PFG (2019) 87:229–247
https://doi.org/10.1007/s41064-019-00084-x

ORIGINAL ARTICLE

Georeferencing of an Unmanned Aerial System by Means of an Iterated Extended Kalman Filter Using a 3D City Model

Johannes Bureick[1] · Sören Vogel[1] · Ingo Neumann[1] · Jakob Unger[2] · Hamza Alkhatib[1]

Received: 15 May 2019 / Accepted: 30 September 2019 / Published online: 30 October 2019

Abstract

In engineering geodesy, the technical progress leads to various kinds of multi-sensor systems (MSS) capturing the environment. Multi-sensor systems, especially those mounted on unmanned aerial vehicles, subsequently called unmanned aerial system (UAS), have emerged in the past decade. Georeferencing for MSS and UAS is an indispensable task to obtain further products of the data captured. Georeferencing comprises at least the determination of three translations and three rotations. The availability and accuracy of Global Navigation Satellite System (GNSS) receivers, inertial measurement units, or other sensors for georeferencing is not or not constantly given in urban scenarios. Therefore, we utilize UAS-based laser scanner measurements on building facades. The building latter are modeled as planes in a three-dimensional city model. We determine the trajectory of the UAS by combining the laser scanner measurements with the plane parameters. The resulting implicit measurement equations and nonlinear equality constraints are covered within an iterated extended Kalman filter (IEKF). We developed a software simulation for testing the IEKF using different scenarios to evaluate the functionality, performance, strengths, and remaining challenges of the IEKF implemented.

Keywords Iterated extended Kalman filter · 3D city model · Unmanned aerial system · Laser scanner measurements · Implicit measurement equation · Equality constraint

Zusammenfassung

Georeferenzierung von Unmanned Aerial Systems mit Hilfe eines iterativen erweiterten Kalman Filters und eines 3D Gebäudemodells. In der Ingenieurgeodäsie führt der technische Fortschritt zu verschiedenen Arten von Multisensorsystemen (MSS), die der Erfassung der Umgebung dienen. In der vergangenen Dekade sind sehr viele MSS hinzugekommen, die auf einem Unmanned Aerial Vehicle montiert wurden. Diese MSS werden nachfolgend als Unmanned Aerial Systems (UAS) bezeichnet. Die Georeferenzierung von MSS und UAS ist ein notwendiger Schritt zur weiteren Datenverarbeitung. Die Georeferenzierung beinhaltet mindestens die Bestimmung von drei Translationen und drei Rotationen. Die erforderlichen Daten aus GNSS-Empfängern, inertialen Messsystemen oder anderen Sensoren zur Georeferenzierung sind in urbanem Umfeld nicht immer lückenlos und mit der erforderlichen Genauigkeit verfügbar. Deshalb werden in diesem Ansatz die Messungen UAS-basierter Laserscanner auf Gebäudefassaden verwendet. Letztere sind als Ebenen in einem 3D-Gebäudemodell modelliert. Die Trajektorie des UAS wird durch Kombination der Laserscanner-Messungen mit den Ebenenparametern ermittelt. Die daraus resultierenden impliziten Beobachtungsgleichungen und die nichtlinearen Restriktionsgleichungen werden innerhalb eines iterativen erweiterten Kalman-Filters (IEKF) modelliert. Außerdem wurde eine Softwaresimulation für den Test des IEKF entwickelt, um mit verschiedenen Szenarien die Funktionalität, Leistungsfähigkeit und verbleibende Herausforderungen zu bewerten.

✉ Johannes Bureick
bureick@gih.uni-hannover.de

[1] Geodetic Institute, Leibniz University Hannover, Nienburger Str. 1, 30167 Hannover, Germany

[2] Institute of Photogrammetry and GeoInformation, Leibniz University Hannover, Nienburger Str. 1, 30167 Hannover, Germany

1 Introduction

It is possible nowadays using the increasing capability and ongoing miniaturization of hardware and sensors to construct various kinds of multi-sensor systems (MSS). The MSS are frequently used in engineering geodesy to capture the environment. It is crucial to know the position and orientation of the MSS with respect to a superordinate coordinate system[1] for a further processing of the data captured. The task to determine position and orientation in a superordinate coordinate system is called georeferencing. Georeferencing usually comprises the determination of three translations and three rotations, also known as six Degrees of Freedom (DoF) or pose.

Unmanned aerial vehicles (UAV) have become a widespread, useful and affordable platform for the MSS in the past decade, optimally suited to capture common scenarios in engineering geodesy. The MSS mounted on a UAV are subsequently called unmanned aerial systems (UAS).

Precise georeferencing of a UAS is a challenging task, especially in urban areas. Global Navigation Satellite System (GNSS) data are affected by shadowing and multipath effects and are, therefore, inaccurate or even unavailable. Basically, low-cost GNSS receivers obtain positional accuracies of a few meters. These accuracies may be improved to decimeter or even centimeter level under good GNSS conditions and using satellite-based augmentation systems, differential GNSS, or real-time kinematics. Lightweight and low-cost inertial measurement units (IMU)[2] obtain orientation accuracies of 0.1° for roll and pitch and slightly worse accuracy, 0.8°, for heading. Combined GNSS-IMU systems improve the heading accuracy to 0.2°–0.5°. However, a general problem with IMU data is that it is often seriously affected by drifts.

A further possibility for georeferencing of an MSS is tracking using an external sensor. This is a cumbersome and inefficient task in the case of georeferencing a UAS due to limited measuring ranges and occultations.

In this paper, we present the georeferencing of a UAS, which is equipped with a three-dimensional (3D) laser scanner, a low-cost GNSS receiver, and an IMU, among other sensors. We set ourselves the requirement to determine the pose of the UAS with a precision (as a measure of the accuracy) better than 10 cm in position and 0.1° for orientation. The accuracy of the pose is decisive for the further results derived, such as digital terrain models or detailed 3D city models. Therefore, we fused the UAS-based laser scanner measurements towards building facades, which are modeled in a 3D city model, and the other sensor data within an iterated extended Kalman filter (IEKF). We developed a software simulation for testing the IEKF developed using different scenarios to evaluate the functionality and performance of the IEKF implemented. All results presented are part of a research project, subsequently called the *UAS project*.

1.1 Georeferencing of MSS/UAS

The strategies to accomplish the georeferencing vary depending on the measurement configuration, the sensors and hardware available, the environment, and the accuracies to be fulfilled. The georeferencing strategies can be generally classified into *direct*, *indirect*, and *data-driven* georeferencing (see, e.g., Schuhmacher and Böhm 2005; Paffenholz 2012; Holst et al. 2015). An alternative classification regarding indoor applications is given in Vogel et al. (2016).

For *direct*[3] georeferencing, sensors which measure the pose of an MSS directly are integrated into the MSS. These sensors could be, for example, a GNSS receiver (Paffenholz 2012; Talaya et al. 2004), an IMU (Talaya et al. 2004), or an external sensor, such as a total station or a laser tracker (Dennig et al. 2017; Hartmann et al. 2018), which determine the pose by angle and distance measurements to a reflector on the MSS.

Indirect[4] georeferencing comprises methods where the pose of the MSS is determined by measurements towards known targets. These targets may be flat markers with a specific pattern (Abmayr et al. 2008) or simple 3D geometries, such as cylinders or spheres (Elkhrachy and Niemeier 2006). The position of the targets within the superordinate coordinate system is determined using an external sensor, for example, a total station. Indirect georeferencing is commonly used in bundle adjustment or for the georeferencing of static terrestrial laser scanners (TLS).

Data-driven georeferencing conforms basically to the indirect georeferencing. Instead of known targets, the data sets are matched to reference data sets. These reference data sets may be point clouds georeferenced already (Soloviev et al. 2007; Glira et al. 2015), digital surface models or 3D city models (Hebel et al. 2009; Li-Chee-Ming and Armenakis 2013; Unger et al. 2016, 2017). The matching is accomplished, for instance, via point-to-point assignment, for example, an iterative closest point algorithm (Besl and McKay 1992) or point-to-object assignment (see, e.g., Schuhmacher and Böhm 2005). Please note that *data-driven* georeferencing approaches vary widely. A huge part can be found in the commonly used Simultaneous Localization and Mapping (SLAM) approaches (see, e.g., Nguyen et al. 2006; Lee et al. 2007; Jutzi et al. 2013; Kaul et al. 2015; Nüchter et al. 2015).

[1] Also called the world coordinate system.

[2] E.g. the GNSS-IMU system in SBG Systems (2019).

[3] Also called *sensor-driven*.

[4] Also called *target-based*.

In principle, all the georeferencing strategies mentioned are suitable for georeferencing static and kinematic MSS. For a static MSS, the six DoF have to be determined only once, whereas for a kinematic MSS, the six DoF for each measuring epoch should be determined. It may be indispensable to model the pose almost continuously depending on the measuring frequency of the sensors used.

1.2 Filtering Techniques for Georeferencing

Georeferencing of a kinematic MSS is commonly covered as part of the state parameter vector within a filtering approach. Filtering is a two-step procedure in which the current state parameter vector is estimated based on the previous state parameter vector and the current observations. In the first step, called prediction, the current state parameter vector is predicted using the previous state parameter vector and a specific system model. The system model takes into account all the controlling variables and environmental noises that affect the current state. In the second step, called the measurement update step, the predictions are modified based on the observation equations which compare the predicted observations with the current original ones.

In recent years, many pose estimation algorithms based on the extended Kalman filter (EKF) have been successfully applied to solve the pose estimation problem of a UAS. Tailanian et al. (2014) focus on the sensor fusion of the local sensors of a UAS in which the GNSS and the IMU have been combined by means of an EKF. In Hol et al. (2007), pose estimation on a six DOF robot using an EKF to fuse this information has been shown. Forster et al. (2013) use the collaboration of multiple UAS for pose estimation to combine multiple SLAM algorithms and create an accurate pose estimation. In their approach, real-time camera pose estimation is accomplished by combining the inertial and vision measurements using nonlinear state estimation approaches.

In the case of using nonlinear observation equations, a linearization (realized by means of Taylor series expansion) has to be applied to overcome the nonlinearity issue (see Denham and Pines 1966). In the case where the effects of the linearization errors tend to affect the efficiency of the filter or its convergence, the re-linearization of the measurement equation around the updated state may reduce these difficulties. Therefore, such a procedure is called IEKF. Researchers commonly use explicit observation equations in the IEKF. This means that the observations are considered as a function of the state parameters. Such an observation model is generally called a Gauss Markov model (GMM).

If the equations relating the observations to the state parameters are condition equations, for example, some 3D points should fulfill the plane equations, then we do not exhibit the typical formulation of the GMM. In other words, the observations and the state parameters are not separable (implicit measurement equations, see Dang 2007). In such a case, we are dealing with a Gauss Helmert model (GHM). In Dang (2007, 2008), Steffen and Beder (2007), Steffen (2013) and Ettlinger et al. (2018), an IEKF is used which deals with implicit measurement equations. In Vogel et al. (2018), an IEKF is used for georeferencing and extended with additional nonlinear equality constraints. In Vogel et al. (2019), the approach is further extended by integrating nonlinear inequality constraints. With the ability to handle implicit measurement equations and nonlinear inequality constraints, it is possible to depict almost any mathematical relationship within an IEKF.

1.3 Contribution

In this paper, we present an adaption of the IEKF in Vogel et al. (2018) for *data-driven* georeferencing of a UAS with an accuracy better than 10 cm in position and 0.1° in orientation. The highlight of the IEKF presented is the fusion of laser scanner measurements towards building facades, modeled as planes in a 3D city model, with the corresponding plane parameters of the 3D city model. This fusion leads to nonlinear implicit measurement equations and additional nonlinear equality constraints, which are covered within the IEKF. To evaluate the functionality and performance of the IEKF, we developed a simulation tool with a simple dynamic system model defining the state of the UAS and other states. On the basis of the states defined, we simulated sensor measurements regarding the sensor specifications and further characteristics. We tested the IEKF for different scenarios and analyzed its sensitivity towards different data characteristics.

1.4 Outline

The dedicated sections are as follows:

In Sect. 2 we give a detailed description of the UAS, the IEKF implemented and the entire workflow for georeferencing. Section 3 displays the simulation tool developed and gives an overview of the specifically chosen values for the IEKF. We present and discuss the results of the IEKF applied to the simulation in Sect. 4. Finally, conclusions and an outlook are drawn in Sect. 5.

2 IEKF for Georeferencing of a UAS

The Geodetic Institute (GIH) and the Institute of Photogrammetry and GeoInformation (IPI) of Leibniz University Hannover (LUH) are currently working on the *UAS project*. The *UAS project* deals with the precise determination of the trajectory of a UAS by integrating camera and laser scanner data in combination with generalized object information. Within this paper, we will focus on the usage of laser scanner data in combination with a non-generalized 3D city model. We will exclusively use simulated data to evaluate the functionality and performance of the algorithms developed. The simulation scenarios were chosen in accordance with common measurement scenarios and data characteristics.

2.1 General Idea

The UAS moves through an urban area where the building facades are modeled as planes within a 3D city model. Three-dimensional city models with a level of detail 1 (block model) and a level of detail 2 (model with differentiated roof structures and boundary surfaces) are freely available for many cities. A detailed classification of the different levels of detail is given in Gröger et al. (2012). The pose of the UAS is roughly known, for example, from measurements of a GNSS receiver and an IMU. A 3D laser scanner captures the environment (red dots in Fig. 1) continuously. These captured 3D points may represent the ground, vegetation, building facades, or other objects. The laser scanner measurements are given in the local sensor coordinate system. The laser scanner measurements are transformed in a superordinate coordinate system according to the roughly known translation $\boldsymbol{t} = \left[t_x, t_y, t_z\right]$ and orientation $\boldsymbol{o} = [\omega, \phi, \kappa]$ (see Fig. 2). The transformed laser scanner measurements are assigned to planes of the 3D city model based on the distance between the scanner points and the planes of the building model (see Sect. 2.3). Only the points that are close enough to a plane of the 3D city model (green points in Fig. 2) are used as observations afterwards. The final pose parameters are estimated within a GHM, resp. filtering approach, by minimizing the distance between the assigned laser scanner measurements and the planes of the 3D city model to which they are assigned.

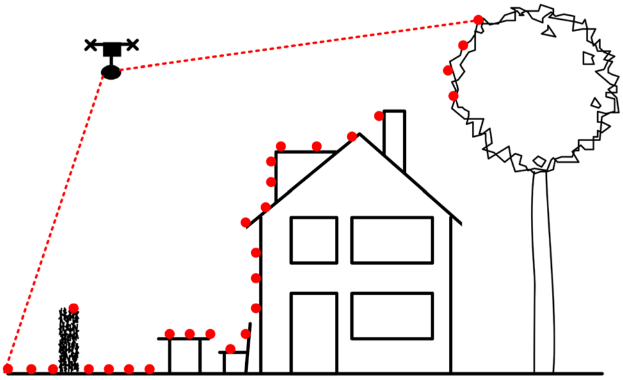

Fig. 1 Schematic depiction of the UAS capturing the environment. The points measured (red dots) may represent the ground, vegetation, building facades, or other objects

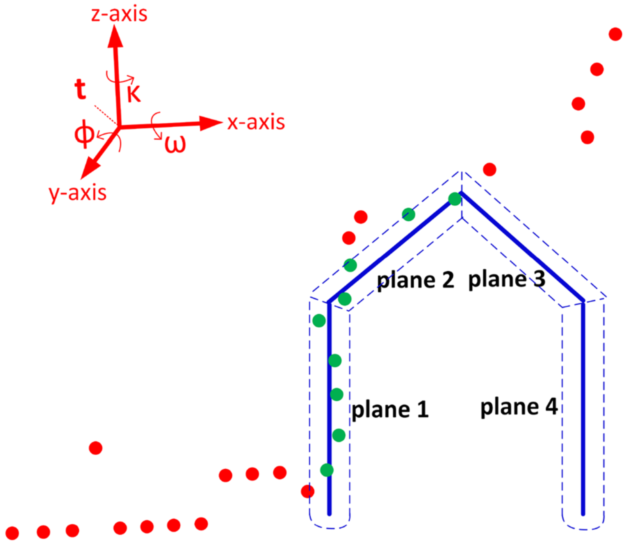

Fig. 2 Schematic depiction of the assignment of the transformed measured points. Some building facades are represented as planes in a generalized 3D city model (solid blue lines). The laser scanner measurements are transformed in the superordinate coordinate system according to roughly known translation $\boldsymbol{t} = \left[t_x, t_y, t_z\right]$ and orientation $\boldsymbol{o} = [\omega, \phi, \kappa]$. The origin of the laser scanner is depicted in the upper left corner. Note the slight difference to the original measurement configuration due to the roughly known pose. The transformed points which are close to one of the planes are assigned to that plane (green dots). The closeness is determined by a distance threshold (dashed blue lines). In this example, 5 points are assigned to plane 1 and 4 points are assigned to plane 2, whereas the other points (red dots) are not assigned. Subsequently, only the green dots are used as observations in the GHM, resp. filtering approach

2.2 UAS

A UAV is equipped with a 3D laser scanner, cameras, a GNSS receiver and an IMU in the *UAS project*. Figure 3 depicts a simplified sketch of the platform setup (without cameras). The laser scanner[5] scans 16 scan lines which are nearly perpendicular to the sensors' vertical axis (that corresponds to the z-axis in Fig. 3). The divergence between each scan line is 2°. Thus, the laser scanner has a field of view of 30° × 360°. It is possible to set the resolution of the points in the scan lines between 0.1° and 0.4°. The rotation rate depends on that setting and is between 5 Hz and 20 Hz. We set the resolution to 0.4° to obtain a higher rotation rate of 20 Hz. Furthermore, it is possible to exclude certain angle areas from the measurement, because they cannot provide any data or the data generated are not needed for further processing. In our case, it would make sense to exclude the angle areas where the laser scanner measures towards the UAV.

[5] Conforms to the Velodyne LiDAR PUCK VLP-16.

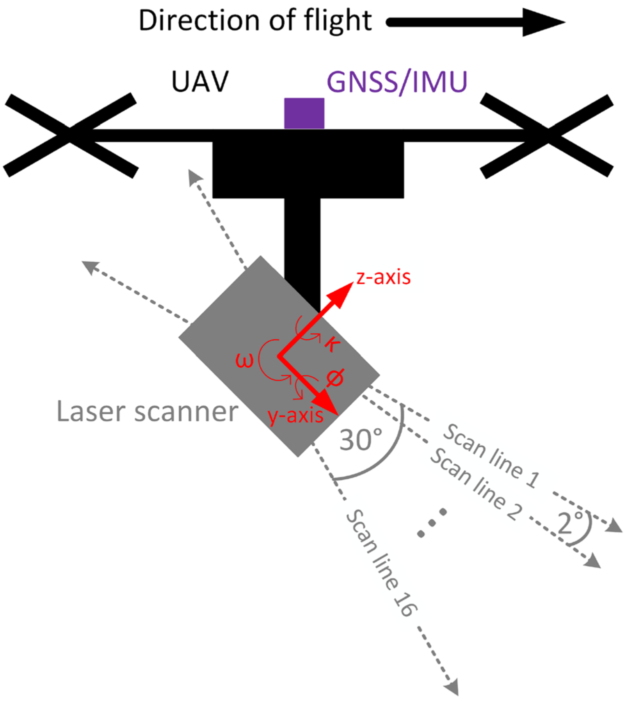

Fig. 3 Sketch of the platform setup. The *x*-axis of the local laser scanner coordinate system (red) points in readers line of sight

2.3 Assignment Algorithm

A crucial task of our georeferencing process is the assignment of measured 3D points $\boldsymbol{C}^{\mathrm{loc}}$, given in a local coordinate system of the laser scanner, to a plane of the city model, given in the global coordinate system. Therefore, we first have to transform $\boldsymbol{C}^{\mathrm{loc}}$ to $\boldsymbol{C}^{\mathrm{glo}}$:

$$\boldsymbol{C}^{\mathrm{glo}} = \boldsymbol{t} + \boldsymbol{R} \cdot \boldsymbol{C}^{\mathrm{loc}}. \tag{1}$$

The parameters of the translation $\boldsymbol{t}$ describe the position of the origin of the local coordinate system. The rotation matrix $\boldsymbol{R}$ is obtained based on the orientation $\boldsymbol{o} = [\omega, \phi, \kappa]$ of the local coordinate system according to Luhmann (2013):

$$\boldsymbol{R}_{\omega} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{bmatrix} \tag{2}$$

$$\boldsymbol{R}_{\phi} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \tag{3}$$

$$\boldsymbol{R}_{\kappa} = \begin{bmatrix} \cos\kappa & -\sin\kappa & 0 \\ \sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

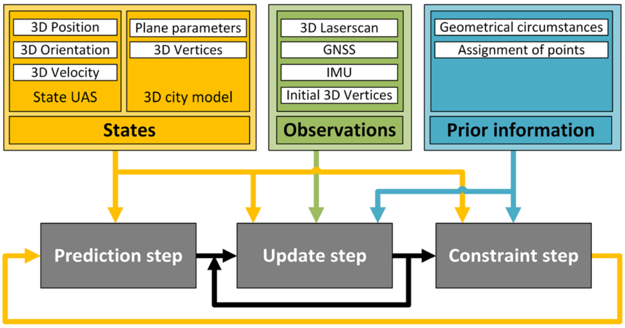

Fig. 4 Simplified basic workflow of the IEKF (grey) for georeferencing of a UAS by means of unknown states (yellow), available observations (green), and known prior information (blue)

$$\boldsymbol{R} = \boldsymbol{R}_{\omega} \cdot \boldsymbol{R}_{\phi} \cdot \boldsymbol{R}_{\kappa}. \tag{5}$$

The assignment is realized by a simple distance criterion, as described in Unger et al. (2016, 2017). The Euclidean distance to each plane of the city model is calculated for each 3D point in $\boldsymbol{C}^{\mathrm{glo}}$. Whether the projection of the points into the plane lies within the bounding polygon of the planes is checked for points where the distance to the nearest plane is less than a threshold value d_{assign}. If it is outside, the distance from the point to the boundary polygon of the plane is calculated and replaces the orthogonal distance. Points that project inside the boundary or are closer than d_{assign} are assigned to their closest plane. d_{assign} has to be selected regarding the accuracy of the points as well as the accuracy of the city model. The accuracy of the points depends mainly on the accuracy of the translation and orientation parameters $\boldsymbol{t}$ and $\boldsymbol{o}$ and the accuracy of the measured points $\boldsymbol{C}^{\mathrm{loc}}$. The accuracy of the city model depends on the geometrical accuracy of its vertices and the extent of generalization effects by which the model deviates from the reality captured. The larger the threshold is fixed, the more points are assigned to planes, but the higher the probability is that incorrect assignments will be set that lead to incorrect results.

2.4 IEKF (Iterated Extended Kalman Filter)

The IEKF which we adapted for georeferencing of the UAS is given in Vogel et al. (2018). This approach can be applied for many different use cases. All available observations, types of prior information, and requested states need to be linked with the respective uncertainty information. Based on this, all information can be introduced into the IEKF.

According to the respective application, several relationships have to be established by means of linear or rather nonlinear functional models. A system model $\boldsymbol{f}(\cdot)$ is needed for the requested states $\boldsymbol{x}_k$ to describe the physical behavior of the system from epoch $k-1$ to the current epoch k. Realization of this transfer is carried out within the prediction step of the IEKF. Suitable functional relationships for each observation $\boldsymbol{l}_k$ have to be formulated within the measurement model $\boldsymbol{h}(\cdot)$. Based on implicit ($\boldsymbol{h}(E(\boldsymbol{l}), \boldsymbol{x}) = \boldsymbol{0}$) or explicit ($E(\boldsymbol{l}) - \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}$) formulations, relations between the observations available and requested states are considered. $E(\cdot)$ is the expected operator of a random vector; here, the expected value of the observation vector $\boldsymbol{l}$, which can be replaced by $E(\boldsymbol{l}) = \boldsymbol{l} + \boldsymbol{v}$. This consideration of current observations is carried out within the measurement update step. Available prior information can also be integrated within the measurement model during the measurement update step. In addition, further suitable prior information can also be applied in terms of state constraints by means of additional linear ($\boldsymbol{D}\boldsymbol{x}_k = \boldsymbol{d}$) or nonlinear ($g\left(\boldsymbol{x}_k\right) = \boldsymbol{b}$) functions. Here, $\boldsymbol{D}$ is a known matrix and $\boldsymbol{d}$ and $\boldsymbol{b}$ are known vectors with respect to related state constraints. Formulation of such restrictions can be implemented by means of equality constraints within the constraint step.

The basic workflow of this IEKF is depicted in Fig. 4 in a simplified way. Detailed equations for initialization, prediction, measurement update, and constraints are given in the following Sects. 2.4.1–2.4.6. Algorithm 1 given in Sect. 2.5 summarizes the required input, different computations steps and the output.

2.4.1 State Parameter Vector

The state parameter vector $\boldsymbol{x}_k$ consists of two parts, as already depicted in Fig. 4. The first part $\boldsymbol{x}_{\mathrm{State},k} \in \mathbb{R}^9$ consists of the current position $\boldsymbol{t}_k$, orientation $\boldsymbol{o}_k$ and velocity $\boldsymbol{v}_k$ of the UAS (see Eq. 6). These state parameters describe the state of the UAS.

The second part consists, on one hand, of the vector $\boldsymbol{x}_{\mathrm{Plane},k} \in \mathbb{R}^{4\cdot E}$, which contains the parameters of all the E city model planes in Hesse normal form (see Eq. 7). The Hesse normal form is defined by the 3×1 normal vector $\boldsymbol{n}_e = [n_{x,e}, n_{y,e}, n_{z,e}]^T$ and the distance to the origin d_e with $e = 1 \dots E$. On the other hand, it consists of the vector $\boldsymbol{x}_{V,k} \in \mathbb{R}^{3\cdot M}$, which contains all the M vertices of the building model (see Eq. 8):

$$\boldsymbol{x}_{\mathrm{State},k} = \left[\underbrace{t_{x,k}, t_{y,k}, t_{z,k}}_{\boldsymbol{t}_k}, \underbrace{\omega_k, \phi_k, \kappa_k}_{\boldsymbol{o}_k}, \underbrace{v_{x,k}, v_{y,k}, v_{z,k}}_{\boldsymbol{v}_k}\right]^T \tag{6}$$

$$\boldsymbol{x}_{\mathrm{Plane},k} = \left[\underbrace{\boldsymbol{n}_{1,k};d_{1,k}}_{\text{plane 1}}; \cdots ; \underbrace{\boldsymbol{n}_{E,k};d_{E,k}}_{\text{plane } E}\right] \tag{7}$$

$$\boldsymbol{x}_{V,k} = \left[\boldsymbol{V}^{\mathrm{glo}}_{1,k}; \dots ;\boldsymbol{V}^{\mathrm{glo}}_{m,k}; \dots ;\boldsymbol{V}^{\mathrm{glo}}_{M,k}\right] \tag{8}$$

with

$$\boldsymbol{V}^{\mathrm{glo}}_{m,k} = \left[V^{\mathrm{glo}}_{x,m,k}, V^{\mathrm{glo}}_{y,m,k}, V^{\mathrm{glo}}_{z,m,k}\right]^T. \tag{9}$$

The integration of plane parameters and vertices into the state parameter vector is used to identify and correct planes that are not accurately represented in the 3D city model. Although this purpose is not part of this paper, we are already introducing the mathematical relationship.

The complete state parameter vector $\boldsymbol{x}_k$ is arranged according to:

$$\boldsymbol{x}_k = \left[\boldsymbol{x}_{\mathrm{State},k}; \boldsymbol{x}_{\mathrm{Plane},k}; \boldsymbol{x}_{V,k}\right]. \tag{10}$$

2.4.2 Observation Vector

The MSS mentioned in Sect. 2.2 provides discrete 3D laser scanner point clouds (LSC) and 6D pose information by means of a GNSS receiver and an IMU. An LSC consists of a full scan rotation (30° × 360°). For the sake of simplification, we will not consider the movement of the UAV during the period of a full scan rotation in the following. This time period is a maximum of 0.05 s for the configuration chosen. It is possible to exclude certain angle areas of the laser scanner, as has already been described in Sect. 2.2. The time period for the non-excluded areas is further reduced by excluding the angular areas in which the laser scanner measures in the direction of the UAV. This is acceptable due to the planned velocity of the UAV of about 1 m s^{-1} and an angular velocity of 2° s^{-1}. In further development steps, the LSC should consist of fewer and temporally closer 3D points, for example, just a half or a quarter scan rotation. In addition, the initial vertices of the 3D city model are introduced as observations into our approach. The observation vector $\boldsymbol{l}_k$ consists of three parts for each epoch $k = 1 \dots K$. The first part $\boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},k}$ consists of the measured LSC in the local sensor coordinate system with a total of N 3D points. $\boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},k}$ only contains the 3D points, which were assigned to a plane of the city model[6] (for the assignment algorithm, see Sect. 2.3). These points are assorted on the basis of the planes they are assigned to:

$$\boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},k} = \left[\boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},1,k}; \dots ; \boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},e,k}; \dots ; \boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},E,k}\right] \tag{11}$$

with

$$\boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},e,k} = \left[\boldsymbol{p}^{\mathrm{loc}}_{e,1,k}; \dots ; \boldsymbol{p}^{\mathrm{loc}}_{e,i,k}; \dots ; \boldsymbol{p}^{\mathrm{loc}}_{e,N_e,k}\right] \tag{12}$$

and

$$\boldsymbol{p}^{\mathrm{loc}}_{e,i,k} = \left[X^{\mathrm{loc}}_{e,i,k}, Y^{\mathrm{loc}}_{e,i,k}, Z^{\mathrm{loc}}_{e,i,k}\right]^T. \tag{13}$$

[6] In photogrammetry, tie points are observed in more than one image and, therefore, connect poses. By contrast, points measured by the laser scanner are only observed from one pose and do not, therefore, contribute to pose estimation in this scenario as long as they are not related to a model plane.

$\boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},e,k}$ is representing the vector with all N_e points assigned to plane e. The total number N of 3D points of the LSC stored in the observation vector is calculated by:

$$N = \sum_{e=1}^{E} N_e. \tag{14}$$

The second part $\boldsymbol{l}^{\mathrm{glo}}_{\mathrm{Pose},k}$ consists of the direct GNSS and IMU observations for the position and orientation:

$$\boldsymbol{l}^{\mathrm{glo}}_{\mathrm{Pose},k} = \left[\underbrace{t^{\mathrm{GNSS}}_{x,k}, t^{\mathrm{GNSS}}_{y,k}, t^{\mathrm{GNSS}}_{z,k}}_{\mathbf{t}^{\mathrm{GNSS}}_{k}}, \underbrace{\omega^{\mathrm{IMU}}_{k}, \phi^{\mathrm{IMU}}_{k}, \kappa^{\mathrm{IMU}}_{k}}_{\mathbf{o}^{\mathrm{IMU}}_{k}} \right]^{T}. \tag{15}$$

Finally, the third part $\boldsymbol{l}^{\mathrm{glo}}_{V,0}$ consists of the, in total M, initial vertices of the E model planes[7]:

$$\boldsymbol{l}^{\mathrm{glo}}_{V,0} = \left[\boldsymbol{V}^{\mathrm{glo}}_{1,0}; \dots ; \boldsymbol{V}^{\mathrm{glo}}_{m,0}; \dots ; \boldsymbol{V}^{\mathrm{glo}}_{M,0} \right] \tag{16}$$

with

$$\boldsymbol{V}^{\mathrm{glo}}_{m,0} = \left[V^{\mathrm{glo}}_{x,m,0}, V^{\mathrm{glo}}_{y,m,0}, V^{\mathrm{glo}}_{z,m,0} \right]^{T} \tag{17}$$

representing a 3D point which is the vertex of at least one plane.

The observation vector is arranged as follows:

$$\boldsymbol{l}_k = \left[\boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},k}; \boldsymbol{l}^{\mathrm{glo}}_{\mathrm{Pose},k}; \boldsymbol{l}^{\mathrm{glo}}_{V,0} \right]. \tag{18}$$

We apply Eq. (19) for building the stochastic model of the observation vector. We neglect correlations in all cases for a better discussion of the mainly important issues:

$$\boldsymbol{\Sigma}_{ll,k} = \mathrm{diag}\left(\underbrace{\sigma^2_{\mathrm{LS}}, \dots, \sigma^2_{\mathrm{LS}}}_{3\cdot N}, \underbrace{\boldsymbol{\sigma}^2_{t}, \boldsymbol{\sigma}^2_{o}}_{6}, \underbrace{\sigma^2_{V}, \dots, \sigma^2_{V}}_{3\cdot M} \right). \tag{19}$$

The standard deviation of a laser scanner coordinate component is denoted by σ_{LS}. $\boldsymbol{\sigma}_t = \left[\sigma_t, \sigma_t, \sigma_t\right]$ and $\boldsymbol{\sigma}_o = \left[\sigma_o, \sigma_o, \sigma_o\right]$ denote the standard deviation of the GNSS and IMU observations, and σ_V denotes the standard deviation of the initial vertices.

2.4.3 System Equation

Formulating the system equation of the kind:

$$\boldsymbol{x}_k = \boldsymbol{f}\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}\right) + \boldsymbol{w}_{k-1}, \tag{20}$$

we neglected the control vector $\boldsymbol{u}$ by setting it to zero. The system noise is normally distributed with $\boldsymbol{w}_{k-1} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_{ww})$. Equation (20) can be formulated as:

$$\boldsymbol{x}_k = \boldsymbol{F}_{x,k} \cdot \boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \tag{21}$$

where $\boldsymbol{F}_{x,k}$ denotes the transition matrix:

$$\boldsymbol{F}_{x,k} = \partial f / \partial \boldsymbol{x}|_{\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1}}. \tag{22}$$

Due to the fact that we do not intend to develop an optimal system model within this paper, we applied rather simple linear system equations. Nevertheless, a subsequent adaption of the system model with more complex system equations is easily possible if the future flight characteristics of the UAS require it. For the system equations chosen, the transition matrix $\boldsymbol{F}_{x,k}$ is given by:

$$\boldsymbol{F}_{x,k} = \begin{bmatrix} \boldsymbol{F}_{x,\mathrm{State},k[9\times 9]} & \mathbf{0}_{[9\times 4\cdot E]} & \mathbf{0}_{[9\times 3\cdot M]} \\ \mathbf{0}_{[4\cdot E\times 9]} & \boldsymbol{I}_{[4\cdot E\times 4\cdot E]} & \mathbf{0}_{[4\cdot E\times 3\cdot M]} \\ \mathbf{0}_{[3\cdot M\times 9]} & \mathbf{0}_{[3\cdot M\times 4\cdot E]} & \boldsymbol{I}_{[3\cdot M\times 3\cdot M]} \end{bmatrix}, \tag{23}$$

with

$$\boldsymbol{F}_{x,\mathrm{State},k} = \begin{bmatrix} \boldsymbol{I}_{[3\times 3]} & \mathbf{0}_{[3\times 3]}\mathrm{diag}([\Delta\tau, \Delta\tau, \Delta\tau])_{[3\times 3]} \\ \mathbf{0}_{[6\times 3]} & \boldsymbol{I}_{[6\times 6]} \end{bmatrix}. \tag{24}$$

$\Delta\tau$ is the time period between two consecutive epochs.

We apply the following equations for building the variance–covariance matrix (VCM) of the system noise $\boldsymbol{\Sigma}_{ww}$:

$$\boldsymbol{\Sigma}_{ww,\mathrm{State}} = \mathrm{diag}\left(\boldsymbol{\sigma}^2_{t,w}, \boldsymbol{\sigma}^2_{o,w}, \boldsymbol{\sigma}^2_{v,w} \right) \tag{25}$$

$$\boldsymbol{\Sigma}_{ww,\mathrm{Plane}} = \mathrm{diag}\left(\boldsymbol{\sigma}^2_{n,w}, \sigma^2_{d,w}, \dots, \boldsymbol{\sigma}^2_{n,w}, \sigma^2_{d,w} \right) \tag{26}$$

$$\boldsymbol{\Sigma}_{ww,V} = \mathrm{diag}\left(\sigma^2_{V,w}, \dots, \sigma^2_{V,w} \right) \tag{27}$$

$$\boldsymbol{\Sigma}_{ww} = \begin{bmatrix} \boldsymbol{\Sigma}_{ww,\mathrm{State}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{ww,\mathrm{Plane}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{ww,V} \end{bmatrix}. \tag{28}$$

The standard deviations of the system noise for translation, orientation, and velocity are denoted by $\boldsymbol{\sigma}_{t,w} = \left[\sigma_{t,w}, \sigma_{t,w}, \sigma_{t,w}\right]$, $\boldsymbol{\sigma}_{o,w} = \left[\sigma_{o,w}, \sigma_{o,w}, \sigma_{o,w}\right]$ and $\boldsymbol{\sigma}_{v,w} = \left[\sigma_{v,w}, \sigma_{v,w}, \sigma_{v,w}\right]$. The system noise for the plane parameters and vertices is given by the standard deviations $\boldsymbol{\sigma}_{n,w} = \left[\sigma_{n,w}, \sigma_{n,w}, \sigma_{n,w}\right]$, $\sigma_{d,w}$ and $\sigma_{V,w}$.

[7] Please note the missing k in the index, which implies that this observation remains unchanged over the epochs.

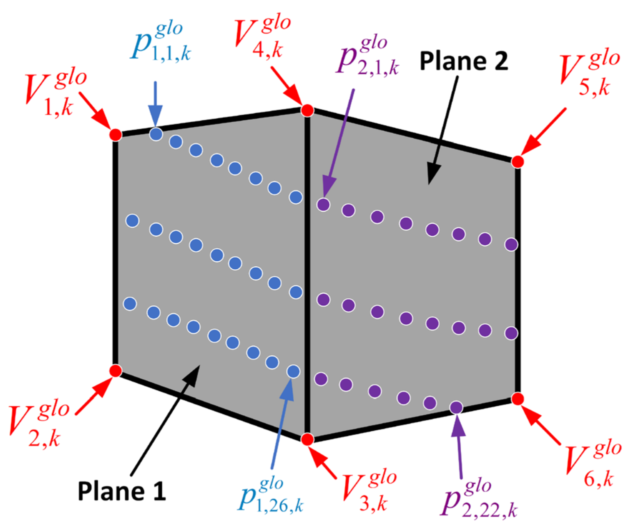

Fig. 5 Example of the relationship between state parameters, observations, and measurement equations in an arbitrary epoch k: The $N_1 = 26$ blue points $\boldsymbol{p}^{\mathrm{glo}}_{1,1,k}$ to $\boldsymbol{p}^{\mathrm{glo}}_{1,26,k}$ are assigned to plane 1. Subsequently, $\boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},1,k}$ consists of these blue points in local sensor coordinate system. The $N_2 = 22$ purple points $\boldsymbol{p}^{\mathrm{glo}}_{2,1,k}$ to $\boldsymbol{p}^{\mathrm{glo}}_{2,22,k}$ are assigned to plane 2. Subsequently, $\boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},2,k}$ consists of these purple points in local sensor coordinate system. If we assume that there are only those $E = 2$ planes in the 3D city model, the observation vector $\boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},k}$ has the dimension $[144 \times 1]$. According to Eq. (14), N is equal to 48. The state parameter vector $\boldsymbol{x}_{V,k}$ consists of the six vertices $\boldsymbol{V}^{\mathrm{glo}}_{1,k}$ to $\boldsymbol{V}^{\mathrm{glo}}_{6,k}$ (in global coordinate system) and has the dimension $[18 \times 1]$. The vector $\boldsymbol{l}^{\mathrm{glo}}_{V,0}$ has the same dimension. With solely two planes in the 3D city model, $\boldsymbol{x}_{\mathrm{Plane},k}$ has the dimension $[8 \times 1]$

2.4.4 Measurement Equation

With the state parameter and observation vector given in Sects. 2.4.1 and 2.4.2, we can formulate three types of nonlinear implicit measurement equations. The first type of measurement equation $\boldsymbol{h}_{\mathrm{I}}$ causes that each transformed LSC $\boldsymbol{l}^{\mathrm{glo}}_{\mathrm{Scan},e,k}$ has to be located in the assigned plane of the city model represented by $\boldsymbol{n}_e$ and d_e. If the point $\boldsymbol{p}^{\mathrm{loc}}_{e,i,k} = \left[X^{\mathrm{loc}}_{e,i,k}, Y^{\mathrm{loc}}_{e,i,k}, Z^{\mathrm{loc}}_{e,i,k}\right]^T$ is in $\boldsymbol{l}^{\mathrm{loc}}_{\mathrm{Scan},e,k}$, which consists of all points assigned to plane e of the 3D city model, the following measurement equation is applied:

$$h_{\mathrm{I},i}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) \;:\; \boldsymbol{n}_e^T \cdot \underbrace{\left[\boldsymbol{t}_k + \boldsymbol{R}_k \cdot E(\boldsymbol{p}^{\mathrm{loc}}_{e,i,k})\right]}_{\boldsymbol{p}^{\mathrm{glo}}_{e,i,k}} - d_e = 0, \tag{29}$$

with $i \in \{1 \ldots N\}$.

The rotation matrix $\boldsymbol{R}_k$ is calculated based on the current orientation $\boldsymbol{o}_k$ according to Eqs. (2)–(5). Subsequently, we obtain N measurement equations of type $\boldsymbol{h}_{\mathrm{I}}$:

$$\boldsymbol{h}_{\mathrm{I}}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) = \left[h_{\mathrm{I},1}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right), \ldots, h_{\mathrm{I},N}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right)\right]^T. \tag{30}$$

Figure 5 shows an illustration of the relationship between state parameters, observations, and measurement equations.

In the second type of measurement equation $\boldsymbol{h}_{\mathrm{II}}$, we calculate the difference between the estimated pose $\left[\boldsymbol{t}_k; \boldsymbol{o}_k\right]$ and observed pose $\boldsymbol{l}^{\mathrm{glo}}_{\mathrm{Pose},k}$ of GNSS and IMU:

$$\boldsymbol{h}_{\mathrm{II}}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) \;:\; \left[\boldsymbol{t}_k; \boldsymbol{o}_k\right] - E(\boldsymbol{l}^{\mathrm{glo}}_{\mathrm{Pose},k}) = \boldsymbol{0}. \tag{31}$$

If the GNSS and IMU data are available, we obtain six measurement equations of type $\boldsymbol{h}_{\mathrm{II}}$. When the GNSS signal is missing, the measurement equation changes to:

$$\boldsymbol{h}_{\mathrm{II}}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) \;:\; \boldsymbol{o}_k - E(\boldsymbol{o}^{\mathrm{IMU}}_k) = \boldsymbol{0}, \tag{32}$$

and we only obtain three measurement equations of this type.

The third type of measurement equation $\boldsymbol{h}_{\mathrm{III}}$ is introduced to avoid a datum defect. It computes the difference between the estimated vertices in $\boldsymbol{x}_{V,k}$ and the initial vertices in $\boldsymbol{l}^{\mathrm{glo}}_{V,0}$. If the vertex $\boldsymbol{V}^{\mathrm{glo}}_{m,k}$ is stored in $\boldsymbol{x}_{V,k}$ and $\boldsymbol{V}^{\mathrm{glo}}_{m,0}$ is its initial observation stored in $\boldsymbol{l}^{\mathrm{glo}}_{V,0}$, the following measurement equations are applied:

$$\boldsymbol{h}_{\mathrm{III},m}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) \;:\; \boldsymbol{V}^{\mathrm{glo}}_{m,k} - E(\boldsymbol{V}^{\mathrm{glo}}_{m,0}) = \boldsymbol{0}, \tag{33}$$

with $m \in \{1 \ldots M\}$.

For a three-dimensional vertex Eq. (33) consists of three measurement equations. Subsequently, we obtain $3 \cdot M$ measurement equations of type $\boldsymbol{h}_{\mathrm{III}}$:

$$\boldsymbol{h}_{\mathrm{III}}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) = \begin{bmatrix} \boldsymbol{h}_{\mathrm{III},1}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) \\ \vdots \\ \boldsymbol{h}_{\mathrm{III},M}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) \end{bmatrix}. \tag{34}$$

Altogether, the measurement equations are given by:

$$\boldsymbol{h}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) = \begin{bmatrix} \boldsymbol{h}_{\mathrm{I}}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) \\ \boldsymbol{h}_{\mathrm{II}}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) \\ \boldsymbol{h}_{\mathrm{III}}\left(E(\boldsymbol{l}_k), \boldsymbol{x}_k\right) \end{bmatrix}. \tag{35}$$

2.4.5 Nonlinear Equality Constraint for the State Parameters

We apply two types of nonlinear equality constraints for the state parameters in our IEKF. The prior information which we want to model describes hard constraints which have to be fulfilled.

The first type of nonlinear equality constraint $\boldsymbol{g}_{\mathrm{I}}$ arises due to the fact that we are using plane parameters within the state parameter vector $\boldsymbol{x}_k$. Here, we must ensure unit normal vectors by means of a length of one. For this, we can make use of the nonlinear equality constraints:

$$g_{\mathrm{I},e}\left(\boldsymbol{x}_k\right) = ||\boldsymbol{n}_e|| = \sqrt{n_{e_x}^2 + n_{e_y}^2 + n_{e_z}^2} = b_{\mathrm{I},e} = 1, \tag{36}$$

with $e \in \{1 \dots E\}$.

The nonlinear equality constraint of type $\boldsymbol{g}_{\text{I}}$ applies to each of the E planes normal vectors:

$$\boldsymbol{g}_{\text{I}}(\boldsymbol{x}_k) = \left[g_{\text{I},1}(\boldsymbol{x}_k), \dots, g_{\text{I},E}(\boldsymbol{x}_k)\right]^T. \tag{37}$$

The right side of the equal sign is stored in vector $\boldsymbol{b}_{\text{I}}$:

$$\boldsymbol{b}_{\text{I}} = \left[b_{\text{I},1}, \dots, b_{\text{I},E}\right]^T. \tag{38}$$

With the second type of nonlinear equality constraint $\boldsymbol{g}_{\text{II}}$, we ensure that each vertex of a plane is located in the plane. By this, we preserve the topology of the 3D city model. If we assume that $\boldsymbol{V}^{\text{glo}}_{m,k}$ is a vertex of plane e, represented by $\boldsymbol{n}_e$ and d_e, the following equality constraint is applied:

$$g_{\text{II},i}(\boldsymbol{x}_k) = \boldsymbol{n}_e^T \cdot \boldsymbol{V}^{\text{glo}}_{m,k} - d_e = \boldsymbol{b}_{\text{II},i} = 0 \tag{39}$$

with $i \in \{1 \dots 4 \cdot E\}$.

This type of nonlinear equality constraint must be fulfilled for each vertex of all planes. In general, 3D city model planes have 4 vertices, but it is also possible for them to have 3 or more than 4 vertices. Thus, the number $4 \cdot E$ is just a rough estimate for the total amount of equality constraints of type $\boldsymbol{g}_{\text{II}}$:

$$\boldsymbol{g}_{\text{II}}(\boldsymbol{x}_k) = \left[g_{\text{II},1}(\boldsymbol{x}_k), \dots, g_{\text{II},4\cdot E}(\boldsymbol{x}_k)\right]^T. \tag{40}$$

The right side of the equal sign is stored in vector $\boldsymbol{b}_{\text{II}}$:

$$\boldsymbol{b}_{\text{II}} = \left[b_{\text{II},1}, \dots, b_{\text{II},4\cdot E}\right]^T. \tag{41}$$

A comparable procedure can be found in Unger et al. (2016). Altogether, the nonlinear equality constraints for the state parameters are given by:

$$\boldsymbol{g}(\boldsymbol{x}_k) = \left[\boldsymbol{g}_{\text{I}}(\boldsymbol{x}_k); \boldsymbol{g}_{\text{II}}(\boldsymbol{x}_k)\right] \tag{42}$$

$$\boldsymbol{b} = \left[\boldsymbol{b}_{\text{I}}; \boldsymbol{b}_{\text{II}}\right]. \tag{43}$$

2.4.6 Initialization

The initial state parameter vector $\boldsymbol{x}_0$ is created by the GNSS and IMU observations. We assume zero for the velocity in each coordinate component:

$$\boldsymbol{v}_0 = \boldsymbol{0}_{[3\times1]}. \tag{44}$$

The initial plane parameters $\boldsymbol{n}_{e,0}$ and $d_{e,0}$ were estimated from the planes' vertices stored in the city model by the Drixler algorithm (Drixler 1993). We make use of the vertices of the 3D city model given for the initial vertices $\boldsymbol{x}_{V,0}$:

$$\boldsymbol{x}_{\text{State},0} = \left[\boldsymbol{t}_{\text{GNSS},0}; \boldsymbol{o}_{\text{IMU},0}; \boldsymbol{v}_0\right] \tag{45}$$

$$\boldsymbol{x}_{\text{Plane},0} = \left[\boldsymbol{n}_{1,0}; d_{1,0}; \dots; \boldsymbol{n}_{E,0}; d_E\right] \tag{46}$$

$$\boldsymbol{x}_{V,0} = \left[\boldsymbol{V}^{\text{glo}}_{1,0}; \dots; \boldsymbol{V}^{\text{glo}}_{M,0}\right] \tag{47}$$

$$\boldsymbol{x}_0 = \left[\boldsymbol{x}_{\text{State},0}; \boldsymbol{x}_{\text{Plane},0}; \boldsymbol{x}_{V,0}\right]. \tag{48}$$

We apply the following equations for building the VCM of the initial state parameter vector $\boldsymbol{\Sigma}_{xx,0}$:

$$\boldsymbol{\Sigma}_{xx,\text{State},0} = \text{diag}\left(\boldsymbol{\sigma}^2_{t,0}, \boldsymbol{\sigma}^2_{o,0}, \boldsymbol{\sigma}^2_{v,0}\right) \tag{49}$$

$$\boldsymbol{\Sigma}_{xx,\text{Plane},0} = \text{diag}\left(\boldsymbol{\sigma}^2_{n,0}, \sigma^2_{d,0}, \dots, \boldsymbol{\sigma}^2_{n,0}, \sigma^2_{d,0}\right) \tag{50}$$

$$\boldsymbol{\Sigma}_{xx,V,0} = \text{diag}\left(\sigma^2_{V,0} \dots, \sigma^2_{V,0}\right) \tag{51}$$

$$\boldsymbol{\Sigma}_{xx,0} = \begin{bmatrix} \boldsymbol{\Sigma}_{xx,\text{State},0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{xx,\text{Plane},0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Sigma}_{xx,V,0} \end{bmatrix}. \tag{52}$$

The standard deviations of the initial state parameters of translation, orientation, and velocity are denoted by $\boldsymbol{\sigma}_{t,0} = \left[\sigma_{t,0}, \sigma_{t,0}, \sigma_{t,0}\right]$, $\boldsymbol{\sigma}_{o,0} = \left[\sigma_{o,0}, \sigma_{o,0}, \sigma_{o,0}\right]$, and $\boldsymbol{\sigma}_{v,0} = \left[\sigma_{v,0}, \sigma_{v,0}, \sigma_{v,0}\right]$. The standard deviations of the initial plane parameters and vertices are denoted by $\boldsymbol{\sigma}_{n,0} = \left[\sigma_{n,0}, \sigma_{n,0}, \sigma_{n,0}\right]$, $\sigma_{d,0}$ and $\sigma_{V,0}$.

2.5 Workflow

The workflow of our algorithm is summarized in Algorithm 1. This is an adaption of the algorithm proposed in Vogel et al. (2018). We highlighted the cross references to equations and sections in blue. The partial derivatives (see lines 9, 22, 23, and 40) were obtained once using

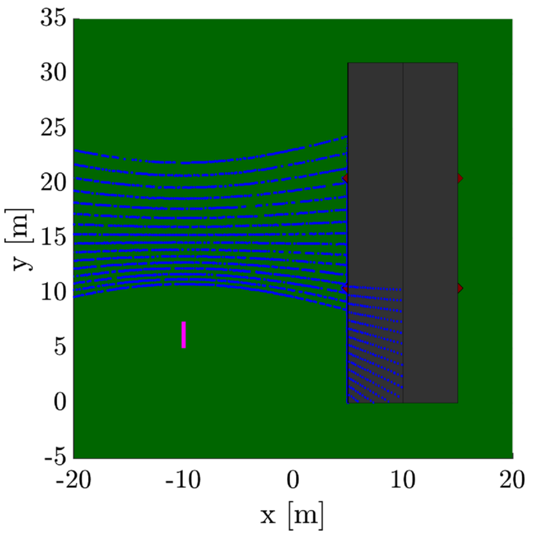

Fig. 6 Top view of the simulation environment with a simulated point cloud (blue dots) of the first epoch and trajectory of the UAS (magenta line)

the symbolic Math toolbox of Matlab©. Subsequently, we implemented the partial derivatives in a function. Another possibility to obtain the partial derivatives could be the use of INTLAB (Rump 1999). We fixed the stop criterion c_{stop} to 1×10^{-12} in the measurement update step. This stop criterion was reached after an average of six iterations in the subsequent simulation.

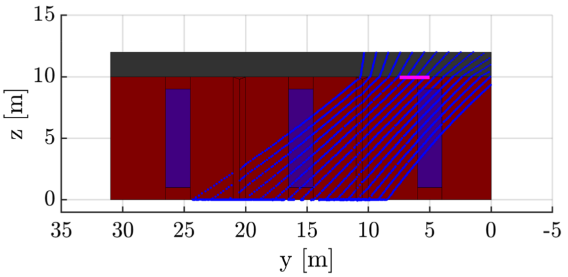

Fig. 7 Side view of the simulation environment with a simulated point cloud (blue dots) of the first epoch and trajectory of the UAS (magenta line)

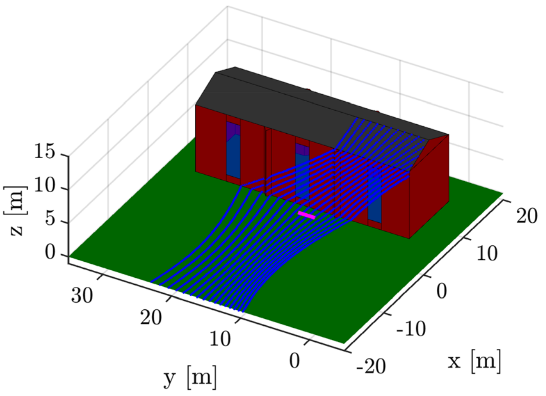

Fig. 8 Oblique view of the simulation environment with a simulated point cloud (blue dots) of the first epoch and trajectory of the UAS (magenta line)

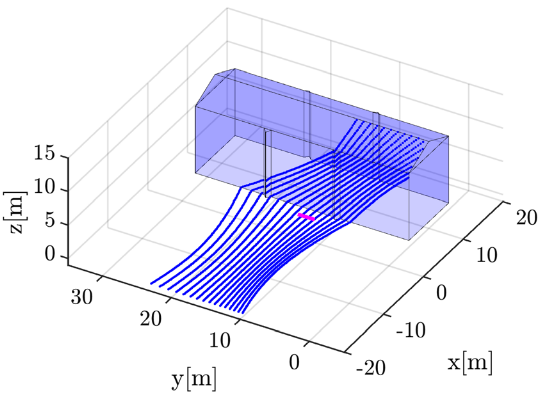

Fig. 9 3D city model with a simulated point cloud (blue dots) of the first epoch and trajectory of the UAS (magenta line)

Algorithm 1: Iterated extended Kalman filter with nonlinear implicit measurement equation and nonlinear equality state constraints

```
System model (see Eqs. 20-28):
$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{w}_{k-1}, \quad \boldsymbol{w}_{k-1} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_{ww})$
Observation model (see Eqs. 29-35):
$\boldsymbol{h}(E(\boldsymbol{l}_k), \boldsymbol{x}_k) = \boldsymbol{h}(\boldsymbol{l}_k + \boldsymbol{v}_k, \boldsymbol{x}_k) = \mathbf{0}, \quad \boldsymbol{v}_k \sim N(\mathbf{0}, \boldsymbol{\Sigma}_{vv})$
Initial parameter vector and its VCM (see Eqs. 44-52):
$\hat{\boldsymbol{x}}_0^+ = \boldsymbol{x}_0, \quad \boldsymbol{\Sigma}_{\hat{x}\hat{x},0}^+ = \boldsymbol{\Sigma}_{xx,0}, \quad k = 1$
while $k <$ K do
    Prediction step
    $\boldsymbol{F}_{x,k} = \partial \boldsymbol{f} / \partial \boldsymbol{x}|_{\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}_{k-1}}$   (see Eqs. 22-24)
    $\hat{\boldsymbol{x}}_k^- = \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}_{k-1}\right)$  (see Eqs. 20-21)
    $\boldsymbol{\Sigma}_{xx,k}^- = \boldsymbol{F}_{x,k} \boldsymbol{\Sigma}_{xx,k-1}^+ \boldsymbol{F}_{x,k}^T + \boldsymbol{\Sigma}_{ww}$
    Assignment (see Section 2.3 and Eq. 1)
    $\boldsymbol{l}_{Scan,k}^{\mathrm{glo}} = \hat{\boldsymbol{t}}_k^- + \hat{\boldsymbol{R}}_k^- \cdot \boldsymbol{l}_{Scan,k}^{\mathrm{loc}}$
    Assign $\boldsymbol{l}_{Scan,k}^{\mathrm{glo}}$ to the planes of the city model
    Arrange observation vector (see Eqs. 11-19)
    Store all points, distinctively assigned to a plane,
      in $\boldsymbol{l}_{Scan,k}^{\mathrm{loc}} = \left[\boldsymbol{l}_{Scan,1,k}^{\mathrm{loc}}, \ldots, \boldsymbol{l}_{Scan,E,k}^{\mathrm{loc}}\right]$
    Arrange the observation vector
      $\boldsymbol{l}_k = \left[\boldsymbol{l}_{Scan,k}^{\mathrm{loc}}, \boldsymbol{l}_{V,0}, \boldsymbol{l}_{Pose,k}\right]^T$ and its VCM
      $\boldsymbol{\Sigma}_{ll,k}$
    Measurement-update step
    $\check{\boldsymbol{l}}_{k,0} = \boldsymbol{l}_k, \quad \check{\boldsymbol{x}}_{k,0} = \hat{\boldsymbol{x}}_k^-$
    $\boldsymbol{\Delta x} = \boldsymbol{\infty}, \quad \boldsymbol{\Delta l} = \boldsymbol{\infty}, \quad m = 0$
    while $\max(\boldsymbol{\Delta x}) > c_{\mathrm{stop}} \vee \max(\boldsymbol{\Delta l}) > c_{\mathrm{stop}}$ do
        $\boldsymbol{H}_{x,k,m} = \partial \boldsymbol{h} / \partial \boldsymbol{x}|_{\check{\boldsymbol{l}}_{k,m}, \check{\boldsymbol{x}}_{k,m}}$
        $\boldsymbol{H}_{l,k,m} = \partial \boldsymbol{h} / \partial \boldsymbol{l}|_{\check{\boldsymbol{l}}_{k,m}, \check{\boldsymbol{x}}_{k,m}}$
        $\boldsymbol{O}_{k,m} = \boldsymbol{H}_{x,k,m} \boldsymbol{\Sigma}_{xx,k}^- \boldsymbol{H}_{x,k,m}^T$
        $\boldsymbol{S}_{k,m} = \boldsymbol{H}_{l,k,m} \boldsymbol{\Sigma}_{vv} \boldsymbol{H}_{l,k,m}^T$
        $\boldsymbol{K}_{k,m} = \boldsymbol{\Sigma}_{xx,k}^- \boldsymbol{H}_{x,k,m}^T \cdot (\boldsymbol{O}_{k,m} + \boldsymbol{S}_{k,m})^{-1}$
        $\boldsymbol{r}_{k,m} =$
          $\boldsymbol{H}_{l,k,m} \cdot (\boldsymbol{l}_k - \check{\boldsymbol{l}}_{k,m}) + \boldsymbol{H}_{x,k,m} \cdot \left(\hat{\boldsymbol{x}}_k^- - \check{\boldsymbol{x}}_{k,m}\right)$
        $\check{\boldsymbol{x}}_{k,m+1} = \hat{\boldsymbol{x}}_k^- - \boldsymbol{K}_{k,m} \cdot \left(\boldsymbol{h}\left(\check{\boldsymbol{l}}_{k,m}, \check{\boldsymbol{x}}_{k,m}\right) + \boldsymbol{r}_{k,m}\right)$
        $\boldsymbol{G}_{k,m} = \boldsymbol{\Sigma}_{vv} \boldsymbol{H}_{l,k,m}^T (\boldsymbol{O}_{k,m} + \boldsymbol{S}_{k,m})^{-1}$
        $\check{\boldsymbol{l}}_{k,m+1} = \boldsymbol{l}_k - \boldsymbol{G}_{k,m} \cdot \left(\boldsymbol{h}\left(\check{\boldsymbol{l}}_{k,m}, \check{\boldsymbol{x}}_{k,m}\right) + \boldsymbol{r}_{k,m}\right)$
        $\boldsymbol{\Delta x} = |\check{\boldsymbol{x}}_{k,m+1} - \check{\boldsymbol{x}}_{k,m}|, \quad \boldsymbol{\Delta l} = |\check{\boldsymbol{l}}_{k,m+1} - \check{\boldsymbol{l}}_{k,m}|$
        $m = m + 1$
    $\hat{\boldsymbol{x}}_k^+ = \check{\boldsymbol{x}}_{k,m}$
    $\hat{\boldsymbol{l}}_k^+ = \check{\boldsymbol{l}}_{k,m}$
    $\boldsymbol{L}_k = \underset{j \times j}{\mathbf{I}} - \boldsymbol{K}_{k,m-1} \boldsymbol{H}_{x,k,m-1}$
    $\boldsymbol{\Sigma}_{\hat{x}\hat{x},k}^+ = \boldsymbol{L}_k \boldsymbol{\Sigma}_{xx,k}^- \boldsymbol{L}_k^T + \boldsymbol{K}_{k,m-1} \boldsymbol{S}_{k,m-1} \boldsymbol{K}_{k,m-1}^T$
    $\boldsymbol{U}_k = \boldsymbol{G}_{k,m-1} \cdot \boldsymbol{H}_{x,k,m-1}$
    $\boldsymbol{\Sigma}_{\hat{v}\hat{v},k}^+ =$
      $\boldsymbol{\Sigma}_{vv} + \boldsymbol{G}_{k,m-1} \boldsymbol{S}_{k,m-1} \boldsymbol{G}_{k,m-1}^T - \boldsymbol{U}_k \boldsymbol{\Sigma}_{xx,k}^- \boldsymbol{U}_k^T$
    Constraint step (see Eqs. 36-43)
    $\boldsymbol{D} = \partial \boldsymbol{g} / \partial \boldsymbol{x}|_{\hat{\boldsymbol{x}}_k^-}$
    $\boldsymbol{d} = \boldsymbol{b} - \boldsymbol{g}\left(\hat{\boldsymbol{x}}_k^-\right) + \boldsymbol{D} \cdot \hat{\boldsymbol{x}}_k^-$
    Set  $\boldsymbol{W} = \underset{j \times j}{\mathbf{I}}$
    $\tilde{\boldsymbol{x}}_k^+ =$
      $\hat{\boldsymbol{x}}_k^+ - \boldsymbol{W}^{-1} \boldsymbol{D}^T \left(\boldsymbol{D} \boldsymbol{W}^{-1} \boldsymbol{D}^T\right)^{-1} \left(\boldsymbol{D} \hat{\boldsymbol{x}}_k^+ - \boldsymbol{d}\right)$
    $\boldsymbol{\Sigma}_{\tilde{x}\tilde{x},k}^+ =$
      $\boldsymbol{\Sigma}_{\hat{x}\hat{x},k}^+ - \boldsymbol{\Sigma}_{\hat{x}\hat{x},k}^+ \boldsymbol{D}^T \left(\boldsymbol{D} \boldsymbol{\Sigma}_{\hat{x}\hat{x},k}^+ \boldsymbol{D}^T\right)^{-1} \boldsymbol{D} \boldsymbol{\Sigma}_{\hat{x}\hat{x},k}^+$
    Set $\hat{\boldsymbol{x}}_k^+ = \tilde{\boldsymbol{x}}_k^+$ and $\boldsymbol{\Sigma}_{\hat{x}\hat{x},k}^+ = \boldsymbol{\Sigma}_{\tilde{x}\tilde{x},k}^+$
```

Table 1 Assumed systematic effects Δ and standard deviations σ of the added noise for all sensors and observations (Obs.)

Sensor	Obs.	Scenario 1		Scenario 2	
		σ	Δ	σ	Δ
Laser scanner	Wall, roof	0.02 m	0 m	0.02 m	0 m
	Ground	0.02 m	0 m	0.2 m	0 m
	Window	0.02 m	0 m	0.1 m	0.6 m
GNSS	t_x, t_y, t_z	0.5 m	0 m	0.5 m	0 m
IMU	ω, ϕ	0.2°	0°	0.2°	0°
	κ	0.2°	0°	0.2°	$0.01° \cdot k$

3 Simulation

Computer simulations are a great tool for analyzing and interpreting engineering systems. Here, we intended to evaluate the functionality and performance of the IEKF implemented. Therefore, we focused on two scenarios which basically conform to the testing environment designated for the *UAS Project*. Furthermore, we focused on the challenging initialization phase of the IEKF and analyzed a rather short trajectory of 2.5 m length and with 50 epochs.

Therefore, we created a model of a fictitious building and a ground plane. Subsequently, we determined a fictitious trajectory beside this building. We assumed a constant velocity for the UAS of 1 m s^{-1} and simulated laser scanner measurements with a frequency of 20 Hz. Consequently, the epochs simulated are at constant distances of 0.05 m. Each of the epochs consists of a laser scan and the desired values of position and orientation obtained from the determined trajectory at a certain time.

Figures 6, 7 and 8 depict different views of the building and ground plane created. The building created contains roof planes (in dark grey), wall planes (in red), and window planes (in transparent blue). Figure 9 depicts the corresponding 3D city model of the building. As we can see, the ground plane is not part of the 3D city model. The 3D city model contains the vertices of each plane and is given in the superordinate coordinate system. The simulated and correctly georeferenced point cloud (blue dots) of the first epoch and the trajectory of the UAS (magenta line) is depicted in each figure.

3.1 Scenarios

We simulated two scenarios in which the laser scanner and IMU measurements were generated differently under certain assumptions regarding measurement accuracy and bias. We repeated the simulation 500 times ($S = 500$) for both scenarios. In scenario 1, we only added normally distributed noise on the laser scanner measurements, the position (representing the GNSS receiver), and the orientation (representing the IMU). In scenario 2, we systematically perturbed the laser scanner measurements, hitting the windows of the simulated building, and increased the (standard deviation of the) noise for laser scanner measurements, hitting the windows and the ground. The systematic and increased random disturbances of the measurements hitting the windows simulate the infiltration behavior for glass. We included the increased measurement uncertainty and the actual infiltration of the laser into the increased noise of these measurements. The increased noise for measurements hitting the ground simulates possible unevenness and vegetation. Thus, this increased noise is more justified by actual structures in the object space than by an increased uncertainty of the measurement. Furthermore, we added a drift on the κ-component of the IMU. Therefore, we linked the systematic effect Δ to the epoch number $k \in \{1, \dots, K\}$. The κ-component conforms to the heading of the UAS. All assumed systematic effects Δ and standard deviations σ of the added noise are depicted in Table 1. Basically, the assumptions of scenario 1 should be consistent with the manufacturer's specifications of the sensors used or planned for our UAS. The assumptions in scenario 2, especially the ones for the laser scanner, are based on experience from test measurements. In our simulation, we assume that the positions and orientations of all sensors in a platform coordinate system or body frame have already been determined in a calibration process. We also assume that the sensors are properly synchronized. To neglect the effect of generalization in the 3D city model, we used the same model for simulation and in the following IEKF algorithm.

Table 2 Assumed standard deviations σ for the initial VCM of the state parameters $\mathbf{\Sigma}_{xx,0}$, the system noise $\mathbf{\Sigma}_{ww}$, and the observation vector $\mathbf{\Sigma}_{ll}$

Initial state noise	$\sigma_{t,0} = 0.5$ m
	$\sigma_{o,0} = 0.2°$
	$\sigma_{v,0} = 1$ m s^{-1}
	$\sigma_{n,0} = 0.0001$
	$\sigma_{d,0} = 0.001$ m
	$\sigma_{V,0} = 0.0001$ m
System noise	$\sigma_{t,w} = 3 \cdot \Delta\tau$ m
	$\sigma_{o,w} = 3 \cdot \Delta\tau°$
	$\sigma_{v,w} = 5 \cdot \Delta\tau$ m s^{-1}
	$\sigma_{n,w} = 0$
	$\sigma_{d,w} = 0$ m
	$\sigma_{V,w} = 0$ m
Measurement noise	$\sigma_{LS} = 0.02$ m
	$\sigma_{t} = 0.5$ m
	$\sigma_{o} = 0.2°$
	$\sigma_{V} = 0.0001$ m

3.2 Values Chosen for the IEKF

We used a constant distance threshold $d_{\text{assign}} = 0.3$ m for the assignment algorithm (see Sect. 2.3) in our simulation. This chosen threshold is a trade-off between the rather imprecise pose in the first epochs and the more precise pose in the last epochs.

The standard deviations used for building the initial VCM $\boldsymbol{\Sigma}_{xx,0}$ (see Eqs. 49–52) are depicted at the top of Table 2. The standard deviations of the initial position $\sigma_{t,0}$ and orientation $\sigma_{o,0}$ conform to the standard deviations of the GNSS and IMU observations. We assume a standard deviation for the initial velocity $\sigma_{v,0}$ which includes the planned maximal velocities of the UAV. For the standard deviations of the initial plane parameters $\sigma_{n,0}$ and $\sigma_{d,0}$, we met the assumption that these values should be very small. Thus, they correspond roughly to the standard deviation of the initial vertices $\sigma_{V,0}$, which, in turn, conform to the standard deviation chosen for the observation vector. If the accuracies of the planes' vertices are known, the standard deviations of the plane parameters could also be determined by means of variance propagation.

The standard deviations chosen for the system noise (see Eqs. 25–28) can be found in the middle of Table 2. We set the standard deviation for translation $\sigma_{t,w}$, orientation $\sigma_{o,w}$ and velocity $\sigma_{v,w}$ depending on the time period $\Delta\tau$ between two epochs and regarding possible unpredictable movements of the UAS in this time period. We set the system noise to zero for $\boldsymbol{\Sigma}_{ww,\text{Plane}}$ and $\boldsymbol{\Sigma}_{ww,V}$, because these parameters are constant over time. Consequently, $\sigma_{n,w}$, $\sigma_{d,w}$, and $\sigma_{V,w}$ are zero.

We used the simulated standard deviation $\sigma_{\text{LS}} = 0.02$ m of the laser scanner measurements for the stochastic model of the observation vector (see Eq. 19). We also introduced the simulated standard deviations $\sigma_t = 0.5$ m and $\sigma_o = 0.2°$ for the GNSS and IMU observations. In the case of the initial vertices of the 3D building model, we introduced a small standard deviation of $\sigma_V = 0.1$ mm. The 3D city model is, thus, almost introduced as a fixed feature. We are aware that this assumption is too optimistic and that the accuracy of

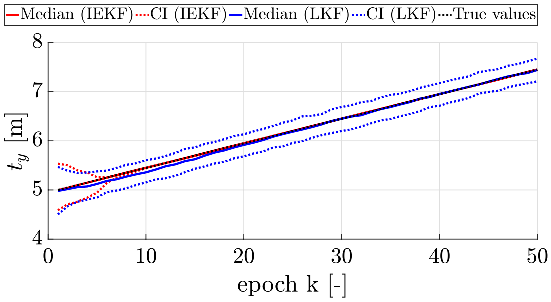

Fig. 11 Median and confidence intervals of t_y in scenario 1

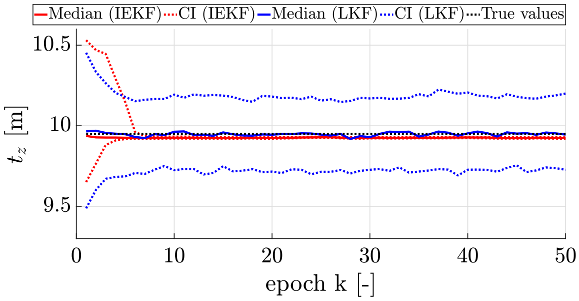

Fig. 12 Median and confidence intervals of t_z in scenario 1

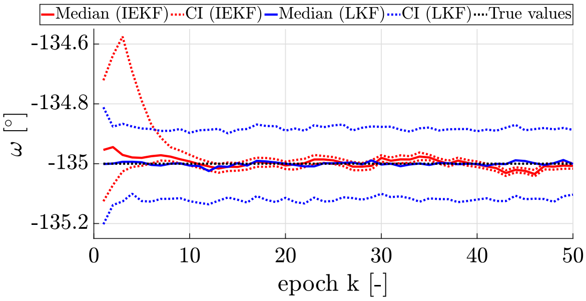

Fig. 13 Median and confidence intervals of ω in scenario 1

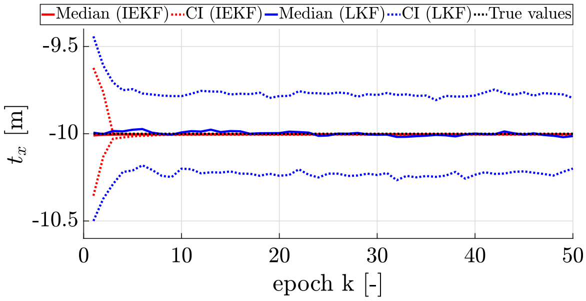

Fig. 10 Median and confidence intervals of t_x in scenario 1

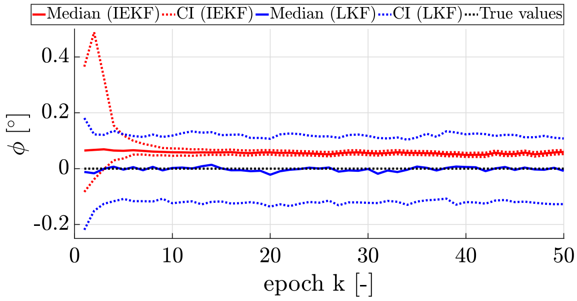

Fig. 14 Median and confidence intervals of ϕ in scenario 1

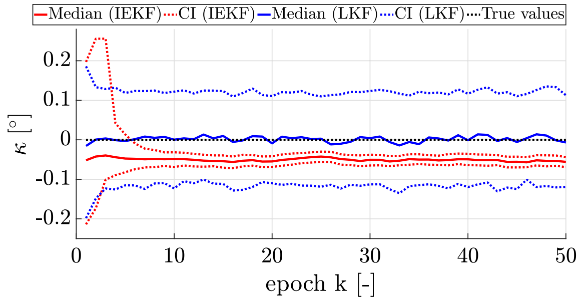

Fig. 15 Median and confidence intervals of κ in scenario 1

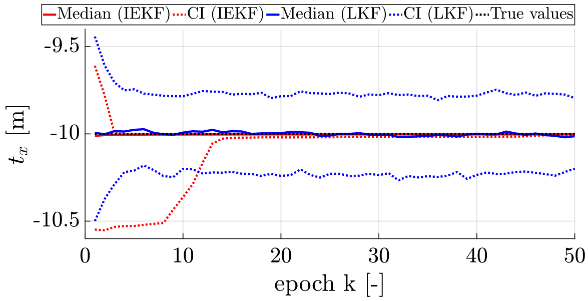

Fig. 16 Median and confidence intervals of t_x in scenario 2

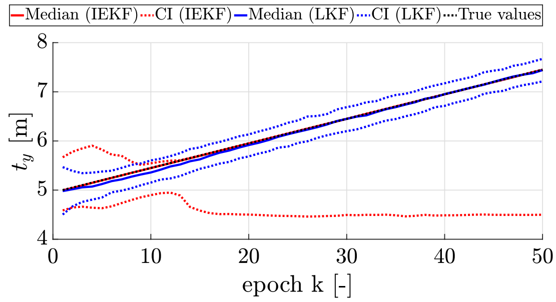

Fig. 17 Median and confidence intervals of t_y in scenario 2

the introduced 3D city model is rather in the range of a few centimeters. However, in the context of this paper, the influence of a less precise 3D city model will not be considered in more detail.

We assume that $\mathbf{\Sigma}_{ll,k}$ and $\mathbf{\Sigma}_{ww}$ do not contain correlations on their own and among themselves.

4 Results and Discussion

In the following, we depict the results of the IEKF for both scenarios and compare them with the results of a linear Kalman filter (LKF), which only uses the simulated GNSS and IMU observations. The comparison between LKF and IEKF evaluates the benefit of introducing laser scanner observations and a city model. The state parameter vector $\boldsymbol{x}_k$ of the LKF solely consists of $\boldsymbol{x}_{\mathrm{State},k}$ (Eq. 6). The observation vector $\boldsymbol{l}_k$ consists of $\boldsymbol{l}^{\mathrm{glo}}_{\mathrm{Pose},k}$ (Eq. 15). The transition matrix $\boldsymbol{F}_{x,k}$ of the LKF is given by $\boldsymbol{F}_{x,\mathrm{State},k}$ (Eq. 24). Consequently, the VCM of the observation vector and of the system noise have to be adapted. The LKFs' only remaining measurement equation (Eq. 31) has to be transformed in explicit form. For the LKF, no linearization is necessary and no constraints are used. For more information concerning the LKF, see (Simon 2010). Figures 10, 11, 12, 13, 14 and 15 depict the median values (solid lines) and 68%(1σ) confidence intervals (CI) (dashed lines) for the translation parameters t_x, t_y, t_z (Figs. 10, 11, 12) and orientation parameters ω, ϕ, κ (Figs. 13, 14, 15) in scenario 1. The CI are calculated numerically using the total number of simulated runs S according to Alkhatib et al. (2009). The red lines result from the filtered state parameters $\hat{\boldsymbol{x}}^+_k$ of the IEKF after the constraint step. The blue lines result from the filtered state parameters $\hat{\boldsymbol{x}}^+_k$ of the LKF, respectively. The true values are plotted in a dashed black line.

The median of the IEKF result for t_x (across flight direction) in Fig. 10 is very close to the true translation, whereas we can see slight random deviations between the median of the LKF results and the true values. The latter can be seen for each pose component and is not explicitly mentioned again in the following. The CI of the IEKF and the LKF shrink rapidly in the first 5 epochs. Whereas the CI of the IEKF is very close to the median, the CI of the LKF is significantly broader. The latter can be seen for each pose component and is also not explicitly mentioned again. The IEKF performs well for that pose component because of the measurement constellation. Each measurement distinctively assigned to a plane provides information for that pose component. In other words, each measured plane of the 3D city model is sensitive for that pose component.

The results are similar for t_y (in flight direction) in Fig. 11. Again, the median of the IEKF results is very close to the true translation and its CI shrinks within the first epochs. For this pose component, the shrinking of the CI (IEKF) is slightly slower than for t_x and lasts until epoch 10. A reason for that slight difference can be found in the weaker measurement constellation for that pose component. Only the triangle-shaped protrusions are sensitive for t_y. Consequently, there are significantly fewer observations which provide information for that pose component.

Figure 12 shows the results for t_z (up-direction). The median of the IEKF results is very constant, but seems to deviate from the true values by a small systematic shift of about 3 cm. This systematic shift is caused by ground points wrongly assigned to a wall plane of the building. This results in a slight rotation around the y-axis (see results for ϕ) in combination with a slight descent of t_z. Again, the CI of the

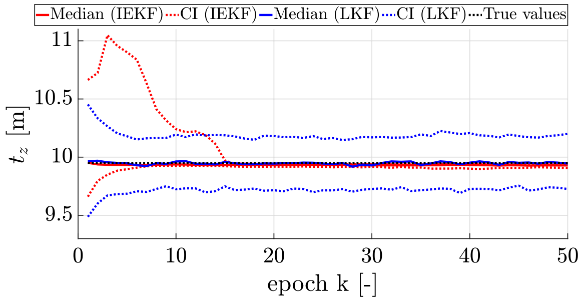

Fig. 18 Median and confidence intervals of t_z in scenario 2

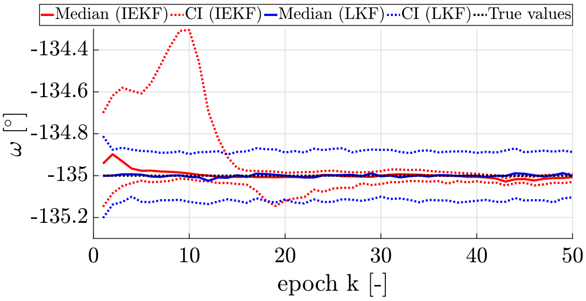

Fig. 19 Median and confidence intervals of ω in scenario 2

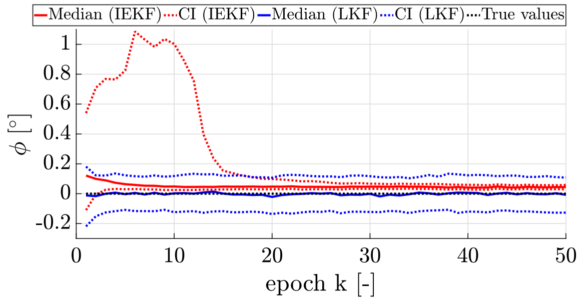

Fig. 20 Median and confidence intervals of ϕ in scenario 2

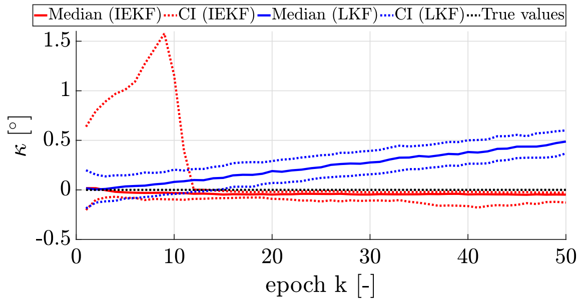

Fig. 21 Median and confidence intervals of κ in scenario 2

IEKF translation shrinks rapidly within the first 6 epochs. The boundaries of the CI in the first epochs are not symmetrical around the median. They are slightly shifted to larger t_z values. That means that the IEKF converges slower towards the true translation with initial t_z values larger than the true translation. An initial t_z value larger than the true translation may cause significantly fewer assignments in the first epochs. Only the roof planes are sensitive for that pose component.

The results for ω, which represents the rotation around the x-axis, are shown in Fig. 13. After 10 epochs, the median of IEKF and LKF obtain comparable results, whereas the median of the IEKF seems to vary slightly more. The CI of the IEKF shrinks within the first 12 epochs. Similar to the t_z pose component, there is a shift of the IEKF CI towards larger ω values. The only sensitive planes for that pose component are the triangle-shaped protrusions and the roof planes. This may explain the larger variations in the median of the IEKF orientation.

Figures 14, 15 show the results for ϕ (rotation around y-axis) and κ (rotation around z-axis), respectively. We can see similar characteristics for these pose components. In both cases, the median of the IEKF results is constant but systematically shifted by a small value of about 0.05°. As has already been mentioned for t_z, the systematic shift in ϕ is caused by wrongly assigned ground points. Because of the skewed alignment of the laser scanner, the wrongly assigned ground points also cause a slight rotation in κ. All planes are sensitive for both pose components.

For scenario 2, we depicted the results in a similar way in Figs. 16, 17, 18, 19, 20 and 21. In Fig. 16, we can see the results for t_x. Similar to scenario 1, the median of the IEKF results comes very close to the true translation. The lower boundary of the CI (IEKF) is farther afield from the median than the upper boundary. It shrinks until epoch 14. An explanation for the one-sided enlarged CI (IEKF) is the systematically extended laser scanner measurements hitting the windows. With an initial t_x translation, which is farther afield from the building than the true translation, these systematically extended laser scanner measurements are wrongly assigned to the walls of the building. After some epochs, these false assignments seem to have disappeared. These effects are analyzed later on (see Fig. 29).

The median of the IEKF results for t_y in Fig. 17 is close to the true translation. The upper boundary of the CI (IEKF) shrinks until epoch 13, whereas the lower boundary of the CI (IEKF) clearly departs from the median from epoch 14 on. Again, these effects are analyzed later on (see Figs. 28 and 29).

We can see the results for t_z in scenario 2 in Fig. 18. The results are comparable to the ones obtained in scenario 1 for that pose component. A significant difference can be found in the upper boundary of the CI (IEKF), which

Table 3 RMSE of position and orientation in scenario 1 (first row) and 2 (second row)

Criterion	RMSE$_{t_x}$, R [m]		RMSE$_{t_y}$, R [m]		RMSE$_{t_z,R}$ [m]		RMSE$_{\omega,R}$ [°]		RMSE$_{\phi,R}$ [°]		RMSE$_{\kappa,R}$ [°]	
	IEKF	LKF	IEKF	LKF	IEKF	LKF	IEKF	LKF	IEKF	LKF	IEKF	LKF
Minimum	0.0033	0.1095	0.0018	0.1121	0.0225	0.1106	0.0112	0.0654	0.0489	0.0629	0.0456	0.0622
	0.0024		0.0023		0.0166		0.0122		0.0384		0.0333	0.1867
Maximum	2.1173	0.2957	3.4644	0.3348	3.0158	0.3085	1.8773	0.1407	2.5912	0.1426	2.1497	0.1466
	2.1955		2.9587		3.2166		2.6897		2.9243		2.3307	0.3233
Mean	0.0654	0.1899	0.1955	0.1941	0.1166	0.1911	0.0694	0.0964	0.1168	0.0973	0.1284	0.0968
	0.1080		0.4496		0.1829		0.1648		0.2008		0.2288	0.2535
Median	0.0079	0.1864	0.0177	0.1898	0.0343	0.1897	0.0245	0.0966	0.0621	0.0980	0.0556	0.0968
	0.0128		0.0424		0.0417		0.0519		0.0727		0.0585	0.2510
SD	0.2517	0.0337	0.5697	0.0362	0.3254	0.0344	0.1578	0.0130	0.2348	0.0123	0.2536	0.0135
	0.3069		0.7712		0.4385		0.2972		0.2933		0.2924	0.0245
↓ CI (68%)	0.0037	0.1561	0.0041	0.1598	0.0246	0.1566	0.0162	0.0834	0.0556	0.0849	0.0501	0.0838
	0.0030		0.0048		0.0194		0.0179		0.0434		0.0404	0.2311
↑ CI (68%)	0.0355	0.2271	0.1010	0.2283	0.0901	0.2254	0.0753	0.1090	0.1050	0.1091	0.1150	0.1091
	0.1498		1.7169		0.2881		0.2400		0.2707		0.4953	0.2783
↓ CI (95%)	0.0034	0.1306	0.0023	0.1320	0.0234	0.1267	0.0136	0.0735	0.0531	0.0723	0.0475	0.0725
	0.0026		0.0028		0.0178		0.0145		0.0402		0.0361	0.2053
↑ CI (95%)	0.7583	0.2614	2.4217	0.2725	0.8801	0.2634	0.5240	0.1229	0.7461	0.1187	1.0262	0.1275
	1.5383		2.3080		1.4151		1.0929		1.0209		0.9829	0.3064
Best RMSE	95.0%	5.0%	89.8%	10.2%	90.2%	9.8%	87.6%	12.4%	80.2%	19.8%	80.6%	19.4%
	90.2%	9.8%	68.2%	31.8%	81.2%	18.8%	61.2%	38.8%	57.4%	42.6%	65.4%	34.6%

converges to the median much slower than in scenario 1. In scenario 2, it does not converge until epoch 15. Again, initial t_z values larger than the true position cause significantly fewer assignments in the first epochs.

We observe comparable results to scenario 1 for the pose parameters ω and ϕ in Figs. 19 and 20. In both figures, we can observe that one boundary of the CI (IEKF) is farther afield from the median than the other and that the CI (IEKF) shrinks slower than in scenario 1.

The pose component κ is depicted in Fig. 21. The CI (IEKF) shrinks until epoch 12. In addition, we can see that the simulated drift in κ does not affect the IEKF results, whereas the median of the LKF departs from the true orientation continuously as simulated.

A suitable value to evaluate the performances of the filters is the root-mean-square error (RMSE). In our simulation, the RMSE is the error between filtered state parameters and the true state parameters. The RMSE of the pose parameter t_x for run $s \in \{1, \ldots, S\}$ is calculated according to:

$$\mathrm{RMSE}_{t_x,R.,s} = \frac{1}{K} \sum_{k=1}^{K} \sqrt{\left(t_{x,s,k} - \bar{t}_{x,k}\right)^2}. \qquad (53)$$

$t_{x,s,k}$ results from the filtered state parameter vector $\hat{\mathbf{x}}_k^+$ from the IEKF or the LKF, respectively. $\bar{t}_{x,k}$ are the true values.

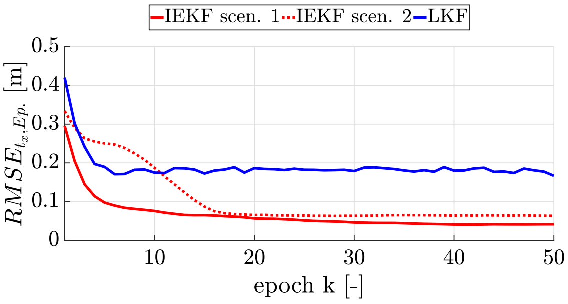

Fig. 22 RMSE$_{t_x,\mathrm{Ep.}}$

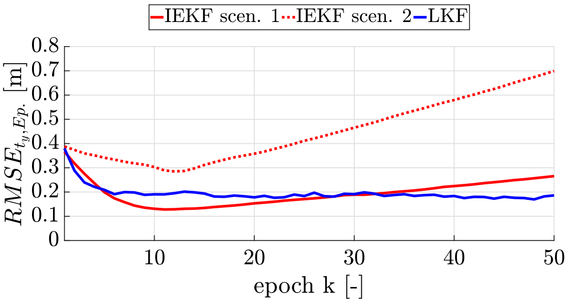

Fig. 23 RMSE$_{t_y,\mathrm{Ep.}}$

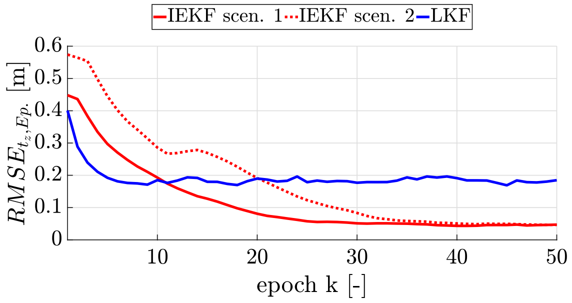

Fig. 24 $\mathrm{RMSE}_{t_z,\mathrm{Ep.}}$

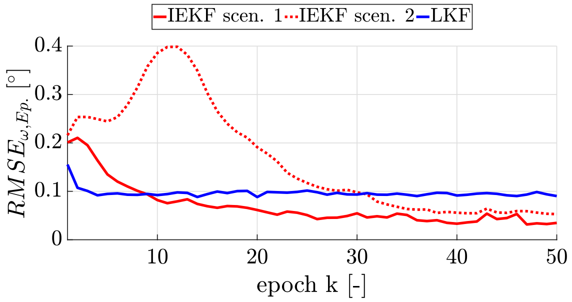

Fig. 25 $\mathrm{RMSE}_{\omega,\mathrm{Ep.}}$

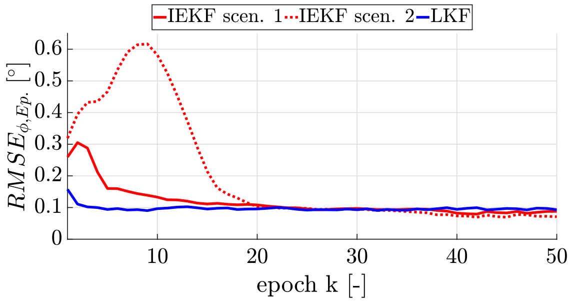

Fig. 26 $\mathrm{RMSE}_{\phi,\mathrm{Ep.}}$

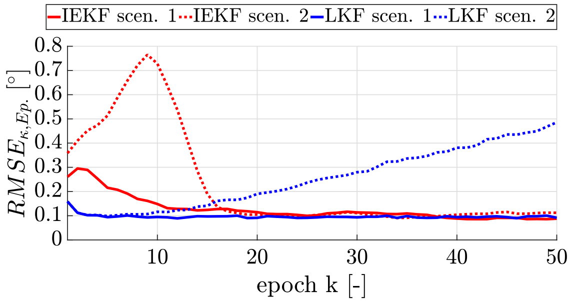

Fig. 27 $\mathrm{RMSE}_{\kappa,\mathrm{Ep.}}$

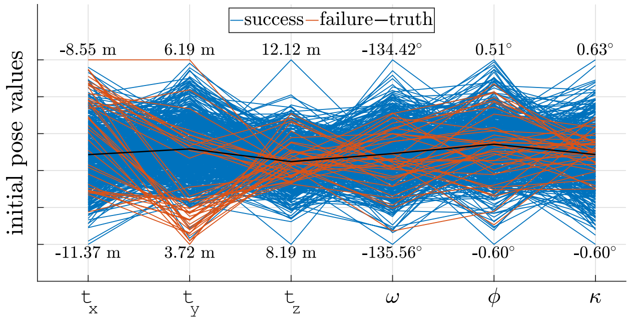

Fig. 28 Initial pose values plotted as parallel coordinates in scenario 1. Total failure rate: 7.6%

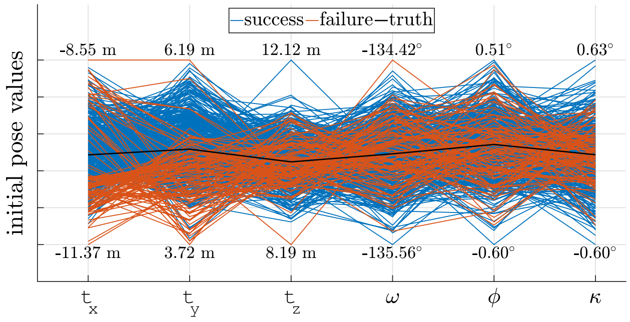

Fig. 29 Initial values plotted as parallel coordinates in scenario 2. Total failure rate: 20.4%

The calculation of the RMSE for the other pose parameters is carried out analogously.

Table 3 depicts the characteristic criterions: Minimum, maximum, mean, median, standard deviation (SD), and the lower bound (↓) and upper bound (↑) of the 68% and 95% CI of different RMSE. In the last row of Table 3, we display the rate of runs where the IEKF obtained a smaller RMSE than LKF and vice versa. Each criterion is divided into two rows regarding scenario 1 (above) and scenario 2 (below).

Table 3 clearly shows that the IEKF obtains better results than the LKF. The median values especially are significantly smaller in each state parameter and each scenario. By contrast, the range of the IEKF results is significantly larger than the range of the LKF results, which can be seen in the SD and 95% CI values. A possible explanation can be found in a larger number of runs, where the IEKF is far away from the true values. We will subsequently call these runs *failures*. These *failures* can be caused by a disadvantageous initial pose parameter in the first epoch. By this, a large number of laser scanner measurements are assigned to wrong planes and the IEKF *converges* to a wrong pose. The interaction between initial pose parameters and performance of the IEKF is analyzed at the end of this section. The *failures* are also the explanation for the large discrepancy between the mean and median values.

To evaluate the performance of the IEKF over the epochs, we calculated a mean RMSE for each epoch over all S simulation runs. The calculation is as follows for the RMSE of t_x in epoch $k \in \{1, \ldots, K\}$:

$$\mathrm{RMSE}_{t_x,\mathrm{Ep.},k} = \frac{1}{S} \sum_{s=1}^{S} \sqrt{\left(t_{x,s,k} - \bar{t}_{x,k}\right)^2}. \tag{54}$$

We obtain the RMSE for the other pose parameters analogously. The results can be found in Figs. 22, 23, 24, 25, 26 and 27.

Figures 22, 23, 24, 25, 26 and 27 generally support the interpretations made before. The RMSE of the translation parameters t_x and t_z determined by the IEKF decrease with an increasing number of epochs. At the last epoch, the $\mathrm{RMSE}_{t_x,\mathrm{Ep.}}$ and $\mathrm{RMSE}_{t_z,\mathrm{Ep.}}$ of the IEKF is significantly smaller than the RMSE obtained by LKF in both scenarios. Again, $\mathrm{RMSE}_{t_y,\mathrm{Ep.}}$ differs and increases after epoch 10 with an increasing number of epochs. In both scenarios, especially in scenario 2, the $\mathrm{RMSE}_{t_y,\mathrm{Ep.}}$ obtained by the IEKF is larger than the one obtained by LKF in the last epochs. The RMSE of the orientation parameter show different characteristics, but they are intrinsically quite similar. The RMSE obtained by IEKF starts with large values and significantly decreases after epoch 10. In the last epoch, $\mathrm{RMSE}_{\phi,\mathrm{Ep.}}$ and $\mathrm{RMSE}_{\kappa,\mathrm{Ep.}}$ of IEKF and LKF are quite similar. $\mathrm{RMSE}_{\omega,\mathrm{Ep.}}$ of the IEKF undercuts the $\mathrm{RMSE}_{\omega,\mathrm{Ep.}}$ of the LKF until epoch 9 (scenario 1), resp. epoch 30 (scenario 2). The $\mathrm{RMSE}_{\kappa,\mathrm{Ep.}}$ of LKF in scenario 2 is clearly affected by the simulated IMU drift. It is remarkable that (except in $\mathrm{RMSE}_{t_y,\mathrm{Ep.}}$) the RMSE of the IEKF between scenario 1 and scenario 2 is very similar in the last epoch.

For a deeper understanding of the link between initialization and performance of the IEKF, we plotted the initial values of GNSS and IMU of the first epoch in parallel coordinates. The parallel coordinate representation was first used by Inselberg (1985). For this representation, we will show all pose components in six parallel axes. On each axis, all initial values belonging to this component are depicted. This representation is used to identify whether some extreme initial pose values could have had a large impact on the divergence of the IEKF. In addition, we colored the runs where the IEKF failed (*failure*) in orange and the others (*success*) in blue. The distinction in *failure* and *success* was selected based on the RMSE of the position in the last epoch. The run is classified as a *failure* when there is a value larger than 10 cm.

In scenario 1 (Fig. 28), the failure rate is 7.6%. For the initial pose parameters t_z, ω, ϕ, and κ, the *failures* are distributed over the whole value range. For t_x, the *failures* are slightly concentrated in a range between − 9.2 and − 8.55 m and a second range around − 10.6 m. For t_y, the *failures* are even more concentrated in a range between 3.72 and 4.3 m. Initial values for t_y in a range between 3.72 and 4.3 m lead to most of the *failures*. As previously mentioned, we have a weak measurement constellation for t_y. Furthermore, the trajectory is parallel to the y-axis with a true value of $t_y = 7.5$ m in epoch 50, the consequence being that with initial values for t_y between the true value (5 m) and 6.19 m, the IEKF may get back to the true values in a later epoch. This possibility is not given for initial values for t_y between 3.72 and 4.3 m.

In scenario 2, the failure rate of 20.4% is significantly higher. Again, for the initial pose parameters t_z, ω, ϕ, and κ, the *failures* are distributed over the whole value range. For t_y, the *failures* are mainly in a range between 3.72 and 5 m. The reason for that has already been explained for scenario 1. For t_x, we can see a strong concentration of *failures* in a range between − 11.37 and − 10.4 m. Thus, especially that range for t_x causes the *failures* in scenario 2. The reason for that can be found in the systematically perturbed laser scanner measurements hitting the windows. These measurements are extended by an additive value $\Delta = 0.6\,\mathrm{m}$. In combination with a disadvantageous initialization, these perturbed measurements are wrongly assigned to the wall plane and, therefore, cause the *failures*.

We used GNSS and IMU observations in the measurement update step for all the displayed results in this section. We performed the same experiment without using the GNSS and IMU observations in the measurement update step. The differences in the results are insignificantly small. Therefore, we do not show these results here.

5 Conclusions and Outlook

In this paper, we presented the mathematical basics and the workflow of an IEKF which makes it possible to determine the trajectory of a UAS better than 5 cm in position (median value) and 0.08° for orientation (median value). It is not mandatory to have continuous GNSS and IMU observations for the implemented IEKF. The trajectory is mainly obtained using laser scanner measurements of building facades, which are modeled as planes in a 3D city model. The laser scanner measurements and the planes of the 3D city model are combined by implicit measurement equations and nonlinear equality constraints within the IEKF.

To demonstrate the functionality and performance of the IEKF implemented, we developed a simulator, which showed that the IEKF is even suited to handle systematically perturbed observations.

Nevertheless, the algorithm demonstrated may be tuned to deal with disadvantageous initial values. A possible starting point can be found in the threshold of the assignment algorithm. Instead of a constant value, this value should be varied regarding the estimated accuracy of the predicted position.

In the future, we plan to evaluate the performance of the IEKF based on real data, especially regarding effects of generalization of the 3D city model and effects of a 3D city model with larger uncertainties. Thus, we can also check whether the assumptions made for the simulation (see Table 1) are also valid in reality. In particular, the performance of the GNSS receiver can deviate strongly from the simulated data with the same precision.

In the IEKF presented, the plane parameters and vertices are introduced as parameters to preserve the topology of the 3D city model. With an increasing number of planes and vertices, the computation becomes inefficient. Here, we plan to reorganize the IEKF presented by means of a dual-state Kalman filter.

As has already been mentioned, we plan to integrate camera measurements into the IEKF. This enables the stabilization of the trajectory in the long term and the improvement of the IEKF's initial behavior. Therefore, we have to integrate the collinearity equations into our approach. Subsequently, we have to integrate the object coordinates of the tie points into the state parameter vector. This emphasizes the need to use a dual-state Kalman filter. By integrating the camera measurements[8] into the IEKF, we must deal with observations captured in different epochs: one or more images observing a tie point were taken in a past epoch and only one image is taken in the current epoch.

Finally, we aim to analyze the benefit of integrating further geometries in addition to the planes of the 3D city model. There are many cylindrical-shaped objects in urban areas, such as lantern, traffic lights or street signs, which may be used in the IEKF.

Acknowledgements This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—NE 1453/5-1, HE 1822/37-1, and as part of the Research Training Group i.c.sens [RTG 2159]. The computations were performed by the compute cluster, which is funded by the Leibniz University Hannover, the Lower Saxony Ministry of Science and Culture (MWK), and DFG.

References

Abmayr T, Härtl F, Hirzinger G, Burschka D, Fröhlich C (2008) A correlation based target finder for terrestrial laser scanning. J Appl Geod 2(3):429. https://doi.org/10.1515/JAG.2008.015

Alkhatib H, Neumann I, Kutterer H (2009) Uncertainty modeling of random and systematic errors by means of Monte Carlo and fuzzy techniques. J Appl Geod 3(2):136. https://doi.org/10.1515/JAG.2009.008

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

Dang T (2007) Kontinuierliche Selbstkalibrierung von Stereokameras. Schriftenreihe/Institut für Mess- und Regelungstechnik, Karlsruher Institut für Technologie vol Nr. 008. Univ.-Verl. Karlsruhe, Karlsruhe

Dang T (2008) An iterative parameter estimation method for observation models with nonlinear constraints. Metrol Meas Syst 15(4):421–432

Denham WF, Pines S (1966) Sequential estimation when measurement function nonlinearity is comparable to measurement error. AIAA J 4(6):1071–1076. https://doi.org/10.2514/3.3606

Dennig D, Bureick J, Link J, Diener D, Hesse C, Neumann I (2017) Comprehensive and highly accurate measurements of crane runways, profiles and fastenings. Sensors (Basel, Switzerland). https://doi.org/10.3390/s17051118

Drixler E (1993) Analyse der Form und Lage von Objekten im Raum. Deutsche Geodätische Kommission. Reihe C. Dissertationen vol Heft Nr. 409. Beck, München

Elkhrachy I, Niemeier W (2006) Stochastic assessment of terrestrial laser scanner. In: Proceedings of the ASPRS annual conference, Reno, United States of America

Ettlinger A, Neuner H, Burgess T (2018) Development of a Kalman filter in the Gauss-Helmert model for reliability analysis in orientation determination with smartphone sensors. Sensors (Basel, Switzerland). https://doi.org/10.3390/s18020414

Forster C, Lynen S, Kneip L, Scaramuzza D (2013) Collaborative monocular slam with multiple micro aerial vehicles. In: 2013 IEEE/RSJ International conference on intelligent robots and systems (IROS), pp 3962–3970. IEEE, Piscataway, NJ. https://doi.org/10.1109/IROS.2013.6696923

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 2:73–80. https://doi.org/10.5194/isprsannals-II-3-W5-73-2015

Gröger G, Kolbe TH, Nagel C, Häfele KH (2012) OGC City Geography Markup Language (CityGML) encoding standard, 2.0.0 edn. Open Geospatial Consortium, Wayland

Hartmann J, Trusheim P, Alkhatib H, Paffenholz JA, Diener D, Neumann I (2018) High accurate pointwise (geo-)referencing of a k-tls based multi-sensor-system. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 4:81–88. https://doi.org/10.5194/isprs-annals-IV-4-81-2018

Hebel M, Arens M, Stilla U (2009) Utilization of 3D city models and airborne laser scanning for terrain-based navigation of helicopters and UAVS. Int Arch Photogramm Remote Sens Spat Inf Sci 38-3/W4:187–192

Hol JD, Schön TB, Luinge H, Slycke PJ, Gustafsson F (2007) Robust real-time tracking by fusing measurements from inertial and vision sensors. J Real Time Image Process 2(2):149–160. https://doi.org/10.1007/s11554-007-0040-2

Holst C, Kuhlmann H, Paffenholz JA, Neumann I (2015) Tls im statischen, stop & go sowie kinematischen einsatz. In: Terrestrisches Laserscanning 2015 (TLS 2015), Beiträge zum ... DVW-Seminar, vol 147. Wißner-Verlag **[Erscheinungsort nicht ermittelbar]**

Inselberg A (1985) The plane with parallel coordinates. Vis Comput 1(2):69–91. https://doi.org/10.1007/BF01898350

Jutzi B, Weinmann M, Meidow J (2013) Improved uav-borne 3D mapping by fusing optical and laserscanner data. ISPRS Int Arch

[8] We do not plan to use stereo cameras.

Photogramm Remote Sens Spat Inf Sci 2:223–228. https://doi.org/10.5194/isprsarchives-XL-1-W2-223-2013

Kaul L, Zlot R, Bosse M (2015) Continuous-time three-dimensional mapping for micro aerial vehicles with a passively actuated rotating laser scanner. J Field Robot. https://doi.org/10.1002/rob.21614

Lee KW, Wijesoma S, Guzmán JI (2007) A constrained SLAM approach to robust and accurate localisation of autonomous ground vehicles. Robot Auton Syst 55(7):527–540. https://doi.org/10.1016/j.robot.2007.02.004

Li-Chee-Ming J, Armenakis C (2013) Determination of UAS trajectory in a known environment from FPV video. Int Arch Photogramm Remote Sens Spat Inf Sci 40-1/W2:247–252

Luhmann T (2013) Close range photogrammetry: 3D imaging techniques. De Gruyter, Berlin

Nguyen V, Harati A, Martinelli A, Siegwart R, Tomatis N (2006) Orthogonal SLAM: a step toward lightweight indoor autonomous navigation. In: 2006 IEEE/RSJ International conference on intelligent robots and systems, pp 5007–5012. https://doi.org/10.1109/IROS.2006.282527

Nüchter A, Borrmann D, Koch P, Kühn M, May S (2015) A man-portable, IMU-free mobile mapping system. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 3:17–23. https://doi.org/10.5194/isprsannals-II-3-W5-17-2015

Paffenholz JA (2012) Direct geo-referencing of 3D point clouds with 3D positioning sensors. In: Zugl.: Hannover, Univ., Diss., 2012, Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover, vol 302. Univ, Hannover

Rump S (1999) INTLAB—INTerval LABoratory. In: Csendes T (ed) Developments in reliable computing. Kluwer Academic Publishers, Dordrecht, pp 77–104. http://www.ti3.tuhh.de/rump/

SBG Systems (2019) Miniature high performance inertial sensors: Elipse 2 series. https://www.sbg-systems.com/wp-content/uploads/Ellipse_Series_Leaflet.pdf. Accessed 14 May 2019

Schuhmacher S, Böhm J (2005) Georeferencing of terrestrial laserscanner data for applications in architectural modeling. https://doi.org/10.18419/opus-3749

Simon D (2010) Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. IET Control Theory Appl 4(8):1303–1318. https://doi.org/10.1049/iet-cta.2009.0032

Soloviev A, Bates D, van Graas F (2007) Tight coupling of laser scanner and inertial measurements for a fully autonomous relative navigation solution. Navigation 54(3):189–205

Steffen R (2013) A robust iterative Kalman filter based on implicit measurement equations. Photogramm Fernerkundung Geoinf 2013(4):323–332. https://doi.org/10.1127/1432-8364/2013/0180

Steffen R, Beder C (2007) Recursive estimation with implicit constraints. In: Hamprecht FA, Schnörr C, Jähne B (eds) Pattern recognition. Lecture notes in computer science, vol 4713. Springer, Berlin, pp 194–203

Tailanian M, Paternain S, Rosa R, Canetti R (2014) Design and implementation of sensor data fusion for an autonomous quadrotor. In: 2014 IEEE International instrumentation and measurement technology conference (I2MTC 2014). IEEE, Piscataway, NJ, pp 1431–1436. https://doi.org/10.1109/I2MTC.2014.6860982

Talaya J, Alamus R, Bosch E, Serra A, Kornus W, Baron A (2004) Integration of a terrestrial laser scanner with GPS/IMU orientation sensors. In: Proceedings of the XXth ISPRS congress, Istanbul, Turkey, vol 1223

Unger J, Rottensteiner F, Heipke C (2016) Integration of a generalised building model into the pose estimation of UAS images. Int Arch Photogramm Remote Sens Spat Inf Sci 89:1057–1064. https://doi.org/10.5194/isprsarchives-XLI-B1-1057-2016

Unger J, Rottensteiner F, Heipke C (2017) Assigning tie points to a generalised building model for UAS image orientation. ISPRS Int Arch Photogramm Remote Sens Spat Inf Sci 2:385–392. https://doi.org/10.5194/isprs-archives-XLII-2-W6-385-2017

Vogel S, Alkhatib H, Bureick J, Moftizadeh R, Neumann I (2019) Georeferencing of laser scanner-based kinematic multi-sensor systems in the context of iterated extended Kalman filters using geometrical constraints. Sensors (Basel, Switzerland). https://doi.org/10.3390/s19102280

Vogel S, Alkhatib H, Neumann I (2016) Accurate indoor georeferencing with kinematic multi sensor systems. In: 2016 International conference on indoor positioning and indoor navigation (IPIN), pp 1–8. https://doi.org/10.1109/IPIN.2016.7743601

Vogel S, Alkhatib H, Neumann I (2018) Iterated extended kalman filter with implicit measurement equation and nonlinear constraints for information-based georeferencing. In: 2018 International conference on information fusion (FUSION), pp 1209–1216. https://doi.org/10.23919/ICIF.2018.8455258