/INS integration, where the GNSS position/velocity estimates are computed by Kalman filter and then passed into another Kalman filter together with the inertial measurements. However, standard Kalman filters assume the measurements to be free of any time correlation – which is not the case for its output – leading to overly optimistic covariance estimates (Crespillo et al., 2020).

The relation between different filtering methods is depicted in Fig. 6. For an in-depth treatment of Bayesian filtering and smoothing refer to Särkkä (2013).

3.4. Linearization and marginalization

The classic delineation between filtering and smoothing has become less useful, as some methods incorporate aspects of both filtering and smoothing, e.g., a sliding window filter that employs a smoother within each window. In this context, a distinction is often made between filter-based and optimization-based methods (cf. Chen et al., 2018a). This is a useful characterization, but more specifically the defining characteristic is whether an iterative optimization (i.e., re-linearization) is performed and if so, which states are involved.

Filter-based methods of course include classical filtering techniques, such as recursive least squares filters, Kalman filters, complementary

[3] The term batch is used here to signify the estimation of all/many states (the batch) at once, in contrast to recursive (or sequential) solutions.

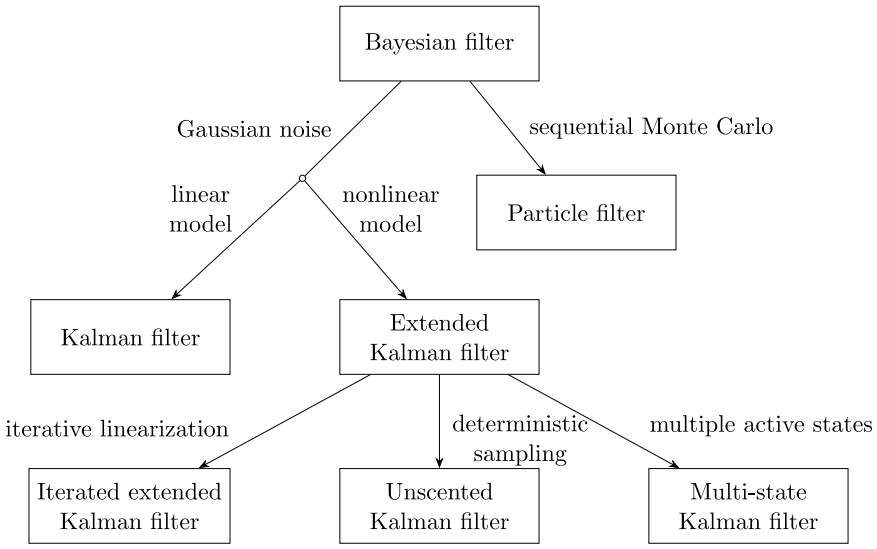

Fig. 6. Taxonomy of filtering methods.

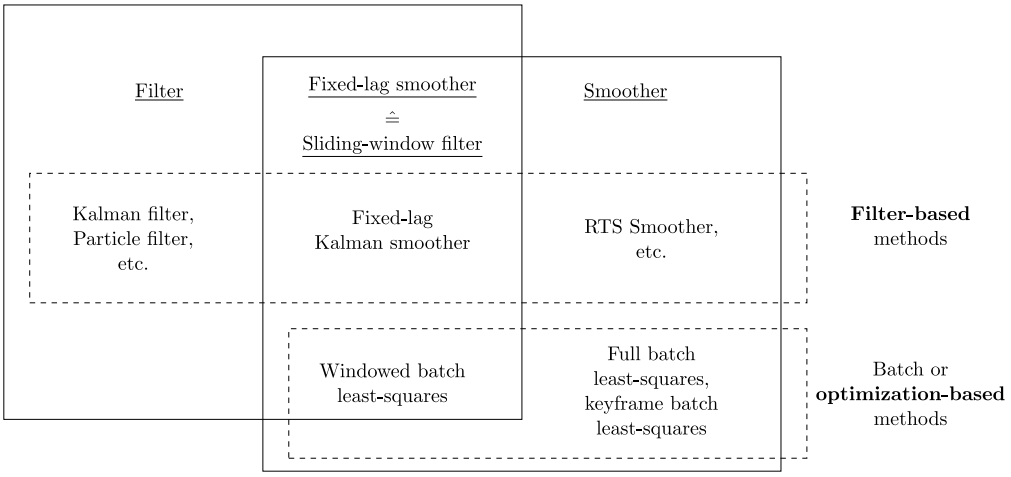

Fig. 7. Filter-based and optimization-based methods. While many filter-based methods also employ optimization in some form, only the methods derived from the full nonlinear MAP problem and incorporating an iterative nonlinear optimization simultaneously over a large set of states are generally referred to as *optimization-based*.

filters and particle filters, and run in an incremental fashion. For trajectory estimation, particle filters (Montemerlo et al., 2002) or variations of the multi-state constraint Kalman filter developed in Mourikis and Roumeliotis (2007) are popular.

Smoothing methods can be constructed from filtering methods, e.g., by combining a forward and a backward filter pass (Särkkä, 2013). Methods based on batch MAP estimation use nonlinear optimization to compute *global* estimates of all states (the *full batch*) and are thus smoothers (Dellaert and Kaess, 2006). An efficient simplification is to optimize only over a subset of states and marginalizing out (Chang et al., 2019) or discarding (Lupton and Sukkarieh, 2012) all others. *Sliding-window* methods such as Huang et al. (2011) and Shan et al. (2020) use a number of consecutive prior states instead of only the current state and incrementally add new and remove old states. On the other hand, *keyframe*-based methods (Leutenegger et al., 2015) keep the problem size tractable by identifying a small set of important states. These partial MAP estimates are incrementally incorporated into a global solution. Both window and keyframe methods are a simplification of the full batch MAP estimate and in general do not result in the same estimate. As in practice these methods are all based on the assumption of Gaussian noise and therefore reduce to a nonlinear least-squares estimation, this approach will be referred to as batch NLS. Notably, MAP estimation with non-Gaussian noise can still be done with least-squares methodology in a statistically correct way (Rosen et al., 2013), although this has not yet seen much use.

Optimization-based usually refers to methods using nonlinear optimization techniques to obtain an MAP estimate (i.e., a solution to the maximization problem Eq. (3.6)) either for the full problem or for a subset of states. The number of states which are optimized simultaneously is referred to as the batch size. The use of the term 'optimization' is somewhat ambiguous in this context, as many filter-based methods can be interpreted as sequential optimization. An example of this is the iterative Kalman filter, which can be seen as a recursive application of the Gauss–Newton method (Bell and Cathey, 1993). Thus, 'optimization-based' emphasizes simultaneous optimization over multiple states, in contrast to the sequential nature of filtering algorithms (see Fig. 7).

For a linear model with Gaussian noise, the Kalman filter is equivalent to the MAP estimate for the current state, given all past measurements. Similarly, the Rauch–Tung–Striebel (RTS) smoother is then equivalent to the full MAP estimate (Särkkä, 2013). The defining difference within classic filter-based methods and optimization-based methods can be traced to how the linearization of the nonlinear model is performed (Fleps et al., 2011; Hesch et al., 2014). An extended Kalman filter (EKF) linearizes only once. Iterative Kalman filters linearize iteratively, but still only for the current state as past states

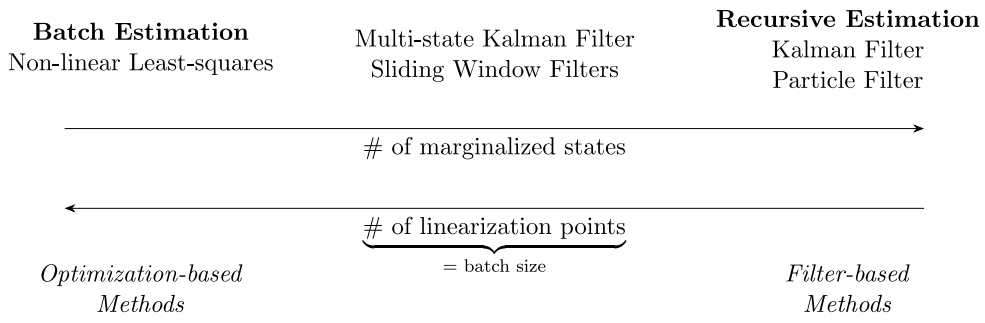

Fig. 8. Trade-off between marginalization and linearization: As more and more states are marginalized, their linearization points are fixed, and thus fewer variables are available for (re-)linearization.

are marginalized out and the filters' belief about past states is never updated. On the other hand, batch least-squares estimation based on nonlinear optimization successively re-linearizes all states to obtain an improved estimate. In this way, the marginalization of states reduces computational effort but also reduces accuracy by introducing linearization errors. Fig. 8 highlights this trade-off: the more states are marginalized, the fewer states are available for linearization.

Whether the linearization error presents a problem in practice depends on the degree of nonlinearity of the model and the accuracy of the initial linearization point. For GNSS/INS integration, error-state formulations perform better than total-state formulations (Madyastha et al., 2011) due to mitigating the nonlinearity in the state propagation by modelling the error of the state instead of the state itself. Multi-state Kalman filters or sliding window filters implicitly use partial re-linearization and will also reduce linearization errors. All else being equal, a batch MAP solution can be expected to be at least as accurate as the corresponding filtering solution and can be more efficient depending on the application (Dellaert and Kaess, 2017). In a comparison of EKF and batch NLS, Wen et al. (2019b) demonstrate better accuracy of the latter, albeit at the cost of computational efficiency. Strasdat et al. (2012) show that bundle adjustment is more accurate and efficient than filtering for visual SLAM. Even compared to a particle filter, which is capable of representing non-unimodal non-Gaussian distributions, sliding window batch NLS can achieve higher accuracy (Wilbers et al., 2019). The benefits of the batch NLS approach in comparison to an EKF approach are summarized as

- at least locally optimal solution of the MAP problem,
- straightforward modelling of time-constant parameters,
- better computational performance for large numbers of observations w.r.t. states (Strasdat et al., 2012),
- more accurate estimates for highly nonlinear models (Wen et al., 2019b).

Since the batch NLS approach involves solving potentially large systems of equations, these benefits come at the expense of performance. Specifically, batch NLS methods often exhibit

- superlinearly growing computational complexity as the number of states increases,
- costly recovery of the posterior parameter covariance, and
- globally optimal solution is still not guaranteed for nonlinear nonconvex problems.

With the increase of available computational resource and the development of more efficient numerical methods, the first two drawbacks become less relevant. Recently, stochastic optimization techniques developed for machine learning have been successfully applied to general nonlinear least squares optimization (Le et al., 2021; Huang et al., 2021a) with promising performance improvements. Similarly, the adaptation of convex optimization methods, such as proximal methods (Eriksson et al., 2016; Fan and Murphey, 2022) or the alternating direction method of multipliers (Boyd, 2010; Ramamurthy et al., 2017), to the (generally nonconvex) nonlinear least squares problem allows efficient optimization in distributed computing environments. These developments allow for the use of the batch approach for increasingly large and complex problems.

4. Modelling framework

Trajectory estimation requires modelling of two somewhat separate aspects: Modelling of the sensor measurements and modelling of the trajectory itself. While the classical methods represent the trajectory (as well as other time-varying quantities) as a discrete time-series of state variables, some newer formulations use a continuous representation.

4.1. Discrete and continuous trajectories

The mathematical trajectory definition Eq. (2.1) is not directly realizable: The trajectory must be discretized somehow, as in practice it obviously cannot be stored for all $t \in [t_a, t_b]$. Trajectory data is commonly available as timestamped discrete samples. For georeferencing, the trajectory is required at least for the time of each LiDAR shot or camera exposure and is accordingly interpolated to obtain position and orientation for all measurements. The necessary frequency thus depends on the sensors used (i.e., measurement frequency) but also on the vehicle dynamics. High frequency measurements can be accurately georeferenced from a low-frequency trajectory by interpolation if the movement is sufficiently smooth. High vehicle dynamics require the trajectory to be more finely resolved in time.

Generally, it is necessary to obtain trajectory estimates at the imaging sensor sample times. If the number of such measurements is large, it may be unfeasible to explicitly estimate the trajectory at all these times, requiring that the trajectory can somehow be represented by a relatively small number of parameters. For this, there are two options: compute trajectory estimates at fixed points in time and use interpolation to derive the required values in between or use an explicit continuous-time representation that can be evaluated whenever required. While interpolation of the position is straightforward, special care needs to be taken for the orientation. In contrast to the position, the 3D rotations do not make up a Euclidean vector space and standard vector-space techniques do not properly account for the underlying mathematical structure (Zefran and Kumar, 1998; Haarbach et al., 2018). When directly applying standard interpolation to e.g., the Euler angle parametrization, the interpolation suffers from singularities and the interpolated values depend on the choice of coordinate systems.

A continuous representation of an arbitrary function $g : [t_a, t_b] \to \mathbb{R}$ can be constructed by linearly combining suitable basis functions $B_i : [t_a, t_b] \to \mathbb{R}$

$$g(t) = \sum_{i=0}^{n} c_i B_i(t), \quad t \in [t_a, t_b], \tag{4.1}$$

given a set of $n+1$ parameters $c_i \in \mathbb{R}$, $0 \leq i \leq n$. The parameters are of course still discrete, but the resulting function g is continuous. Often, B-Splines are used as basis functions (Furgale et al., 2012). As with interpolation, special care has to be taken for the orientations, e.g., by using cumulative B-Splines. The cumulative B-Spline introduced in Kim et al. (1995) is a widely used method (cf. Lovegrove (2013) and Li et al. (2020)) to generate arbitrary order quaternion curves which are invariant to coordinate system changes (Sommer et al., 2016). Such a basis function representation for position and orientation can be written as

$$\boldsymbol{x}^e_{eb}(t) = \sum_{i=0}^{n} \boldsymbol{x}_i B_i(t), \qquad \boldsymbol{x}_i \in \mathbb{R}^3, \tag{4.2}$$

$$\boldsymbol{R}^b_e(t) = \boldsymbol{R}_0 \prod_{i=1}^{n} \exp(\log(\boldsymbol{R}_{i-1}^T \boldsymbol{R}_i)\tilde{B}_i(t)), \qquad \boldsymbol{R}_i \in \mathrm{SO}(3), \tag{4.3}$$

where $\tilde{B}_i(t) = \sum_{j=i}^{n} B_j(t)$ are the *cumulative basis functions* and $\exp, \log$ are the exponential and logarithmic maps[4] in SO(3). The locality of classical B-Splines carries over to the cumulative formulation: when B-splines of order k are chosen as basis functions B_i, all but $k+1$ terms in the sum and product above vanish.

A continuous-time representation can also be achieved by employing a stochastic formulation based on Gaussian process regression with a suitable prior (Tong et al., 2013). The Gaussian process is defined by the a-priori mean μ and covariance kernel[5] K (see also Rasmussen and Williams, 2006)

$$f(t) \sim \mathcal{GP}(\mu(t), K(t, t')). \tag{4.4}$$

Given an estimate of the trajectory at given times $t_0, \ldots, t_n$, function values at an arbitrary time t can then be recovered in a way similar to Gaussian process regression or Kriging

$$f(t) = \mu(t) + \begin{pmatrix} K(t,t_0) & \ldots & K(t,t_n) \end{pmatrix} \times \begin{pmatrix} K(t_0,t_0) & \ldots & K(t_0,t_n) \\ \vdots & \ddots & \\ K(t_n,t_0) & \ldots & K(t_n,t_n) \end{pmatrix}^{-1} \begin{pmatrix} \mu(t_0) - f(t_0) \\ \vdots \\ \mu(t_n) - f(t_n) \end{pmatrix}. \tag{4.5}$$

To extend this approach for matrix Lie groups (e.g., the rotational component of the trajectory), the Gaussian process is defined locally in the Lie group's tangent space (Dong et al., 2018). Note that while the practical implementation differs, the two approaches are somewhat analogous as spline interpolation can be interpreted as type of Gaussian process regression for a specific covariance kernel (Kimeldorf and Wahba, 1970).

Trajectory priors

SLAM methods often rely only on visual information which can be sparse depending on the environmental features. This motivates making certain assumptions on the platform trajectory to constrain the trajectory towards realistic values, which can be done either explicitly or implicitly. In filter-based methods, such assumptions are modelled by choosing a suitable process model. Similarly, the Gaussian-process-based estimation methods model the trajectory as a Gaussian process with a suitable prior distribution. These stochastic priors only steer the solution towards the prior assumptions, but do allow deviations from the prior based on the actual measurements. On the other hand, parametrized continuous-time methods implicitly impose a sort of 'deterministic prior' by the choice of the basis functions. Compared to stochastic priors, the choice of basis functions is a hard constraint as the solution inherently cannot deviate from this representation.

4.2. Sensors and measurements

The multi-sensor system consists of multiple sensors providing measurements of different physical or geometric quantities. Measurements from different sources can be combined in different ways, which is described by the coupling of the sensors. Below follows a discussion of the common sensors in kinematic mapping, GNSS, INS, laser scanner and camera, and their respective measurements.

Global navigation satellite system

GNSS provides global positioning using passive ranging by radio signals transmitted from satellites. This makes it an integral component of most georeferencing solution, as it provides stand-alone globally georeferenced position measurements. Multi-antenna GNSS configurations may be used for orientation determination (Teunissen and Montenbruck, 2017), although in this work only GNSS positioning with a single antenna is discussed. The primary measurements supplied by GNSS through the standardized receiver independent exchange format (RINEX) are pseudorange, carrier phase and Doppler measurements. For high-precision kinematic applications, carrier-phase differential GNSS provides up to cm-level accuracy in post-processing mode (Hofmann-Wellenhof et al., 2008), given that the integer ambiguities are correctly resolved.

Loosely coupled integration refers to the use of separately computed GNSS solution in the GNSS/INS integration. This standalone GNSS solution is usually computed by an extended Kalman filter (Teunissen and Montenbruck, 2017), which estimates the antenna position and velocity

$$\widetilde{\boldsymbol{x}}^e_{eg}(t), \ \widetilde{\boldsymbol{v}}^e_{eg}(t), \quad t \in \{t_0, \ldots, t_n\}, \tag{4.6}$$

as well as acceleration and other model parameters such as receiver clock error. In loose coupling, the GNSS-derived position and velocity is described by the measurement equation[6]

$$\widetilde{\boldsymbol{x}}^e_{eg}(t) = \boldsymbol{x}^e_{eb}(t) + \boldsymbol{R}^e_b(t)\boldsymbol{x}^b_{bg} + \boldsymbol{w}_x(t), \tag{4.7}$$

$$\widetilde{\boldsymbol{v}}^e_{eg}(t) = \boldsymbol{v}^e_{eb}(t) + \dot{\boldsymbol{R}}^e_b(t)\boldsymbol{x}^b_{bg} + \boldsymbol{w}_v(t). \tag{4.8}$$

The noise terms $\boldsymbol{w}_x, \boldsymbol{w}_v$ describe errors in the processed GNSS solution and are made up of the original measurement errors as well as estimation errors (e.g., wrongly resolved integer ambiguities). Therefore, the assumption of uncorrelated measurements, which is integral to most estimation methods, is in general not given (Miller et al., 2012; Niu et al., 2014). The time-correlated nature of the measurement has to be explicitly taken into account in order to obtain a realistic trajectory estimate (Niu et al., 2018).

Tight coupling directly integrates the raw GNSS pseudoranges & carrier phase measurements (Falco et al., 2017) with the INS. The benefit of tight coupling is most apparent for real-time processing, with less accurate INS' and in areas with bad GNSS reception (Falco et al., 2017), especially when combined with a robust estimator (Crespillo et al., 2018) to further reduce the impact of outliers. Additionally, if less than 4 GNSS satellites are available, stand-alone GNSS cannot

[4] The exponential map $\exp : \mathbb{R}^3 \to \mathrm{SO}(3)$ 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 cross product map

$$[\,\cdot\,]^\times : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$$

$$\boldsymbol{\omega} \mapsto \boldsymbol{\Omega} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}, \quad \text{s.t. } [\boldsymbol{\omega}]^\times \boldsymbol{v} = \boldsymbol{\omega} \times \boldsymbol{v} \ \forall \boldsymbol{v} \in \mathbb{R}^3,$$

and the matrix exponential $\exp(\boldsymbol{\Omega}) = \sum_{i=0}^{\infty} \boldsymbol{\Omega}^i$. Its inverse is the logarithm $\log : \mathrm{SO}(3) \to \mathbb{R}^3$. As multiple vectors can have the same exponential, making a suitable restriction, e.g., $\|\omega\| < \pi$, is required for bijectivity. Note that in general $\exp(\boldsymbol{A})\exp(\boldsymbol{B}) \neq \exp(\boldsymbol{A}+\boldsymbol{B})$, $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{3\times3}$.

[5] While the covariance kernel K in Eqs. (4.4) and (4.5) and the covariance $\mathbb{V}$ in Eq. (3.7) refer to similar concepts, we use the notation common in the respective context.

[6] The notation $\widetilde{\boldsymbol{z}}$ signifies a noisy measurement of the true quantity $\boldsymbol{z}$.

provide a position/velocity solution and loose coupling degrades to a pure INS while tight coupling still allows for the integration of the available GNSS measurements. The performance disadvantage of loose coupling can however be reduced in a post-processing situation via Kalman smoothing (Wang et al., 2020a). In GNSS/INS integration, loose coupling is often also referred to as *position-domain* integration and tight coupling as *range-domain* integration. This corresponds to our notion of trajectory-level and sensor-level error modelling.

Deep coupling methods go even further to combine the INS integration with GNSS signal processing (Lashley et al., 2010). However, this requires very direct access to the underlying GNSS hardware and as such is hard to implement as an end-user compared to the loosely and tightly coupled approaches which only depend on the RINEX output. Deep GNSS/INS coupling enables information from the INS to be used to improve GNSS signal acquisition and tracking. It also enables detection and rejection of outliers, such as non-line-of-sight (NLOS) effects, before they are included in the navigation filter.

Inertial sensors

The inertial sensors, namely accelerometer and gyroscope, can theoretically provide a standalone trajectory (Titterton and Weston, 2004) via integration, but due to the relative nature of the measurements, errors accumulate and cause large drifts in the solution over time.

Gyroscopes measure angular rate $\boldsymbol{\omega}_{ib}^{b}$ and accelerometers the specific force $\boldsymbol{f}_{ib}^{b}$ of the body frame w.r.t. the inertial frame, in body frame coordinates. The relation between absolute position $\boldsymbol{x}_{eb}^{e}$ and attitude $\boldsymbol{R}_{e}^{b}$ of the body frame and the specific force $\boldsymbol{f}_{ib}^{b}$ and angular velocity $\boldsymbol{\omega}_{ib}^{b}$ is given by the navigation equations[7]

$$\begin{aligned}\boldsymbol{\Omega}_{ib}^{b} &= \boldsymbol{\Omega}_{ie}^{b} + \boldsymbol{\Omega}_{eb}^{b} \\ &= \boldsymbol{R}_{e}^{b}\boldsymbol{\Omega}_{ie}^{e}\boldsymbol{R}_{b}^{e} + \boldsymbol{R}_{e}^{b}\dot{\boldsymbol{R}}_{b}^{e},\end{aligned} \tag{4.9}$$

$$\begin{aligned}\boldsymbol{f}_{ib}^{b} &= \boldsymbol{a}_{ib}^{b} - \boldsymbol{\gamma}_{ib}^{b} \\ &= \boldsymbol{R}_{i}^{b}\ddot{\boldsymbol{x}}_{ib}^{i} - \boldsymbol{R}_{e}^{b}\boldsymbol{\gamma}_{ib}^{e} \\ &= \boldsymbol{R}_{e}^{b}\left((\boldsymbol{\Omega}_{ie}^{e})^{2}\boldsymbol{x}_{eb}^{e} + 2\boldsymbol{\Omega}_{ie}^{e}\dot{\boldsymbol{x}}_{eb}^{e} + \ddot{\boldsymbol{x}}_{eb}^{e} - \boldsymbol{\gamma}_{ib}^{e}\right).\end{aligned} \tag{4.10}$$

Here, $\boldsymbol{\gamma}_{ib}^{b}$ is the acceleration due to gravitational force. The skew-symmetric matrix $\boldsymbol{\Omega}_{ib}^{b}$ of the angular rate vector $\boldsymbol{\omega}_{ib}^{b}$ is defined as

$$\boldsymbol{\Omega}_{ib}^{b} = \left[\boldsymbol{\omega}_{ib}^{b}\right]^{\times} := \begin{pmatrix} 0 & -\omega_{ib,z}^{b} & \omega_{ib,y}^{b} \\ \omega_{ib,z}^{b} & 0 & -\omega_{ib,x}^{b} \\ -\omega_{ib,y}^{b} & \omega_{ib,x}^{b} & 0 \end{pmatrix},$$

and analogously, the earth rotation vector $\boldsymbol{\omega}_{ie}^{e}$ defines the skew-symmetric matrix $\boldsymbol{\Omega}_{ib}^{b}$.

It should be emphasized that IMU measurements can take different forms. The basic output are the *instantaneous values* $\boldsymbol{\omega}_{ib}^{b}(t)$, $\boldsymbol{f}_{ib}^{b}(t)$, as they occur in the above equations. Higher-end IMUs internally measure the angular rate and specific force at a very high sample rate (>1 kHz), but output down-sampled *averaged* or *integrated* values

$$\Delta\boldsymbol{v}_{ib}^{b}(t_i) \approx \int_{t_{i-1}}^{t_i} \boldsymbol{f}_{ib}^{b}(t)\,\mathrm{d}t, \qquad \Delta\boldsymbol{\theta}_{ib}^{b}(t_i) \approx \int_{t_{i-1}}^{t_i} \boldsymbol{\omega}_{ib}^{b}(t)\,\mathrm{d}t \tag{4.11}$$

at a lower rate (typically 100–400 Hz, cf. Groves, 2013, Ch. 4). The exact formula for calculating the delta values depends on the IMU model. Often, additional corrections are applied to reduce coning and sculling errors in subsequent integration (Al-Jlailaty and Mansour, 2021). If coning and sculling corrections are applied, the integrated IMU measurements $\Delta\boldsymbol{v}_{ib}^{b}$ and $\Delta\boldsymbol{\theta}_{ib}^{b}$ represent the difference in velocity and orientation between two successive sample times (Savage, 1998a,b; Groves, 2013)

$$\boldsymbol{R}_{e}^{b}(t_i) \approx \exp\left(-\Delta\boldsymbol{\theta}_{ib}^{b}(t_i)\right)\boldsymbol{R}_{e}^{b}(t_{i-1})\exp\left(\boldsymbol{\omega}_{ie}^{e}\,\Delta t_i\right), \tag{4.12}$$

$$\boldsymbol{v}_{eb}^{e}(t_i) \approx \boldsymbol{v}_{eb}^{e}(t_{i-1}) + \boldsymbol{R}_{b}^{e}(t_i)\Delta\boldsymbol{v}_{ib}^{b}(t_i) \tag{4.13}$$

[7] For the complete derivations, refer to Groves (2013, Ch. 2.5).

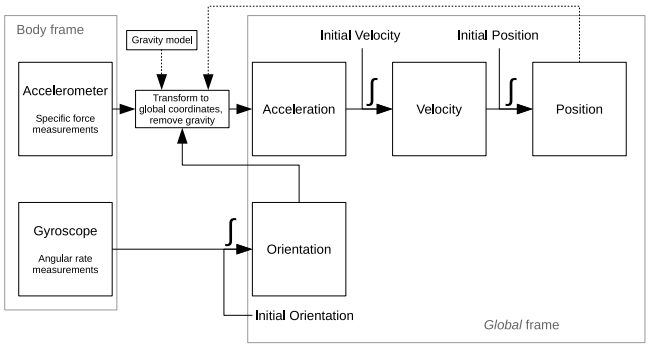

Fig. 9. Inertial measurements as input in the process model, cf. Eqs. (4.12) and (4.13).

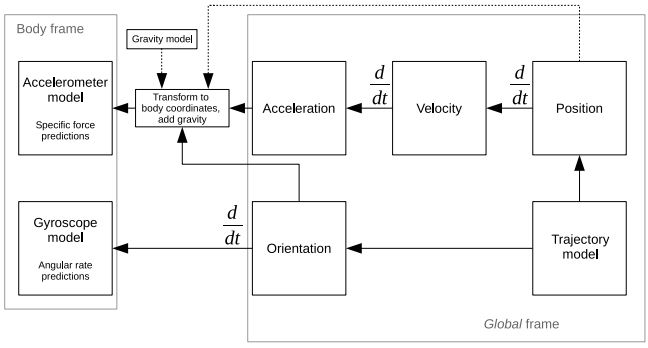

Fig. 10. Inertial measurements as observations in the measurement model, cf. Eqs. (4.9) and (4.10).

$$+\,\Delta t_i\left(\boldsymbol{\gamma}_{ib}^{e} - \left(\boldsymbol{\Omega}_{ie}^{e}\right)^{2}\boldsymbol{x}_{eb}^{e}(t_{i-1}) - 2\boldsymbol{\Omega}_{ie}^{e}\boldsymbol{v}_{eb}^{e}(t_{i-1})\right),$$

where $\Delta t_i = t_i - t_{i-1}$. Eqs. (4.12) and (4.13) approximate the integral solution of Eqs. (4.9) and (4.10) from t_{i-1} to t_i.

In practice, inertial measurements can be incorporated in the trajectory estimation in two ways. The classic way is as input to the process model Eq. (3.1), where the specific force and angular velocity measurements are numerically integrated to obtain predictions for position, velocity and orientation (Fig. 9). In newer optimization-based approaches, inertial measurements are incorporated into the measurement model (Fig. 10), with predictions of specific force and angular rate derived by differentiation from position and orientation. If no trajectory prior is explicitly specified, then there is no process model, only a measurement model. This has the benefit of being an explicit formulation in the form of Eq. (3.2), so that the noise is treated optimally in the least-squares sense. In comparison, the numerical integration of IMU measurements requires propagating the error stochastic properties through the integration process (cf. Forster et al., 2017).

Errors in the inertial measurements cause a time-dependent drift in the position and orientation. Integration with other sensors allows for in-run calibration of the inertial sensor errors. For this, an accurate model of the inertial sensors error characteristics is required (Groves, 2013). The general error model is

$$\begin{aligned}\widetilde{\boldsymbol{f}}_{ib}^{b} &= (\boldsymbol{I} + \boldsymbol{M}_{f})\boldsymbol{f}_{ib}^{b} + \boldsymbol{b}_{f} + \boldsymbol{w}_{f}, \\ \widetilde{\boldsymbol{\omega}}_{ib}^{b} &= (\boldsymbol{I} + \boldsymbol{M}_{\omega})\boldsymbol{\omega}_{ib}^{b} + \boldsymbol{g}_{\omega}(\boldsymbol{f}_{ib}^{b}) + \boldsymbol{b}_{\omega} + \boldsymbol{w}_{\omega},\end{aligned} \tag{4.14}$$

where $\boldsymbol{b}_{\star}$ are the sensor biases, $\boldsymbol{w}_{\star}$ are the sensor noises and $\boldsymbol{g}_{\omega}(\boldsymbol{f}_{ib}^{b})$ is an error caused by gyroscope sensitivity to acceleration. The matrices $\boldsymbol{M}_{f}$ and $\boldsymbol{M}_{\omega}$ are of the form

$$\boldsymbol{M} = \begin{pmatrix} s_x & m_{xy} & m_{xz} \\ m_{yx} & s_y & m_{yz} \\ m_{zx} & m_{zy} & s_z, \end{pmatrix} \tag{4.15}$$

with scale factors $s_{\star}$ and axis misalignment parameters $m_{\star\star}$.

IMUs are generally calibrated in a laboratory setting to remove systematic errors (see Poddar et al., 2017). Some errors, such as bias and scale factor, can vary from run to run (turn-on bias) as well as within a run (time-varying bias). An accurate stochastic model for their behaviour is required. Time-varying biases and scale factors are traditionally modelled as random walk or first-order Gauss–Markov processes (Farrell et al., 2019), or as ARMA processes (Hemerly, 2017). The noises $\boldsymbol{w}_f$, $\boldsymbol{w}_\omega$ are assumed white with a known power spectral density. The g-dependent errors $\boldsymbol{g}_\omega$ and axis misalignments $m_{\star\star}$ are usually assumed fixed and pre-calibrated, but may also be estimated in-run (Farrell, 2008).

Frame camera

Ideal frame cameras record a 2D perspective projection of a 3D scene. A 3D representation can be recovered from multiple images, e.g., by dense image matching (Remondino et al., 2013). This requires knowledge of camera orientation and calibration, which can be obtained by bundle adjustment (Triggs et al., 2000). Underlying this process is a parametric model of the camera which describes how a 3D point $\boldsymbol{x}^c \in \mathbb{R}^3$ in camera coordinates corresponds to measured image point $\widetilde{\boldsymbol{u}} \in \mathbb{R}^2$ in *image space*.

A standard camera model is perspective projection

$$\boldsymbol{u}(\boldsymbol{x}^c) = \frac{1}{\boldsymbol{x}^c_z}\begin{pmatrix} f\boldsymbol{x}^c_x \\ f\boldsymbol{x}^c_y \end{pmatrix} - \boldsymbol{u}_0, \tag{4.16}$$

where f is the focal length and $\boldsymbol{u}_0$ is the principal point. Internal camera parameters include focal length and principal point, as well as additional parameters such as lens distortion. The coordinates $\boldsymbol{x}^c$ in the camera coordinate system of a point $\boldsymbol{x}^e$ depend on the cameras exterior orientation at the acquisition time t, which may be derived from the platform trajectory and camera mounting

$$\begin{aligned} \boldsymbol{x}^c(\boldsymbol{x}^e) &= \boldsymbol{R}^c_e(t)(\boldsymbol{x}^e - \boldsymbol{x}^e_{ec}(t)), \\ &= \boldsymbol{R}^c_b\left(\boldsymbol{R}^b_e(t)(\boldsymbol{x}^e - \boldsymbol{x}^e_{eb}(t)) - \boldsymbol{x}^b_{bc}\right). \end{aligned} \tag{4.17}$$

In summary, the measurement equation for an image measurement $\widetilde{\boldsymbol{u}}$ corresponding to a 3D point $\boldsymbol{x}^e$ is given by

$$\widetilde{\boldsymbol{u}} = \boldsymbol{u}(\boldsymbol{x}^c(\boldsymbol{x}^e)) + \boldsymbol{w}_u, \tag{4.18}$$

where $\boldsymbol{w}_u$ is the image measurement noise and $\boldsymbol{u}$ implicitly depends on trajectory, camera calibration and camera mounting parameters.

Laser scanner

In kinematic laser scanning, the scanner moves during acquisition and thus the scanner trajectory is required for generation of a 3D point cloud. Each laser scanner measurement is primarily a range. Through knowledge of the beam direction, the range measurement describes a single point $\boldsymbol{x}^s$ in the scanner's coordinate system. While all pixels of an image obtained by frame camera share the same acquisition time, each laser shot is temporally separate and has its own exterior orientation.

To obtain a 3D point cloud in a global coordinate system, the scanners orientation $\boldsymbol{R}^b_s$ and lever arm $\boldsymbol{x}^s_{bs}$ with respect to the platform coordinate system and the platform's position $\boldsymbol{x}^e_{eb}$ and orientation $\boldsymbol{R}^e_b$ are required. A point's position in the earth-centered earth-fixed coordinate system is then given by the georeferencing equation

$$\boldsymbol{x}^e = \boldsymbol{x}^e_{eb}(t) + \boldsymbol{R}^e_b(t)\,\boldsymbol{R}^b_s\left(\boldsymbol{x}^s_{bs} + \boldsymbol{x}^s\right). \tag{4.19}$$

Indirect sensor orientation, as with camera images, is not generally possible. This poses a chicken-and-egg problem for using laser scanners alone as sensors for trajectory estimation. The problem can be reduced to a point cloud alignment problem if scans are taken in a stop-and-scan manner (Nüchter et al., 2007). Relative position and orientation between the scan positions is derived from the transformation between the two point clouds of the two scans. This transformation is estimated via *scan matching*, e.g., using ICP. If the movement of the scanner is slow compared to the scan speed, the distortion caused by the movement may be neglected or modelled with certain assumptions, such as constant velocity during a scan (Zhang and Singh, 2017). When used in conjunction with other sensors, such as GNSS/INS, the existing trajectory can be used to a-priori correct for scanner motion (Chang et al., 2019). This is similar to strip adjustment in kinematic laser scanning (Glira et al., 2016), where the GNSS/INS trajectory is used as initial solution and constraint to the adjustment. The underlying principle is that of geometric invariance between repeated observations of the same object. In this way, the laser measurements provide constraints either between different points in time or with respect to existing reference information.

Object space model

Neither camera nor laser scanner directly measure physical 3D coordinates, but both can be used to recover the 3D structure. In contrast to the raw sensor measurements, this derived and primarily geometric information lives in 3D, in the *object space*. This suggests a unified modelling in object space of all imaging sensor measurements. The *object space model* describes how information from the imaging sensors is incorporated into the trajectory estimation. This of course depends on the sensor used as well as on the application and the environment.

In this context, we distinguish between physical *objects* in 3D space and corresponding *features*. The objects, also called landmarks in SLAM, are entities in 3D space and composed of at least a location and are possibly parametrized further, e.g., a building facade is modelled as a plane and represented by a point and a normal vector. A feature is derived from sensor data and serves as a description of an object, which may change depending on sensor orientation and calibration. The process of extracting features from images or point clouds is called *feature detection*. Corresponding features describe the same physical object and are determined by *feature matching*. In this way, objects scanned or imaged at different times can provide anchors to constrain the trajectory, creating an interdependency between trajectory and object space representation of imaging sensor measurements. How features are modelled and how the correspondences are established varies depending on sensor and application. For LiDAR point clouds, correspondences are derived from local geometric properties the point cloud in object space, such as lines or planes. On the other hand, image correspondences are commonly constructed based on keypoints in the images themselves.

While the observed geometric properties differ for frame cameras and laser scanners, both bundle adjustment and strip adjustment typically rely on the principle that the observed geometry is static[8] during the acquisition period and as such, the measurements of the same object at different times must be consistent. Exploiting this requires establishing correspondences between measurements at different times or between measurements and a known ground truth. In a hybrid camera and laser scanner adjustment, correspondences have to be established between images and laser scanner measurements, in addition to the correspondences within the measurements of a single sensor type. Thus, object space serves as interface for integrating measurements from the different imaging sensors.

A distinction is made depending on whether any parameters describing the object space are estimated together with the trajectory and the sensor calibration (cf. Fig. 11):

- **Explicit** object space model: Features are associated with the corresponding objects in object space. The objects are explicitly parametrized by *object parameters*, which are either known a-priori or estimated together with the sensor orientation.

[8] Some works also allow for dynamically changing environments by detecting and discarding (Brasch et al., 2018) or explicitly modelling (Bibby and Reid, 2010) dynamic objects, but this is not considered here.

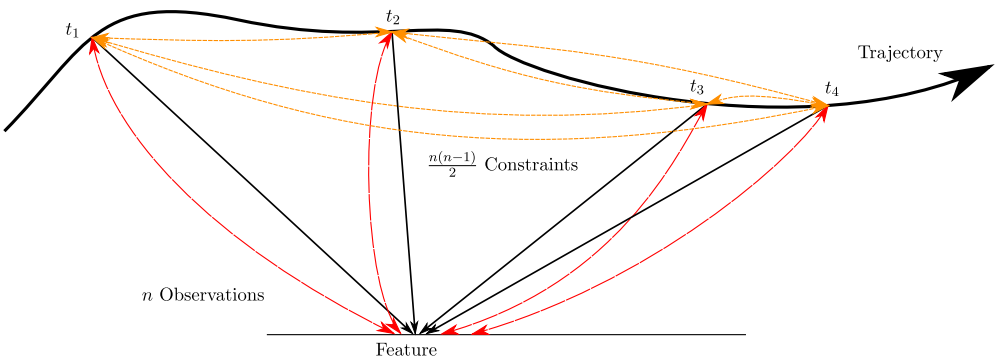

Fig. 11. Feature constraints and observations. If feature correspondences are included in the adjustment as constraints, observing an object from n different viewpoints results in up to $\frac{n(n-1)}{2}$ cross-temporal constraints.

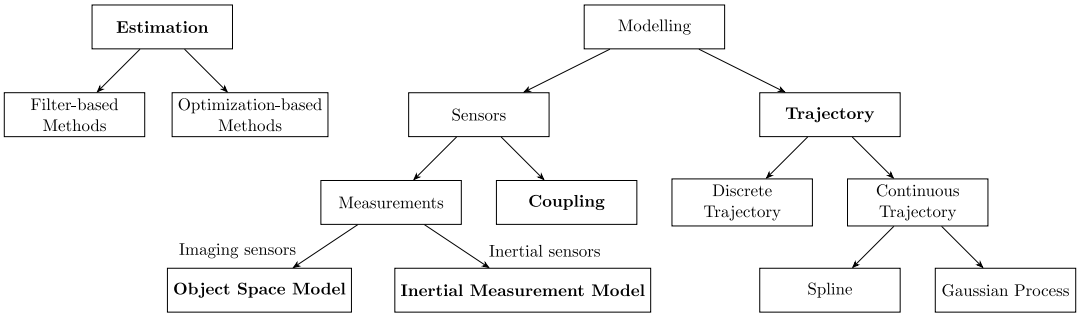

Fig. 12. Trajectory estimation is characterized based on estimation method, sensor coupling, as well as measurement and trajectory models.

- **Implicit** or **structure-free** object space model: Features are associated with corresponding features. While correspondences are established in object space, the final estimation incorporates them as constraints between measurements made at different points in time and thus different exterior orientations. Object parameters can be recovered separately if needed.

The process of image feature detection and matching is a central topic in computer vision and photogrammetry (Ma et al., 2021). In the point cloud domain, 3D feature detection is used for rigid registration of point clouds (Huang et al., 2021b). However, detecting and matching 3D features in kinematic laser scanning is more challenging due to the following reasons:

- If the trajectory or the sensor calibration is wrong, objects will look different at different times.
- During optimization the trajectory and point cloud change, and in turn the feature or correspondence may change. Depending on the magnitude of the change, this requires recalculation of features and correspondences.
- Trade-off between small-scale and large-scale features: Smaller features have better time-locality, but matching is harder. Actual 1-to-1 correspondences are rare (i.e., in practice, there are no exact point-to-point correspondences).
- LiDAR-derived features can be aggregates of a number of individual measurements, which are taken at slightly different points in time. Such features are in a sense synthetic measurements and need to be associated with a timestamp (e.g., closest actual point).

5. Characterization and survey

In Section 3, two approaches to trajectory estimation were discussed based on the probabilistic estimation method. Filtering methods can be further divided into deterministic (e.g., Kalman filter) and nondeterministic (e.g., particle filter). Within the optimization-based approaches, there are multiple paradigms differing in their exact formulation, application and origin. This section reviews the literature on trajectory estimation and gives a characterization based on the modelling and estimation framework established above (see Fig. 12). Table 2 shows a comparison of trajectory estimation methods, each with novel methodology or application.

Estimation method

It was already argued above that a batch NLS estimation method is in many cases superior to a filtering approach. This lead to increasing popularity of such methods recently, which is also reflected in Table 2. In SLAM, optimization-based methods have become dominant in the past years (Cadena et al., 2016; Debeunne and Vivet, 2020). The *pose graph optimization* (PGO, Grisetti et al., 2010) is a special case of factor graph optimization, where each measurement is a relative transformation between two poses. As these relative transformations are derived from various sources, such as wheel, image or lidar odometry, this represents a relatively general method for loosely coupling various sensors. Methods exploiting the special structure of PGO to provide global optimality (Rosen et al., 2019) or significant performance improvements (Moreira et al., 2021) have begun to replace the usual Gauss–Newton-based NLS optimization.

In photogrammetry, bundle adjustment is well established and lends itself to inclusion of additional sensors' measurements directly in a single adjustment step. Cucci et al. (2017) integrates bundle adjustment with GNSS and INS, which proves particularly useful for low-cost sensors and in GNSS-denied situations. The method is based on so-called *dynamic networks*, which are a generalization of geodetic network adjustment to allow joint static and dynamic modelling, i.e., network adjustment with time-dependent quantities (Colomina and Blázquez, 2004). This approach has been validated in a first simulation study with inertial and laser measurements (Rouzaud and Skaloud, 2011) as well as practically for UAV laser scanning (Brun et al., 2022). While these are post-processing methods, for many other applications solving the optimization problem from scratch whenever new measurements

Table 2
Trajectory estimation literature overview: Selected seminal works with novel methodology or application (marked in bold). A '/' implies to corresponding column is not mentioned in the paper or is not applicable, e.g., in a purely theoretical paper. Parentheses in the coupling column specify coupling type: A + B means that sensor A is tightly coupled to sensor B, while (A) + B implies sensor A is loosely coupled to sensor B.
NLS: Nonlinear least-squares. NLSi: Incrementally solved NLS. EM: Expectation–Maximization algorithm. PF: Particle filter. MSCKF: Multi-state constraint Kalman filter.

	Application	Estimation method	Batch size	Trajectory representation	Sensors and coupling	Inertial measurement model	Object space model
Jung and Taylor, 2001	3D Reconstruction, ...	2 × NLS	All	**Spline**	(Camera)+INS	Integrated/ Instantaneous	/
Montemerlo et al., 2002	**SLAM**	**PF**	Current	Discrete	/	/	/
Colomina and Blázquez, 2004	Dynamic Modelling	**NLS**	All	Discrete	/	/	/
Dellaert and Kaess, 2006	**SLAM**	**NLS**	All/subset	Discrete	/	/	/
Mourikis and Roumeliotis, 2007	Navigation	**MSCKF**	Subset	Discrete	Camera+INS	Integrated	Implicit
Soloviev, 2008	Navigation	EKF	Current	Discrete	**GNSS+LiDAR+INS**	Integrated	Implicit
Kaess et al., 2008	SLAM	**NLSi**	All	Discrete	/	/	/
Nagai et al., 2009	**UAV Mapping**	NLS + EKF	Current	Discrete	**(Camera)+(GNSS) +INS**	Integrated	/
Lupton and Sukkarieh, 2009	SLAM	NLSi	All	Discrete	INS + Camera	**Pre-integrated**	Explicit
Bibby and Reid, 2010	SLAM	**EM/NLS**	Window	**Spline**	Radar	/	Explicit
Rouzaud and Skaloud, 2011	Mobile laser scanning	**NLS**	All	Discrete	LiDAR+INS	**Instantaneous**	Explicit
Fleps et al., 2011	Calibration	NLS[a]	All	**Spline**	**(Camera) + INS**	Instantaneous	/
Indelman et al., 2012	**Bundle adjustment**	**NLSi**	All	Discrete	Camera	/	**Implicit**
Furgale et al., 2012	SLAM	NLS	All	**Spline**	**INS+Camera**	Instantaneous	Explicit
Klein and Filin, 2012	Mobile laser scanning	**EKF**	Current	Discrete	**INS+LiDAR**	Integrated	Known
Angelats et al., 2012	Airborne Mapping	NLS	All	Discrete	Camera+LiDAR	Integrated	Explicit
Furgale et al., 2013	**Calibration**[b]	NLS	All	Spline	Camera+INS	Instantaneous	Explicit
Vu et al., 2013	Vehicle trajectory	**NLS**	All	Discrete	**GNSS+INS**	Integrated	/
Lovegrove, 2013	SLAM	NLS	All	**Spline**[c]	INS+Camera	Instantaneous	Explicit
Tong et al., 2013	SLAM	NLS	All	**Gaussian Process**	LiDAR+Wheel	/	Explicit
Anderson et al., 2014	SLAM	NLS	All	**Wavelet**[d]	LiDAR	/	Implicit
Zhang and Singh, 2014	**Odometry**	NLS	Current	Discrete	**(LiDAR)+(INS)**	/	Implicit
Anderson et al., 2015	SLAM	NLS	Window	Spline/**Relative**	Camera	/	Explicit
Glira et al., 2016	Airborne Laser Scanning	NLS	All	Discrete/Spline	(GNSS+INS)+**LiDAR**	/	Implicit
Cucci et al., 2017	**Bundle adjustment**	NLS	All	Discrete	**(GNSS)+INS +Camera**	Instantaneous	Explicit
Ovrén and Forssén, 2019	Structure from Motion	NLS	All	**Spline**[e]	INS+Camera	Instantaneous	Explicit
Glira et al., 2019	Hybrid adjustment	NLS	All	Spline	(GNSS+INS)+ **Camera+LiDAR**	/	/
Ye et al., 2019	LiDAR-Inertial Odometry	NLS	All	Discrete	**INS+LiDAR**	Pre-integrated	Implicit
Wen et al., 2019a	Autonomous Driving	**NLSi**	All	Discrete	**GNSS+INS, Camera**[f]	Integrated	/
Rückert and Stamminger, 2021	SLAM	**2 × NLS**[g]	Subset	Discrete	INS+Camera	Pre-integrated	Explicit
Brun et al., 2022	**UAV laser scanning**	**NLS**	All	Discrete	**(GNSS)+INS+LiDAR**	Pre-integrated	Implicit

[a]Sequential quadratic programming (SQP) is used to solve a nonlinear least squares problem with equality constraints.
[b]Allows for spatio-temporal calibration by estimating a time offset in addition to the other calibration parameters.
[c]First to use a Lie-group cumulative spline formulation (cf. Kim et al. (1995)).
[d]Trajectory is represented by a hierarchical B-Spline wavelet decomposition.
[e]Spline knot spacing is determined by the method introduced in Ovren and Forssen (2018).
[f]The camera is used only for GNSS aiding via none-line-of-sight detection.
[g]The problem is split into two interdependent parts: an IMU-based pose estimation and a bundle adjustment.

arrive is costly. One solution is to keep the *batch size* low by using a sliding window or keyframe approach. Kaess et al. (2008) introduces an incremental smoothing solution for the MAP problem, which is further improved by the development of specialized data structures (Kaess et al., 2012). This is used in Indelman et al. (2013) for real-time capable GNSS/INS/camera-based navigation with much lower computational requirements than solving the full problem.

Trajectory representation

The second important aspect of trajectory estimation is the representation of the trajectory itself. The spline-based representation was first used in this context by Jung and Taylor (2001) for the estimation of a camera trajectory. This was achieved by fitting a spline to accelerometer measurements after recovering the orientation from both gyroscope measurements and image relative orientations. More recently, Bibby

and Reid (2010) used splines to represent both the trajectory as well as objects in a dynamic environment. In Furgale et al. (2015), the authors provide a thorough exposition of their spline-based trajectory estimation method as well as a review of continuous-time trajectory estimation in general.

An alternative approach to continuous-time trajectory estimation is based on Gaussian processes (Tong et al., 2013). The representation of the trajectory as a Gaussian process requires specifying a prior on the trajectory, based on suitable assumptions on the vehicle dynamics (Dong et al., 2018). Commonly used priors such as the *white-noise-on-acceleration* or *white-noise-on-jerk* motion prior (Tang et al., 2019) result in extremely sparse inverse covariance matrices and can therefore be solved efficiently (Barfoot et al., 2014).

The continuous-time formulation has two main benefits: Firstly, position and orientation, and derivatives thereof, are available at all times. Secondly, the continuous-time representation simplifies estimation of sensor time offsets between asynchronous sensors. Because of this, both spline-based (Furgale et al., 2013) and Gaussian-process-based (Li et al., 2021) representations are well suited for spatio-temporal multi-sensor calibration and have been demonstrated to perform better compared to existing discrete-time approaches (Lv et al., 2022). A comparative study of the spline-based continuous-time methods and discrete-time methods is done in Cioffi et al. (2022) for camera-based SLAM, where the continuous-time representation is again shown to be especially effective for recovering time offsets between camera and IMU.

Independent of the type, continuous trajectory representation makes a trade-off between reducing parameter count and trajectory fidelity. This trade-off often results in a smoothing of higher frequencies, which may or may not be desirable depending on the application.

Sensor coupling

An arguably even bigger differentiator of the different trajectory estimation methods are the sensors used, as well as their coupling. Generally, looser coupling allows for modularization of the individual systems which makes implementation less complex and possible computationally more efficient. This comes at the cost of reducing the information available to the trajectory estimation algorithm. For the purpose of obtaining georeferenced 3D models, the sensors of interest are: GNSS, INS and imaging sensors (here: laser scanner and/or camera). The standard processing pipeline for point cloud generation (Fig. 1) is loose coupling of GNSS/INS with the imaging sensor, which consists of an initial GNSS/INS processing with subsequent bundle *or* strip adjustment.

The methods of Glira et al. (2019), Cledat and Skaloud (2020) and Zhou et al. (2021) combine bundle adjustment with strip adjustment in a hybrid manner, where all imaging sensors are adjusted together. This represents a tight coupling of laser scanner and camera, with an existing GNSS/INS trajectory used as position and orientation observations within the adjustment. This *hybrid adjustment* avoids inconsistencies between laser and image point clouds and achieves more robust and accurate trajectory correction and sensor calibration (Toschi et al., 2018; Haala et al., 2022), resulting in higher quality 3D models (Mandlburger et al., 2017; Haala et al., 2022). Different ways of establishing the required correspondences between laser scanner and camera measurements are reviewed in Cledat and Skaloud (2020).

Recently there has been a trend towards integrating the imaging sensors tightly with GNSS and INS. Combining GNSS, IMU and LiDAR measurements in the same adjustment process, Brun et al. (2022) improves point cloud accuracy compared to the classic two-stage procedure (Fig. 1). Cucci et al. (2017) presents a method for integration a position/velocity GNSS solution, IMU measurements and image correspondences in a unified bundle adjustment.

A similar trend is visible in other fields. The tightly coupled IMU/LiDAR odometry frameworks presented in Ye et al. (2019) and Shan et al. (2020) are reported to outperform previous state-of-the-art methods (e.g., Zhang and Singh, 2014). Seo and Chou (2019) tightly couple image and LiDAR measurements in a visual odometry algorithm and show that it outperforms both stand-alone visual and LiDAR odometry as well as depth-enhanced visual odometry. Vu et al. (2013) adapted the batch NLS method of Dellaert and Kaess (2006) for tightly coupled GNSS/INS integration by employing a two-stage process where the results of a batch optimization with pseudo-range and Doppler measurements are used to resolve the carrier-phase integer ambiguities. Newer results from optimization-based tightly coupled GNSS/INS integration aided by other sensors also show promise (Wen et al., 2019a; Wen, 2020). Generally however, loose coupling of GNSS with other sensors is still predominant due to the added complexity of modelling all GNSS errors and resolving the integer ambiguities. Soloviev (2008) avoids the latter problem by considering carrier-phase differences between subsequent GNSS measurement epochs.

Inertial measurement model

The inertial measurement model describes vehicle dynamics and relates the trajectory to the inertial measurements. Filter-based methods (e.g., Mourikis and Roumeliotis (2007)) treat the integrated inertial measurements as input and therefore as part of the process model. The optimization-based approaches (Rouzaud and Skaloud, 2011; Fleps et al., 2011) provide a way of incorporating instantaneous inertial measurements in an explicit Gauss–Markov formulation. Predictions of the instantaneous values of angular velocity and specific force can be computed, given that derivatives of position and orientation are available. Analytical derivatives are readily available for continuous-time formulations (Furgale et al., 2015). For discrete representations, they can be computed by finite-differencing (Cucci et al., 2017).

IMU pre-integration (Lupton and Sukkarieh, 2009) is an approach that greatly reduces the computational workload and has seen much use recently (Le Gentil et al., 2018; Toschi et al., 2018; Song and Hsu, 2021). A pre-integrated measurement aggregates many IMU measurements by integrating specific force and angular rate measurements to obtain fictitious relative measurements, one for position and one for orientation. However, due to the way the inertial sensor errors are included in the pre-integration, estimating the time-correlated components of the sensor errors becomes more difficult (Cucci and Skaloud, 2019). Even if the original sensor noise is Gaussian, the errors of the integrated measurements are not necessarily so, violating the underlying assumptions of most methods. Modern pre-integration formulations do take care to properly account for the Lie-group structure for angular velocity pre-integration and propagation of the associated measurement covariance (Forster et al., 2017; Le Gentil and Vidal-Calleja, 2021).

Object space model

The various approaches of incorporating imaging sensor measurements differ mainly in two aspects: The type of feature, how those features are found, as well as how correspondences established and subsequently incorporated in the estimation. The classic approach for LiDAR-derived features is to identify points that lie on a common surface, most practically a plane and minimize the point-to-plane distances while estimating the plane parameters (Skaloud and Lichti, 2006; Kaess, 2015). Alternatively, pairwise constraints between points lying on the same plane may be formed in a *structure-less* approach as in Glira et al. (2015), where the correspondences are iteratively re-established in an ICP-like procedure.

The former is similar to the approach taken in classical bundle adjustment (Triggs et al., 2000), which consists of estimating the point coordinates for an image correspondence jointly with the sensor orientations. Conversely, the structure-less approach, does not explicitly model the objects and instead incorporates a correspondence as constraint between two (Rodriguez et al., 2011) or three images (Schneider et al., 2017). Apart from points, more complicated features such as lines (Zhang et al., 2007) or even splines (Lee and Yu, 2009) may also be used.

The underlying assumption to the methods described so far is that the object space is predominantly static. Any changes in the observed geometry between two observations is unmodelled and would cause systematic errors if undetected. A notable exception to this is Bibby and Reid (2010), where objects are allowed to move and their trajectories are modelled by time-varying splines.

In some applications, prior knowledge of the environment is available (cf. Toth et al., 2009). In this case, the *known* object coordinates are fixed in the optimization. Klein and Filin (2012) and Bureick et al. (2019) use laser measurements and known building models in an integrated INS/LiDAR navigation.

In summary, five defining characteristics of trajectory estimation methods are identified: The estimation method, the sensors used and their coupling, the representation of the trajectory, the inertial measurement model, and the object space model.

6. Discussion

6.1. Trends

Trajectory estimation is central to the task of obtaining accurate 3D models from moving imaging sensors. As an inherently multi-disciplinary problem, it has benefitted strongly from an increased convergence between the fields of remote sensing, computer vision, robotics and navigation. We have identified four essential trends in trajectory estimation; although in this work we have discussed trajectory estimation from the viewpoint of kinematic mapping using imaging sensors, these trends largely apply to multi-sensor fusion in general.

From recursive filtering to batch optimization. Solutions for integrated navigation were previously based on recursive estimation, namely Kalman filter type solutions. Recently, approaches based on the batch MAP estimate have been gaining traction. Incremental or simplified solutions exist, but mainly for performance reasons: The full batch approach is theoretically preferable, although computationally expensive.

From discrete to continuous trajectory representations. Classical filtering methods parametrize the evolution of position and orientation as a discrete set of states. As the inclusion of multiple, possibly asynchronously sampled sensors requires estimation of position and orientation at very high frequencies, a continuous trajectory parametrization has proven useful and is widely used especially in the robotics community. This allows evaluation of position and orientation at arbitrary times. Apart from stochastic priors, additional 'deterministic' prior information on the trajectory can be incorporated by choosing suitable trajectory representations. On the other hand, the continuous formulation can be more computationally demanding if measurements are synchronous or linear interpolation suffices. It should be mentioned that no exhaustive comparison between all the different trajectory representations has been done to date.

From trajectory level to sensor level error modelling. Trajectory level error modelling is effective for small errors or errors following certain characteristics. However, this approach can lead to a deformation of the trajectory and subsequently the 3D model, as the underlying measurements (GNSS and IMU) are disregarded. Advances in estimation algorithms as well as available computational power have caused a trend towards a holistic multi-sensor estimation methodology, where errors are modelled at the sensor level, i.e., in a tightly coupled manner.

Towards mathematically rigorous modelling. Recently presented methodologies have employed more sophisticated mathematical techniques. This manifests in different ways: For the trajectory representation, increasingly the classic Euler-angle representation is criticized and replaced by parametrizations respecting the underlying Lie-group structure. Previously, weighted least-squares was employed simply as an optimization tool, but it is now mostly used within a probabilistic framework, as the need for accurate stochastic error modelling becomes more evident. For these statistical interpretations, work is done in order to incorporate errors with different, possibly a-priori unknown probability densities. Estimators providing strong theoretical guarantees, both with respect to statistical optimality as well as numerical performance, are preferred where computationally feasible.

6.2. Challenges

From a purely theoretic standpoint, there is no drawback to a tightly-coupled full batch estimation, wherein all measurements are considered simultaneously. Compared to an incremental or loosely coupled approach, a tightly coupled global approach can be expected to perform better in terms of robustness as well as accuracy. However, a practical formulation of this 'fully tightly-coupled' method has proven elusive. In most cases, simplifications are made to allow for more efficient processing, or sensors are partly coupled in a loose manner to simplify the model and the necessary computations. Among the various simplifications and different coupling combinations, no clearly superior method has been established. If the trend is there, and global tight coupling is superior, then why is this not reflected in practice? We argue that this is mainly due to two reasons, outlined below.

Model complexity. While methods employing tighter coupling often show higher performance in practice, it requires significant effort. For sensor-level error modelling, in-depth knowledge of the sensor is needed to accurately model the errors without introducing systematic errors in the estimation. For this, access to the raw sensor data is necessary. This is problematic especially for high-quality IMUs which are encumbered by export restrictions and often do not easily allow access to raw data, only providing the integrated navigation solution. For GNSS data however, this problem is already solved: RINEX provides a standardized, documented data exchange format.

Nevertheless, tight coupling requires estimation of all relevant parameters and measurements at the same time. This leads to higher processing and memory requirements compared to loose coupling. More efficient, parallelizable algorithms and faster hardware will enable further exploitation of tight sensor coupling.

Lack of comparative studies. All methods described above were experimentally evaluated, but for different sensor modalities and applications, and mostly on an individual basis and not comparatively. An exception is Cioffi et al. (2022), where a discrete-time and a spline-based SLAM implementation are compared to each other for different robotics scenarios. However, a general in-depth comparison between different methods is difficult, for one due to lack of high-accuracy reference data, but also because few implementations are openly available and re-implementation is not often done due to the complexity of the models. Example applications of different methods, specifically of different trajectory representations, are available in the literature presented in Section 5. In some studies, a combination of low-grade and high-grade hardware is used to test new methodologies with respect to a reference trajectory from the *better* system (e.g., Brun et al., 2022). However, the problem of evaluating the better system remains. Any comparison is also necessarily limited to a specific use case: The requirements on the trajectory solution, as well as assumptions on the system behaviour such as platform dynamics, vibrations and sensor noise, differ depending on the application. For mobile laser scanning, especially in urban environments, positioning errors due to bad GNSS reception or multipath effects, play a large role. For airborne applications, GNSS is generally unproblematic but high-accuracy orientation is required, as errors in orientation have large effects on the georeferenced data due to the longer measurement range.

Challenges for kinematic mapping applications

The generally high precision and accuracy requirements in kinematic mapping applications pose further challenges. Although much work has been done in this regard to improve the accuracy of low-cost sensors, many surveying applications still depend on expensive high-grade hardware. Available processing solutions require experienced operators, due to necessary fine-tuning of the various algorithmic components. Others are based on commercial black-box solutions for parts of the processing workflow, which are often too optimistic in terms of reported accuracy.

6.3. Outlook

In summary, there is a lot of progress with contributions from many different fields. Several clear trends have been established. However, the best solution remains unclear. As such, the main challenge is to allow for rigorous evaluation and comparison of the different methods, which in turn requires suitable benchmark data. Future work in this direction will allow for the development of more accurate, reliable and automatic trajectory estimation methodology, which in turn benefits all kinematic mapping related activities.

Declaration of competing interest

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

Acknowledgements

This work was carried out as part of the project ZAP-ALS (883660) funded by the Austrian Research Promotion Agency (FFG[9]). The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme. We also wish to thank Dr. Andreas Ullrich (RIEGL Laser Measurement Systems GmbH) for very valuable discussions on the topic.

References

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.

Anderson, S., Dellaert, F., Barfoot, T.D., 2014. A hierarchical wavelet decomposition for continuous-time SLAM. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 373–380. http://dx.doi.org/10.1109/ICRA.2014.6906884.

Anderson, S., MacTavish, K., Barfoot, T.D., 2015. Relative continuous-time SLAM. Int. J. Robot. Res. 34 (12), 1453–1479. http://dx.doi.org/10.1177/0278364915589642.

Anderson, B., Moore, J., 2005. In: Dover (Ed.), Optimal Filtering, unabridged republ ed. In: Dover Books on Engineering, Dover Publ, Mineola, NY.

Angelats, E., Blázquez, M., Colomina, I., 2012. Simultaneous orientation and calibration of images and laser point clouds with straight segments. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXIX-B1, 91–96. http://dx.doi.org/10.5194/isprsarchives-XXXIX-B1-91-2012.

Aravkin, A., Burke, J.V., Ljung, L., Lozano, A., Pillonetto, G., 2017. Generalized Kalman smoothing: Modeling and algorithms. Automatica 86, 63–86. http://dx.doi.org/10.1016/j.automatica.2017.08.011.

Bailey, T., Durrant-Whyte, H., 2006. Simultaneous localization and mapping: Part II. IEEE Robot. Autom. Mag. 13 (3), 108–117. http://dx.doi.org/10.1109/MRA.2006.1678144.

Barfoot, T.D., 2017. State Estimation for Robotics. Cambridge University Press, Cambridge, http://dx.doi.org/10.1017/9781316671528.

Barfoot, T.D., Hay Tong, C., Sarkka, S., 2014. Batch continuous-time trajectory estimation as exactly sparse Gaussian process regression. In: Robotics: Science and Systems X. Robotics: Science and Systems Foundation, pp. 221–238. http://dx.doi.org/10.15607/RSS.2014.X.001.

Bell, B., Cathey, F., 1993. The iterated Kalman filter update as a Gauss-Newton method. IEEE Trans. Automat. Control 38 (2), 294–297. http://dx.doi.org/10.1109/9.250476.

Bibby, C., Reid, I., 2010. A hybrid SLAM representation for dynamic marine environments. In: 2010 IEEE International Conference on Robotics and Automation. IEEE, Anchorage, AK, pp. 257–264. http://dx.doi.org/10.1109/ROBOT.2010.5509262.

van den Bos, A., 2007. Parameter Estimation for Scientists and Engineers. Wiley-Interscience, Hoboken, N.J.

Boyd, S., 2010. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. 3 (1), 1–122. http://dx.doi.org/10.1561/2200000016.

Brasch, N., Bozic, A., Lallemand, J., Tombari, F., 2018. Semantic monocular SLAM for highly dynamic environments. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, Madrid, pp. 393–400. http://dx.doi.org/10.1109/IROS.2018.8593828.

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.

Bureick, J., Vogel, S., Neumann, I., Unger, J., Alkhatib, H., 2019. Georeferencing of an unmanned aerial system by means of an iterated extended Kalman filter using a 3D city model. PFG –J. Photogramm. Remote Sens. Geoinf. Sci. 87 (5–6), 229–247. http://dx.doi.org/10.1007/s41064-019-00084-x.

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, arXiv:1606.05830.

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., Tang, J., Jiang, C., Zhu, L., Lehtomäki, M., Kaartinen, H., Kaijaluoto, R., Wang, Y., Hyyppä, J., Hyyppä, H., Zhou, H., Pei, L., Chen, R., 2018b. The accuracy comparison of three simultaneous localization and mapping (SLAM)-based indoor mapping technologies. Sensors (Basel, Switzerland) 18 (10), 3228. http://dx.doi.org/10.3390/s18103228.

Chen, C., Zhu, H., Li, M., You, S., 2018a. A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives. Robotics 7 (3), 45. http://dx.doi.org/10.3390/robotics7030045.

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.

Cledat, E., Skaloud, J., 2020. Fusion of photo with airborne laser scanning. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. V-1-2020, 173–180. http://dx.doi.org/10.5194/isprs-annals-V-1-2020-173-2020.

Colomina, I., 2015. On trajectory determination for photogrammetry and remote sensing: Sensors, models and exploitation. In: Fritsch, D. (Ed.), Proceedings of Photogrammetric Week 2015. Stuttgart, pp. 131–142.

Colomina, I., Blázquez, M., 2004. A unified approach to static and dynamic modelling in photogrammetry and remote sensing. In: ISPRS.

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., Rehak, M., Skaloud, J., 2017. 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.

Debeunne, C., Vivet, D., 2020. A review of visual-LiDAR fusion based simultaneous localization and mapping. Sensors 20 (7), 2068. http://dx.doi.org/10.3390/s20072068.

Del Moral, P., 1996. Non linear filtering: Interacting particle solution. Markov Process. Rel. Fields 2, 555–580.

Dellaert, F., Kaess, M., 2006. Square root SAM: simultaneous localization and mapping via square root information smoothing. Int. J. Robot. Res. 25 (12), 1181–1203. http://dx.doi.org/10.1177/0278364906072768.

Dellaert, F., Kaess, M., 2017. Factor graphs for robot perception. Found. Trends Robot. 6 (1–2), 1–139. http://dx.doi.org/10.1561/2300000043.

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.

Durrant-Whyte, H., Bailey, T., 2006. Simultaneous localization and mapping: Part I. IEEE Robot. Autom. Mag. 13 (2), 99–110. http://dx.doi.org/10.1109/MRA.2006.1638022.

El-Sheimy, N., Youssef, A., 2020. Inertial sensors technologies for navigation applications: State of the art and future trends. Satell. Navig. 1 (1), 2. http://dx.doi.org/10.1186/s43020-019-0001-5.

[9] Österreichische Forschungsförderungsgesellschaft (FFG), Vienna, Austria, www.ffg.at.

Eriksson, A., Bastian, J., Chin, T.-J., Isaksson, M., 2016. A consensus-based framework for distributed bundle adjustment. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Las Vegas, NV, USA, pp. 1754–1762. http://dx.doi.org/10.1109/CVPR.2016.194.

Falco, G., Pini, M., Marucco, G., 2017. Loose and tight GNSS/INS Integrations: Comparison of performance assessed in real urban scenarios. Sensors 17 (2), 255. http://dx.doi.org/10.3390/s17020255.

Fan, T., Murphey, T., 2022. Generalized proximal methods for pose graph optimization. In: Asfour, T., Yoshida, E., Park, J., Christensen, H., Khatib, O. (Eds.), Robotics Research - the 19th International Symposium ISRR. In: Springer Proceedings in Advanced Robotics, Springer Nature, pp. 393–409. http://dx.doi.org/10.1007/978-3-030-95459-8_24.

Farrell, J., 2008. Aided Navigation: GPS with High Rate Sensors. In: Electronic Engineering, McGraw-Hill, New York.

Farrell, J., Silva, F., Rahman, F., Wendel, J., 2019. IMU error state modeling for state estimation and sensor calibration: A tutorial. p. 17.

Fengguang, Z., Li, J., Wen, Y., Bin, G., Zhaoxing, L., Haitao, Q., Li, J., 2017. An airborne position and orientation system (POS) for remote sensing and its current state. In: 2017 IEEE International Conference on Imaging Systems and Techniques (IST). IEEE, Beijing, pp. 1–6. http://dx.doi.org/10.1109/IST.2017.8261499.

Fleps, M., Mair, E., Ruepp, O., Suppa, M., Burschka, D., 2011. Optimization based IMU camera calibration. In: IEEE/RSJ. p. 8.

Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D., 2017. On-manifold preintegration for real-time visual-inertial odometry. IEEE Trans. Robot. 33 (1), 1–21. http://dx.doi.org/10.1109/TRO.2016.2597321.

Furgale, P., Barfoot, T.D., Sibley, G., 2012. Continuous-time batch estimation using temporal basis functions. In: 2012 IEEE International Conference on Robotics and Automation. IEEE, St Paul, MN, USA, pp. 2088–2095. http://dx.doi.org/10.1109/ICRA.2012.6225005.

Furgale, P., Rehder, J., Siegwart, R., 2013. Unified temporal and spatial calibration for multi-sensor systems. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, Tokyo, pp. 1280–1286. http://dx.doi.org/10.1109/IROS.2013.6696514.

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.

Glennie, C., 2007. Rigorous 3D error analysis of kinematic scanning lidar systems. J. Appl. Geod. 1 (3), http://dx.doi.org/10.1515/jag.2007.017.

Glira, P., Pfeifer, N., Briese, C., Ressl, C., 2015. Rigorous strip adjustment of airborne laserscanning data based on the ICP algorithm. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. II-3/W5, 73–80. http://dx.doi.org/10.5194/isprsannals-II-3-W5-73-2015.

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.

Greenspan, R., 1996. GPS and inertial integration. In: Global Positioning System: Theory and Applications, third ed. In: Progress in Astronautics and Aeronautics, vol. 2, American Institute of Aeronautics and Astronautics.

Grisetti, G., Kümmerle, R., Stachniss, C., Burgard, W., 2010. A tutorial on graph-based SLAM. IEEE Intell. Transp. Syst. Mag. 2 (4), 31–43. http://dx.doi.org/10.1109/MITS.2010.939925.

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

Haala, N., Kölle, M., Cramer, M., Laupheimer, D., Zimmermann, F., 2022. Hybrid georeferencing of images and LiDAR Data for UAV-based point cloud collection at millimetre accuracy. ISPRS Open J. Photogramm. Remote Sens. 4, 100014. http://dx.doi.org/10.1016/j.ophoto.2022.100014.

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.

Hartley, R., Zisserman, A., 2003. Multiple View Geometry in Computer Vision, second ed. Cambridge University Press, Cambridge, UK; New York.

Hemerly, E.M., 2017. MEMS IMU stochastic error modelling. Syst. Sci. Control Eng. 5 (1), 1–8. http://dx.doi.org/10.1080/21642583.2016.1262801, arXiv:https://doi.org/10.1080/21642583.2016.1262801.

Hesch, J.A., Kottas, D.G., Bowman, S.L., Roumeliotis, S.I., 2014. Camera-IMU-based localization: Observability analysis and consistency improvement. Int. J. Robot. Res. 33 (1), 182–201. http://dx.doi.org/10.1177/0278364913509675.

Hofmann-Wellenhof, B., Lichtenegger, H., Wasle, E., 2008. GNSS - Global Navigation Satellite Systems: GPS, Glonass, Galileo, and More. Springer, Wien; New York.

Huang, J., Huang, S., Sun, M., 2021a. DeepLM: Large-scale nonlinear least squares on deep learning frameworks using stochastic domain decomposition. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Nashville, TN, USA, pp. 10303–10312. http://dx.doi.org/10.1109/CVPR46437.2021.01017.

Huang, X., Mei, G., Zhang, J., Abbas, R., 2021b. A comprehensive survey on point cloud registration. arXiv:2103.02690 [Cs].

Huang, G.P., Mourikis, A.I., Roumeliotis, S.I., 2011. An observability-constrained sliding window filter for SLAM. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, San Francisco, CA, pp. 65–72. http://dx.doi.org/10.1109/IROS.2011.6095161.

Indelman, V., Roberts, R., Beall, C., Dellaert, F., 2012. Incremental light bundle adjustment. In: Procedings of the British Machine Vision Conference 2012. British Machine Vision Association, Surrey, pp. 134.1–134.11. http://dx.doi.org/10.5244/C.26.134.

Indelman, V., Williams, S., Kaess, M., Dellaert, F., 2013. Information fusion in navigation systems via factor graph based incremental smoothing. Robot. Auton. Syst. 61 (8), 721–738. http://dx.doi.org/10.1016/j.robot.2013.05.001.

Jazwinski, A.H., 1970. Stochastic Processes and Filtering Theory. In: Mathematics in Science and Engineering, (vol. 64), Academic Press, New York.

Jung, S.-H., Taylor, C., 2001. Camera trajectory estimation using inertial sensor measurements and structure from motion results. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Vol. 2. IEEE Comput. Soc, Kauai, HI, USA, pp. II–732–II–737. http://dx.doi.org/10.1109/CVPR.2001.991037.

Kaess, M., 2015. Simultaneous localization and mapping with infinite planes. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, Seattle, WA, USA, pp. 4605–4611. http://dx.doi.org/10.1109/ICRA.2015.7139837.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J.J., Dellaert, F., 2012. iSAM2: Incremental smoothing and mapping using the Bayes tree. Int. J. Robot. Res. 31 (2), 216–235. http://dx.doi.org/10.1177/0278364911430419.

Kaess, M., Ranganathan, A., Dellaert, F., 2008. iSAM: incremental smoothing and mapping. IEEE Trans. Robot. 24 (6), 1365–1378. http://dx.doi.org/10.1109/TRO.2008.2006706.

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.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. J. Basic Eng. 82 (1), 35–45. http://dx.doi.org/10.1115/1.3662552.

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.

Kimeldorf, G.S., Wahba, G., 1970. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. Ann. Math. Stat. 41 (2), 495–502. http://dx.doi.org/10.1214/aoms/1177697089.

Klein, I., Filin, S., 2012. LiDAR and INS fusion in periods of GPS outages for mobile laser scanning mapping systems. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXVIII-5/W12, 231–236. http://dx.doi.org/10.5194/isprsarchives-XXXVIII-5-W12-231-2011.

Kolar, P., Benavidez, P., Jamshidi, M., 2020. Survey of datafusion techniques for laser and vision based sensor integration for autonomous navigation. Sensors 20 (8), 2180. http://dx.doi.org/10.3390/s20082180.

Kschischang, F.R., Frey, B.J., Loeliger, H., 2001. Factor graphs and the sum-product algorithm. IEEE Trans. Inform. Theory 47 (2), 498–519. http://dx.doi.org/10.1109/18.910572.

L'Afflitto, A., 2017. A Mathematical Perspective on Flight Dynamics and Control. Springer Berlin Heidelberg, New York, NY.

Lashley, M., Bevly, D.M., Hung, J.Y., 2010. Analysis of deeply integrated and tightly coupled architectures. In: IEEE/ION Position, Location and Navigation Symposium. IEEE, Indian Wells, CA, USA, pp. 382–396. http://dx.doi.org/10.1109/PLANS.2010.5507127.

Le, H., Zach, C., Rosten, E., Woodford, O.J., 2021. Progressive batching for efficient non-linear least squares. In: Ishikawa, H., Liu, C.-L., Pajdla, T., Shi, J. (Eds.), Computer Vision – ACCV 2020, Vol. 12624. Springer International Publishing, Cham, pp. 506–522. http://dx.doi.org/10.1007/978-3-030-69535-4_31.

Le Gentil, C., Vidal-Calleja, T., 2021. Continuous integration over SO(3) for IMU preintegration. In: Robotics: Science and Systems XVII. Robotics: Science and Systems Foundation, p. 9. http://dx.doi.org/10.15607/RSS.2021.XVII.078.

Le Gentil, C., Vidal-Calleja, T., Huang, S., 2018. 3D lidar-IMU calibration based on upsampled preintegrated measurements for motion distortion correction. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, Brisbane, QLD, pp. 2149–2155. http://dx.doi.org/10.1109/ICRA.2018.8460179.

Lee, W.H., Yu, K., 2009. Bundle block adjustment with 3D natural cubic splines. Sensors 9 (12), 9629–9665. http://dx.doi.org/10.3390/s91209629.

Lesjak, R., Dorn, M., Huber, K., Wieser, M., 2015. The influence of the GNSS solution on the estimated parameters in the course of sensor integration. In: 2015 International Association of Institutes of Navigation World Congress (IAIN). IEEE, Prague, Czech Republic, pp. 1–8. http://dx.doi.org/10.1109/IAIN.2015.7352233.

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, W., Cui, X., Lu, M., 2020. High-precision positioning and mapping using feature-based RTK/LiDAR/INS Integrated System for urban environments. In: Proceedings of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2020. pp. 2628–2640. http://dx.doi.org/10.33012/2020.17745.

Li, S., Wang, L., Li, J., Tian, B., Chen, L., Li, G., 2021. 3D LiDAR/IMU calibration based on continuous-time trajectory estimation in structured environments. IEEE Access 1. http://dx.doi.org/10.1109/ACCESS.2021.3114618.

Lovegrove, S., 2013. Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. In: BMVC. p. 12.

Lupton, T., Sukkarieh, S., 2009. Efficient integration of inertial observations into visual SLAM without initialization. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, St. Louis, MO, USA, pp. 1547–1552. http://dx.doi.org/10.1109/IROS.2009.5354267.

Lupton, T., Sukkarieh, S., 2012. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. IEEE Trans. Robot. 28 (1), 61–76. http://dx.doi.org/10.1109/TRO.2011.2170332.

Lv, J., Zuo, X., Hu, K., Xu, J., Huang, G., Liu, Y., 2022. Observability-aware intrinsic and extrinsic calibration of LiDAR-IMU systems. IEEE Trans. Robot. 1–20. http://dx.doi.org/10.1109/TRO.2022.3174476.

Ma, J., Jiang, X., Fan, A., Jiang, J., Yan, J., 2021. Image matching from handcrafted to deep features: A survey. Int. J. Comput. Vis. 129 (1), 23–79. http://dx.doi.org/10.1007/s11263-020-01359-2.

Madyastha, V., Ravindra, V., Mallikarjunan, S., Goyal, A., 2011. Extended Kalman filter vs. Error state Kalman filter for aircraft attitude estimation. In: AIAA Guidance, Navigation, and Control Conference. American Institute of Aeronautics and Astronautics, Portland, Oregon, p. 23. http://dx.doi.org/10.2514/6.2011-6615.

Mandlburger, G., Pfennigbauer, M., Schwarz, R., Flöry, S., Nussbaumer, L., 2020. Concept and performance evaluation of a novel UAV-Borne topo-bathymetric LiDAR sensor. Remote Sens. 12 (6), 986. http://dx.doi.org/10.3390/rs12060986.

Mandlburger, G., Wenzel, K., Spitzer, A., Haala, N., Glira, P., Pfeifer, N., 2017. Improved topographic models via concurrent airborne lidar and dense image matching. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. IV-2/W4, 259–266. http://dx.doi.org/10.5194/isprs-annals-IV-2-W4-259-2017.

Meng, X., Wang, H., Liu, B., 2017. A robust vehicle localization approach based on GNSS/IMU/DMI/LiDAR Sensor Fusion for autonomous vehicles. Sensors 17 (9), 2140. http://dx.doi.org/10.3390/s17092140.

Miller, C., O'Keefe, K., Gao, Y., 2012. Time correlation in GNSS positioning over short baselines. J. Surv. Eng. 138 (1), 17–24. http://dx.doi.org/10.1061/(ASCE)SU.1943-5428.0000057.

Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., 2002. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In: Proceedings of the AAAI National Conference on Artificial Intelligence. p. 6.

Moreira, G., Marques, M., Costeira, J.P., 2021. Fast pose graph optimization via Krylov-Schur and Cholesky Factorization. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, Waikoloa, HI, USA, pp. 1897–1905. http://dx.doi.org/10.1109/WACV48630.2021.00194.

Mourikis, A.I., Roumeliotis, S.I., 2007. A multi-state constraint Kalman filter for vision-aided inertial navigation. In: Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, Rome, Italy, pp. 3565–3572. http://dx.doi.org/10.1109/ROBOT.2007.364024.

Nagai, M., Chen, T., Shibasaki, R., Kumagai, H., Ahmed, A., 2009. UAV-Borne 3-d mapping system by multisensor integration. IEEE Trans. Geosci. Remote Sens. 47 (3), 701–708. http://dx.doi.org/10.1109/TGRS.2008.2010314.

Nex, F., Armenakis, C., Cramer, M., Cucci, D., Gerke, M., Honkavaara, E., Kukko, A., Persello, C., Skaloud, J., 2022. UAV in the advent of the twenties: Where we stand and what is next. ISPRS J. Photogramm. Remote Sens. 184, 215–242. http://dx.doi.org/10.1016/j.isprsjprs.2021.12.006.

Niu, X., Chen, Q., Zhang, Q., Zhang, H., Niu, J., Chen, K., Shi, C., Liu, J., 2014. Using allan variance to analyze the error characteristics of GNSS positioning. GPS Solut. 18 (2), 231–242. http://dx.doi.org/10.1007/s10291-013-0324-x.

Niu, X., Wu, J., Zhang, Q., 2018. Research on measurement error model of GNSS/INS Integration Based on consistency analysis. Gyrosc. Navig. 9 (4), 243–254. http://dx.doi.org/10.1134/S2075108718040053.

Nüchter, A., Lingemann, K., Hertzberg, J., Surmann, H., 2007. 6D SLAM-3D mapping outdoor environments. J. Field Robotics 24 (8–9), 699–722. http://dx.doi.org/10.1002/rob.20209.

Ovren, H., Forssen, P.-E., 2018. Spline error weighting for robust visual-inertial fusion. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, Salt Lake City, UT, pp. 321–329. http://dx.doi.org/10.1109/CVPR.2018.00041.

Ovrén, H., Forssén, P.-E., 2019. Trajectory representation and landmark projection for continuous-time structure from motion. Int. J. Robot. Res. 38 (6), 686–701. http://dx.doi.org/10.1177/0278364919839765.

Pirotti, F., 2013. State of the art of ground and aerial laser scanning technologies for high-resolution topography of the earth surface. Eur. J. Remote Sens. 46, 66–78. http://dx.doi.org/10.5721/EuJRS20134605.

Poddar, S., Kumar, V., Kumar, A., 2017. A comprehensive overview of inertial sensor calibration techniques. J. Dyn. Syst. Meas. Control 139 (1), 011006. http://dx.doi.org/10.1115/1.4034419.

Qingqing, L., Queralta, J.P., Gia, T.N., Zou, Z., Westerlund, T., 2020. Multi sensor fusion for navigation and mapping in autonomous vehicles: Accurate localization in urban environments. Unmanned Syst. 08 (03), 229–237. http://dx.doi.org/10.1142/S2301385020500168, arXiv:2103.13719.

Ramamurthy, K.N., Lin, C.-C., Aravkin, A., Pankanti, S., Viguier, R., 2017. Distributed bundle adjustment. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). pp. 2146–2154. http://dx.doi.org/10.1109/ICCVW.2017.251.

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. In: Adaptive Computation and Machine Learning, MIT Press, Cambridge, Mass.

Remondino, F., Spera, M.G., Nocerino, E., Menna, F., Nex, F., Gonizzi-Barsanti, S., 2013. Dense image matching: Comparisons and analyses. In: 2013 Digital Heritage International Congress (DigitalHeritage). IEEE, Marseille, France, pp. 47–54. http://dx.doi.org/10.1109/DigitalHeritage.2013.6743712.

Rodriguez, A.L., Lopez-de-Teruel, P.E., Ruiz, A., 2011. Reduced epipolar cost for accelerated incremental SfM. In: CVPR 2011. IEEE, Colorado Springs, CO, USA, pp. 3097–3104. http://dx.doi.org/10.1109/CVPR.2011.5995569.

Rosen, D.M., Carlone, L., Bandeira, A.S., Leonard, J.J., 2019. SE-sync: A certifiably correct algorithm for synchronization over the Special Euclidean Group. Int. J. Robot. Res. 38 (2–3), 95–125. http://dx.doi.org/10.1177/0278364918784361.

Rosen, D.M., Kaess, M., Leonard, J.J., 2013. Robust incremental online inference over sparse factor graphs: Beyond the Gaussian case. In: 2013 IEEE International Conference on Robotics and Automation. IEEE, Karlsruhe, Germany, pp. 1025–1032. http://dx.doi.org/10.1109/ICRA.2013.6630699.

Rouzaud, D., Skaloud, J., 2011. Rigorous integration of inertial navigation with optical sensors by dynamic networks. Navigation 58 (2), 141–152. http://dx.doi.org/10.1002/j.2161-4296.2011.tb01797.x.

Rückert, D., Stamminger, M., 2021. Snake-SLAM: Efficient global visual inertial SLAM using decoupled nonlinear optimization. In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS). pp. 219–228. http://dx.doi.org/10.1109/ICUAS51884.2021.9476760.

Sabatini, A., 2006. Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing. IEEE Trans. Biomed. Eng. http://dx.doi.org/10.1109/TBME.2006.875664.

Särkkä, S., 2013. Bayesian Filtering and Smoothing. Cambridge University Press, Cambridge, http://dx.doi.org/10.1017/CBO9781139344203.

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.

Schmidt, J., Niemann, H., Schmidt, J., Niemann, H., 2001. Using quaternions for parametrizing 3–D rotations in unconstrained nonlinear optimization. In: Vision, Modeling, and Visualization 2001. AKA/IOS Press, pp. 399–406.

Schneider, J., Stachniss, C., Förstner, W., 2017. On the quality and efficiency of approximate solutions to bundle adjustment with epipolar and trifocal constraints. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. IV-2/W3, 81–88. http://dx.doi.org/10.5194/isprs-annals-IV-2-W3-81-2017.

Seo, Y., Chou, C.-C., 2019. A tight coupling of vision-lidar measurements for an effective odometry. In: 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 1118–1123. http://dx.doi.org/10.1109/IVS.2019.8814164.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D., 2020. LIO-SAM: tightly-coupled lidar inertial odometry via smoothing and mapping. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, Las Vegas, NV, USA, pp. 5135–5142. http://dx.doi.org/10.1109/IROS45743.2020.9341176.

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.

Skaloud, J., Schaer, P., Stebler, Y., Tomé, P., 2010. Real-time registration of airborne laser data with sub-decimeter accuracy. ISPRS J. Photogramm. Remote Sens. 65 (2), 208–217. http://dx.doi.org/10.1016/j.isprsjprs.2009.12.003.

Soloviev, A., 2008. Tight coupling of GPS, laser scanner, and inertial measurements for navigation in urban environments. In: 2008 IEEE/ION Position, Location and Navigation Symposium. IEEE, Monterey, CA, USA, pp. 511–525. http://dx.doi.org/10.1109/PLANS.2008.4570059.

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.

Song, Y., Hsu, L.-T., 2021. Tightly coupled integrated navigation system via factor graph for UAV indoor localization. Aerosp. Sci. Technol. 108, 106370. http://dx.doi.org/10.1016/j.ast.2020.106370.

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.

Stuelpnagel, J., 2006. On the parametrization of the three-dimensional rotation group. SIAM Rev. http://dx.doi.org/10.1137/1006093.

Tang, Y., Wu, Y., Wu, M., Wu, W., Hu, X., Shen, L., 2009. INS/GPS Integration: Global observability analysis. IEEE Trans. Veh. Technol. 58 (3), 1129–1142. http://dx.doi.org/10.1109/TVT.2008.926213.

Tang, T.Y., Yoon, D.J., Barfoot, T.D., 2019. A white-noise-on-jerk motion prior for continuous-time trajectory estimation on SE(3). IEEE Robot. Autom. Lett. 4 (2), 594–601. http://dx.doi.org/10.1109/LRA.2019.2891492.

Teunissen, P.J., Montenbruck, O. (Eds.), 2017. Springer Handbook of Global Navigation Satellite Systems. Springer International Publishing, Cham, http://dx.doi.org/10.1007/978-3-319-42928-1.

Titterton, D.H., Weston, J.L., 2004. Strapdown Inertial Navigation Technology, second ed. In: IEE Radar, Sonar, Navigation, and Avionics Series, vol. 17, Institution of Electrical Engineers, Stevenage.

Tong, C.H., Furgale, P., Barfoot, T.D., 2013. Gaussian process Gauss–Newton for non-parametric simultaneous localization and mapping. Int. J. Robot. Res. 32 (5), 507–525. http://dx.doi.org/10.1177/0278364913478672.

Toschi, I., Remondino, F., Rothe, R., Klimek, K., 2018. Combining airborne oblique camera and LiDAR Sensors: Investigation and new perspectives. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLII-1, 437–444. http://dx.doi.org/10.5194/isprs-archives-XLII-1-437-2018.

Toth, C., Grejner-Brzezinska, D., Oh, J.H., Markiel, J.N., 2009. Terrain-based navigation: A tool to improve navigation and feature extraction performance of mobile mapping systems. Bol. Ciênc. Geod. 15 (5), 18.

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.

Trainelli, L., Croce, A., 2004. A comprehensive view of rotation parametrization. In: European Congress on Computational Methods in Applied Sciences and Engineering. Jyväskylä, p. 18.

Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W., 2000. Bundle adjustment — A modern synthesis. In: Goos, G., Hartmanis, J., van Leeuwen, J., Triggs, B., Zisserman, A., Szeliski, R. (Eds.), Vision Algorithms: Theory and Practice, Vol. 1883. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 298–372. http://dx.doi.org/10.1007/3-540-44480-7_21.

Vallet, J., Gressin, A., Clausen, P., Skaloud, J., 2020. Airborne and Mobile LiDAR, which sensors for which application? In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XLIII-B1-2020. Copernicus GmbH, pp. 397–405. http://dx.doi.org/10.5194/isprs-archives-XLIII-B1-2020-397-2020.

Vu, A., Farrell, J.A., Barth, M., 2013. Centimeter-Accuracy smoothed vehicle trajectory estimation. IEEE Intell. Transp. Syst. Mag. 5 (4), 121–135. http://dx.doi.org/10.1109/MITS.2013.2281009.

Wan, E., Van Der Merwe, R., 2000. The unscented Kalman filter for nonlinear estimation. In: Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373). pp. 153–158. http://dx.doi.org/10.1109/ASSPCC.2000.882463.

Wang, Z., Wu, Y., Niu, Q., 2020b. Multi-sensor fusion in automated driving: A survey. IEEE Access 8, 2847–2868. http://dx.doi.org/10.1109/ACCESS.2019.2962554.

Wang, M., Yu, P., Li, Y., 2020a. Performance analysis of GNSS/INS Loosely Coupled Integration Systems Under GNSS Signal Blocking Environment. E3S Web Conf. 206, 02013. http://dx.doi.org/10.1051/e3sconf/202020602013.

Wen, W., 2020. 3D LiDAR aided GNSS and its tightly coupled integration with INS via factor graph optimization. In: Proceedings of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2020. pp. 1649–1672. http://dx.doi.org/10.33012/2020.17557.

Wen, W., Bai, X., Kan, Y.C., Hsu, L.-T., 2019a. Tightly coupled GNSS/INS Integration via factor graph and aided by fish-eye camera. IEEE Trans. Veh. Technol. 68 (11), 10651–10662. http://dx.doi.org/10.1109/TVT.2019.2944680.

Wen, W., Kan, Y.C., Hsu, L.-T., 2019b. Performance comparison of GNSS/INS Integrations Based on EKF and factor graph optimization. In: 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019). Miami, Florida, pp. 3019–3032. http://dx.doi.org/10.33012/2019.17129.

Wilbers, D., Merfels, C., Stachniss, C., 2019. A comparison of particle filter and graph-based optimization for localization with landmarks in automated vehicles. In: 2019 Third IEEE International Conference on Robotic Computing (IRC). IEEE, Naples, Italy, pp. 220–225. http://dx.doi.org/10.1109/IRC.2019.00040.

Ye, H., Chen, Y., Liu, M., 2019. Tightly coupled 3D lidar inertial odometry and mapping. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 3144–3150. http://dx.doi.org/10.1109/ICRA.2019.8793511, arXiv:1904.06993.

Yuan, S., Wang, H., Xie, L., 2021. Survey on localization systems and algorithms for unmanned systems. Unmanned Syst. 1–35. http://dx.doi.org/10.1142/S230138502150014X.

Zefran, M., Kumar, V., 1998. Two methods for interpolating rigid body motions. In: Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), Vol. 4. IEEE, Leuven, Belgium, pp. 2922–2927. http://dx.doi.org/10.1109/ROBOT.1998.680708.

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.

Zhang, J., Singh, S., 2017. Low-drift and real-time lidar odometry and mapping. Auton. Robots 41 (2), 401–416. http://dx.doi.org/10.1007/s10514-016-9548-2.

Zhang, Z., Zhang, J., Zhang, H., 2007. Photogrammetric modeling of linear features with generalized point photogrammetry. Photogramm. Eng. Remote Sens. 73, 1119–1127. http://dx.doi.org/10.14358/PERS.73.9.1119.

Zhou, T., Hasheminasab, S.M., Habib, A., 2021. Tightly-coupled camera/LiDAR integration for point cloud generation from GNSS/INS-Assisted UAV Mapping Systems. ISPRS J. Photogramm. Remote Sens. 180, 336–356. http://dx.doi.org/10.1016/j.isprsjprs.2021.08.020.