Article

The International Journal of Robotics Research
2015, Vol. 34(3) 314–334

DOI: 10.1177/0278364914554813
ijr.sagepub.com

Keyframe-based visual–inertial odometry using nonlinear optimization

Stefan Leutenegger[1,2], Simon Lynen[2], Michael Bosse[2], Roland Siegwart[2] and Paul Furgale[2]

Abstract

Combining visual and inertial measurements has become popular in mobile robotics, since the two sensing modalities offer complementary characteristics that make them the ideal choice for accurate visual–inertial odometry or simultaneous localization and mapping (SLAM). While historically the problem has been addressed with filtering, advancements in visual estimation suggest that nonlinear optimization offers superior accuracy, while still tractable in complexity thanks to the sparsity of the underlying problem. Taking inspiration from these findings, we formulate a rigorously probabilistic cost function that combines reprojection errors of landmarks and inertial terms. The problem is kept tractable and thus ensuring real-time operation by limiting the optimization to a bounded window of keyframes through marginalization. Keyframes may be spaced in time by arbitrary intervals, while still related by linearized inertial terms. We present evaluation results on complementary datasets recorded with our custom-built stereo visual–inertial hardware that accurately synchronizes accelerometer and gyroscope measurements with imagery. A comparison of both a stereo and monocular version of our algorithm with and without online extrinsics estimation is shown with respect to ground truth. Furthermore, we compare the performance to an implementation of a state-of-the-art stochastic cloning sliding-window filter. This competitive reference implementation performs tightly coupled filtering-based visual–inertial odometry. While our approach declaredly demands more computation, we show its superior performance in terms of accuracy.

Keywords

Visual–inertial odometry, simultaneous localization and mapping (SLAM), robotics, sensor fusion, stereo camera, inertial measurement unit (IMU), keyframes, bundle adjustment

1. Introduction

Visual and inertial measurements offer complementary properties which make them particularly suitable for fusion, in order to address robust and accurate localization and mapping, a primary need for any mobile robotic system. The rich representation of structure projected into an image, together with the accurate short-term estimates by gyroscopes and accelerometers contained in an inertial measurement unit (IMU) have been acknowledged to complement each other, with promising use-cases in airborne (Mourikis and Roumeliotis, 2007; Weiss et al., 2012) and automotive (Li and Mourikis, 2012a) navigation. Moreover, with the availability of these sensors in most smart phones, there is great interest and research activity in effective solutions to visual–inertial simultaneous localization and mapping (SLAM) (Li et al., 2013).

Historically, there have been two main concepts towards approaching the visual–inertial estimation problem: batch *nonlinear optimization* methods and recursive *filtering* methods. While the former jointly minimizes the error originating from integrated IMU measurements and the (reprojection) errors from visual terms (Jung and Taylor, 2001), recursive algorithms commonly use the IMU measurements for state propagation while updates originate from the visual observations (Chai et al., 2002; Roumeliotis et al., 2002).

Batch approaches offer the advantage of repeated linearization of the inherently nonlinear cost terms involved in the visual–inertial state estimation problem and thus they limit linearization errors. For a long time, however, the lack of computational resources made recursive algorithms a favorable choice for online estimation. Nevertheless, both

[1]Department of Computing, Imperial College London, London, UK
[2]Autonomous Systems Laboratory (ASL), ETH Zurich, Switzerland

Corresponding author:
Stefan Leutenegger, Department of Computing, Imperial College London, 180 Queen's Gate, London, SW7 2AZ, UK.
Email: s.leutenegger@imperial.ac.uk

Fig. 1. ETH main building indoor reconstruction of both structure and pose as resulting from our suggested visual–inertial odometry framework (stereo variant in this case, including online camera extrinsics calibration). The stereo-vision plus IMU sensor was walked handheld for 470 m through loops on three floors as well as through staircases. (a) Side view of the ETH main building. (b) Three-dimensional view of the building.

paradigms have recently shown improvements over and compromises towards the other, so that recent work (Indelman et al., 2012; Leutenegger et al., 2013; Nerurkar et al., 2013) showed batch-based algorithms reaching real-time operation and filtering-based methods providing results of nearly equal quality (Mourikis and Roumeliotis, 2007; Li et al., 2013) to batch-based methods. Leaving aside computational demands, batch based methods promise results of higher accuracy compared with filtering approaches, given the inherent algorithmic differences as discussed in detail later in this article.

Apart from the separation into batch and filtering, the visual–inertial fusion approaches found in the literature can be divided into two other categories: *loosely coupled* systems independently estimate the pose by a vision only algorithm and fuse IMU measurements only in a separate estimation step, limiting computational complexity. *Tightly coupled* approaches in contrast include both the measurements from the IMU and the camera into a common problem where all states are jointly estimated, thus considering all correlations amongst them. Comparisons of both approaches, however, show (Leutenegger et al., 2013) that these correlations are key for any high-precision visual–inertial navigation system (VINS), which is also why all high-accuracy visual–inertial estimators presented recently have implemented a tightly-coupled VINS: for example Mourikis and Roumeliotis (2007) proposed an extended Kalman filter (EKF)-based real-time fusion using monocular vision, named multi-state constraint Kalman filter (MSCKF). This work performs impressively with open-loop errors below 0.5% of the distance traveled. We therefore compare our results with a competitive implementation of this sliding window filter using on-the-fly feature marginalization as published by Mourikis et al. (2009). For simpler reference we denote this algorithm by "MSCKF" in the rest of the article, keeping in mind that the available reference implementation does not include *all* of the possible modifications from Li and Mourikis (2012a), Li and Mourikis (2012b), Li et al. (2013) and Hesch et al. (2013).

In this article, which extends our previous work (Leutenegger et al., 2013), we propose a method that respects the aforementioned findings: we advocate *tightly coupled* fusion for best exploitation of all measurements and *nonlinear optimization* where possible rather than filtering, in order to reduce suboptimality due to linearization. Furthermore, the optimization approach allows for employing robust cost functions which may drastically increase accuracy in the presence of outliers that may occasionally occur mostly in the visual part, even after application of sophisticated rejection schemes.

We devise a cost function that combines visual and inertial terms in a fully probabilistic manner. We adopt the concept of *keyframes* due to its successful application in classical vision-only approaches: it is implemented using partial linearization and marginalization, i.e. variable elimination, a compromise towards filtering that is made for real-time compliance and tractability. The keyframe paradigm accounts for drift-free estimation also when slow or no motion at all is present: rather than using an optimization window of time-successive poses, our kept keyframes may be spaced arbitrarily far in time, keeping visual constraints, while still incorporating an IMU term. Our formulation of relative uncertainty between keyframes takes inspiration from RSLAM (Mei et al., 2011), although our parameterization uses global coordinates. We provide a strictly probabilistic derivation of IMU error terms and the respective information matrix, relating successive image frames without explicitly introducing states at IMU rate. At the system level, we developed both the hardware and the algorithms for accurate real-time SLAM, including robust keypoint matching, bootstrapping and outlier rejection using inertial cues.

Figure 1 shows the output of our stereo visual–inertial odometry algorithm as run on an indoor dataset: the stereo-vision plus IMU sensor was walked for 470 m through several floors and staircases in the ETH main building. Along with the state consisting of pose, speed, and IMU biases, we also obtain an impression of the environment represented as a sparse map of 3D landmarks. Note that map and path are automatically aligned with gravity thanks to tightly coupled IMU fusion.

In relation to the conference paper (Leutenegger et al., 2013), we make the following main additional contributions.

- After having shown the superior performance of the suggested method compared with a loosely coupled

approach, we present extensive evaluation results with respect to a stochastic cloning sliding window filter (following the MSCKF implementation of Mourikis et al. (2009), which includes first-estimate Jacobians) in terms of accuracy on different motion profiles. Our algorithm consistently outperforms the filtering-based method, while it admittedly incurs higher computational cost. To the best of the authors' knowledge, such a direct comparison of visual–inertial state estimation algorithms as suggested by different research groups is novel to the field.

- Our framework has been extended to be used with a monocular camera setup. We present the necessary adaptations concerning the estimation and bootstrapping parts. The monocular version was needed for fair comparison with the reference implementation of the MSCKF algorithm which is currently only published in a monocular form. The result is a generic N-camera ($N \geq 1$) visual–inertial odometry framework. In the stereo version, the performance will gradually transform into the monocular case when the ratio between camera baseline and distance to structure becomes small.
- We present the formulation for online camera extrinsics estimation that may be applied after standard intrinsics calibration. Evaluation results demonstrate the applicability of this method, when initializing with inaccurate camera pose estimates with respect to the IMU.
- We make an honest attempt to present our work to a level of detail that would allow the reader to re-implement our framework.
- Various new datasets featuring individual characteristics in terms of motion, appearance, and scene depth were recorded with our new hardware iteration ranging from hand-held indoor motion to bicycle riding. The comprehensive evaluation demonstrates superior performance compared with our previously published results, owing to better calibration and hardware synchronization available, as well as to algorithmic and software-level adaptations.

The remainder of this work is structured as follows. In Section 2 we provide a more detailed overview of how our work relates to existing literature and differentiates itself. Section 3 introduces the notation and definitions used throughout this article. The nonlinear error terms from camera and IMU measurements are described in depth in Section 4, which is then followed by an overview of frontend processing and initialization in Section 5. As a last key element of the method, Section 6 introduces how the keyframe concept is applied by marginalization. Section 7 describes the experimental setup, evaluation scheme and presents extensive results on the different datasets.

2. Related work

The vision-only algorithms which form the foundation for today's VINS can be categorized into batch structure-from-motion (SfM) and filtering-based methods. Due to computational constraints, for a long time, vision-based real-time odometry or SLAM algorithms such as those presented by Davison (2003) were only possible using a filtering approach. Subsequent research (Strasdat et al., 2010), however, has shown that nonlinear optimization based approaches, as commonly used for offline SfM, can provide better accuracy for a similar computational work when compared to filtering approaches, given that the structural sparsity of the problem is preserved. Henceforth, it has been popular to maintain a relatively sparse graph of keyframes and their associated landmarks subject to nonlinear optimizations (Klein and Murray, 2007).

The earliest results in VINS originate from the work of Jung and Taylor (2001) for (spline-based) batch and of Chai et al. (2002) and Roumeliotis et al. (2002) for filtering-based approaches. Subsequently, a variety of filtering-based approaches have been published based on EKFs (Kim and Sukkarieh, 2007; Mourikis and Roumeliotis, 2007; Li and Mourikis, 2012a; Weiss et al., 2012; Lynen et al., 2013), iterated EKFs (IEKFs) (Strelow and Singh, 2003, 2004) and unscented Kalman filters (UKFs) (Shin and El-Sheimy, 2004; Ebcin and Veth, 2007; Kelly and Sukhatme, 2011) to name a few, which over the years showed an impressive improvement in precision and a reduction computational complexity. Today such six-degree-of-freedom (6-DoF) visual–inertial estimation systems can be run online on consumer mobile devices (Li and Mourikis, 2012c; Li et al., 2013).

In order to limit computational complexity, many works follow the loosely coupled approach. Konolige et al. (2011) integrate IMU measurements as independent inclinometer and relative yaw measurements into an optimization problem using stereo vision measurements. In contrast, Weiss et al. (2012) use vision-only pose estimates as updates to an EKF with indirect IMU propagation. Similar approaches can be followed for loosely coupled batch based algorithms such as in Ranganathan et al. (2007) and Indelman et al. (2012), where relative stereo pose estimates are integrated into a factor-graph with nonlinear optimization including inertial terms and absolute GPS measurements. It is well known that loosely coupled approaches are inherently sub-optimal since they disregard correlations amongst internal states of different sensors.

A notable contribution in the area of filtering-based VINS is the work of Mourikis and Roumeliotis (2007) who proposed an EKF-based real-time fusion using monocular vision, called the MSCKF which performs nonlinear triangulation of landmarks from a set of camera poses over time before using them in the EKF update. This contrasts with other works that only use visual constraints between pairwise camera poses (Bayard and Brugarolas, 2005). Mourikis and Roumeliotis (2007) also show how the correlations between errors of the landmarks and the camera locations, which are introduced by using the estimated camera poses for triangulation, can be eliminated and thus result in an estimator which is consistent and optimal up to linearization errors. Another monocular visual–inertial

filter was proposed by Jones and Soatto (2011), presenting results on a long outdoor trajectory including IMU to camera calibration and loop closure. Li and Mourikis (2013) showed that further increases in the performance of the MSCKF are attainable by switching between the landmark processing model, as used in the MSCKF, and the full estimation of landmarks, as employed by EKF-SLAM.

Further improvements and extensions to both loosely and tightly coupled filtering-based approaches include an alternative rotation parameterization (Li and Mourikis, 2012b), inclusion of rolling shutter cameras (Jia and Evans, 2012; Li et al., 2013), offline (Lobo and Dias, 2007; Mirzaei and Roumeliotis, 2007, 2008) and online (Jones and Soatto, 2011; Kelly and Sukhatme, 2011; Dong-Si and Mourikis, 2012; Weiss et al., 2012) calibration of the relative position and orientation of camera and IMU.

In order to benefit from increased accuracy offered by re-linearization in batch optimization, recent work focused on approximating the batch problem in order to allow real-time operation. Approaches to keep the problem tractable for online estimation can be separated into three groups (Nerurkar et al., 2013). First, incremental approaches, such as the factor-graph-based algorithms by Bryson et al. (2009); Kaess et al. (2012), apply incremental updates to the problem while factorizing the associated information matrix of the optimization problem or the measurement Jacobian into square-root form (Bryson et al., 2009; Indelman et al., 2012). Second, fixed-lag smoother or sliding-window filter approaches (Sibley et al., 2010; Dong-Si and Mourikis, 2011; Huang et al., 2011) consider only poses from a fixed time interval in the optimization. Poses and landmarks which fall outside the window are marginalized with their corresponding measurements being dropped. Forming nonlinear constraints between different optimization parameters in the marginalization step however destroys the sparsity of the problem, such that the window size has to be kept fairly small for real-time performance. The smaller the window, however, the smaller the benefit of repeated re-linearization. Third, keyframe-based approaches preserve sparsity by maintaining only a subset of camera poses and landmarks and discard (rather than marginalize) intermediate quantities. Nerurkar et al. (2013) present an efficient offline MAP algorithm which uses all information from non-keyframes and landmarks to form constraints between keyframes by marginalizing a set of frames and landmarks without impacting the sparsity of the problem. While this form of marginalization shows small errors when compared with the full batch MAP estimator, we target a version with a fixed window size suitable for online and real-time operations. In this article and our previous work (Leutenegger et al., 2013) we therefore drop measurements from non-keyframes and marginalize the respective state. When keyframes drop out of the window over time, we marginalize the respective states and some landmarks commonly observed to form a (linear) prior for a remaining sub-part of the optimization problem. Our approximation scheme strictly keeps the sparsity of the original problem. This is in contrast to, e.g., Sibley et al. (2010), who accept some loss of sparsity due to marginalization. The latter sliding window filter, in a visual–inertial variant, is used for comparison by Li and Mourikis (2012a): it proves to perform better than the original MSCKF, but interestingly, an improved MSCKF variant using first-estimate Jacobians yields even better results. We aim at performing similar comparisons between an MSCKF implementation, (that includes the use first estimate Jacobians) and our keyframe as well as optimization-based algorithm.

Apart from the differentiation between batch and filtering approaches, it has been a major interest to increase the estimation accuracy by studying the observability properties of VINS. There is substantial work on the observability properties given a particular combination of sensors or measurements (Martinelli, 2011; Weiss, 2012) or only using data from a reduced set of IMU axes (Martinelli, 2014). Global unobservability of yaw and position, as well as growing uncertainty with respect to an initial pose of reference are intrinsic to the visual–inertial estimation problem (Hesch et al., 2012b, 2013; Huang et al., 2013). This property is therefore of particular interest when comparing filtering approaches to batch algorithms: the representation of pose and its uncertainty in a global frame of reference usually becomes numerically problematic as the uncertainty for parts of the state undergoes unbounded growth, while remaining low for the observable sub-parts of the state. Our batch approach therefore uses a formulation of relative uncertainty of keyframes to avoid expressing global uncertainty.

Unobservability of the VINS problem poses a particular challenge to filtering approaches where repeated linearization is typically not possible: Huang et al. (2009) have shown that these linearization errors may erroneously render parts of the estimated state numerically observable. Hesch et al. (2012a) and others (Huang et al., 2011; Hesch et al., 2012b; Kottas et al., 2012; Hesch et al., 2013; Huang et al., 2013) derived formulations allowing them to choose the linearization points of the VINS system in such a way that the observability properties of the linearized and nonlinear system are equal. In our proposed algorithm, we employ first-estimate Jacobians, i.e. whenever linearization of a variable is employed, we fix the linearization point for any subsequent linearization involving that particular variable.

3. Notation and definitions

3.1. Notation

We employ the following notation throughout this work: $\underrightarrow{\mathcal{F}}_A$ denotes a reference frame A; a point P represented in frame $\underrightarrow{\mathcal{F}}_A$ is written as position vector ${}_A\mathbf{r}_P$, or ${}_A\boldsymbol{r}_P$ when in homogeneous coordinates. A transformation between frames is represented by a homogeneous transformation matrix $\boldsymbol{T}_{AB}$ that transforms the coordinate representation of

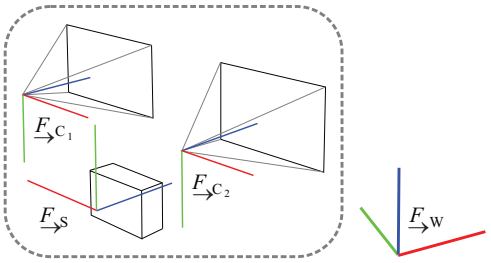

Fig. 2. Coordinate frames involved in the hardware setup used: two cameras are placed as a stereo setup with respective frames, $\mathcal{F}_{C_i}, i \in \{1,2\}$. IMU data is acquired in $\mathcal{F}_S$. The algorithms estimate the position and orientation of $\mathcal{F}_S$ with respect to the world (inertial) frame $\mathcal{F}_W$.

homogeneous points from $\mathcal{F}_B$ to $\mathcal{F}_A$. Its rotation matrix part is written as $\mathbf{C}_{AB}$; the corresponding quaternion is written as $\mathbf{q}_{AB} = [\boldsymbol{\epsilon}^{\mathrm{T}}, \eta]^{\mathrm{T}} \in S^3$, $\boldsymbol{\epsilon}$ and η representing the imaginary and real parts. We adopt the notation introduced by Barfoot et al. (2011): concerning the quaternion multiplication $\mathbf{q}_{AC} = \mathbf{q}_{AB} \otimes \mathbf{q}_{BC}$, we introduce a left-hand side compound operator $[.]^+$ and a right-hand side operator $[.]^\oplus$ that output matrices such that $\mathbf{q}_{AC} = [\mathbf{q}_{AB}]^+ \mathbf{q}_{BC} = [\mathbf{q}_{BC}]^\oplus \mathbf{q}_{AB}$. Taking velocity as an example of a physical quantity represented in frame $\mathcal{F}_A$ that relates frame $\mathcal{F}_B$ and $\mathcal{F}_C$, we write ${}_A\mathbf{v}_{BC}$, i.e. the velocity of frame $\mathcal{F}_B$ with respect to $\mathcal{F}_C$.

3.2. Frames

The performance of the proposed method is evaluated using an IMU and camera setup schematically depicted in Figure 2. It is used both in monocular and stereo mode, where we want to emphasize that our methodology is generic enough to handle an N-camera setup. Inside the tracked body that is represented relative to an inertial frame, $\mathcal{F}_W$, we distinguish camera frames, $\mathcal{F}_{C_i}$ (subscripted with $i = 1,\ldots,N$), and the IMU sensor frame, $\mathcal{F}_S$.

3.3. States

The variables to be estimated comprise the robot states at the image times (index k) $\mathbf{x}_{\mathrm{R}}^k$ and landmarks $\mathbf{x}_{\mathrm{L}}$. $\mathbf{x}_{\mathrm{R}}$ holds the robot position in the inertial frame ${}_W\mathbf{r}_S$, the body orientation quaternion $\mathbf{q}_{WS}$, the velocity expressed in the sensor frame ${}_S\mathbf{v}_{WS}$ (written in short as ${}_S\mathbf{v}$), as well as the biases of the gyroscopes $\mathbf{b}_{\mathrm{g}}$ and the biases of the accelerometers $\mathbf{b}_{\mathrm{a}}$. Thus, $\mathbf{x}_{\mathrm{R}}$ is written as

$$\mathbf{x}_{\mathrm{R}} := \left[{}_W\mathbf{r}_S^{\mathrm{T}}, \mathbf{q}_{WS}^{\mathrm{T}}, {}_S\mathbf{v}^{\mathrm{T}}, \mathbf{b}_{\mathrm{g}}^{\mathrm{T}}, \mathbf{b}_{\mathrm{a}}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^3 \times S^3 \times \mathbb{R}^9 \qquad (1)$$

Furthermore, we use a partition into the pose states $\mathbf{x}_{\mathrm{T}} := [{}_W\mathbf{r}_S^{\mathrm{T}}, \mathbf{q}_{WS}^{\mathrm{T}}]^{\mathrm{T}}$ and the speed/bias states $\mathbf{x}_{\mathrm{sb}} := [{}_S\mathbf{v}^{\mathrm{T}}, \mathbf{b}_{\mathrm{g}}^{\mathrm{T}}, \mathbf{b}_{\mathrm{a}}^{\mathrm{T}}]^{\mathrm{T}}$.

The jth landmark is represented in homogeneous (World) coordinates: $\mathbf{x}_{\mathrm{L}^j} := {}_W\boldsymbol{l}^j \in \mathbb{R}^4$. At this point, we set the fourth component to one.

Optionally, we may include camera extrinsics estimation as part of an online calibration process. Camera extrinsics denoted $\mathbf{x}_{\mathrm{C_i}} := [{}_S\mathbf{r}_{C_i}^{\mathrm{T}}, \mathbf{q}_{SC_i}^{\mathrm{T}}]^{\mathrm{T}}$ can either be treated as constant entities to be calibrated or time-varying states subjected to a first-order Gaussian process allowing to track changes that may occur e.g. due to temperature-induced mechanical deformation of the setup.

In general, the states live in a manifold, therefore we use a perturbation in tangent space $\mathfrak{g}$ and employ the group operator $\boxplus$, that is not commutative in general, the exponential exp and logarithm log. Now, we can define the perturbation $\delta\mathbf{x} := \mathbf{x} \boxplus \overline{\mathbf{x}}^{-1}$ around the estimate $\overline{\mathbf{x}}$. We use a minimal coordinate representation $\delta\boldsymbol{\chi} \in \mathbb{R}^{\dim \mathfrak{g}}$. A bijective mapping $\Phi : \mathbb{R}^{\dim \mathfrak{g}} \to \mathfrak{g}$ transforms from minimal coordinates to tangent space. Thus, we obtain the transformations from and to minimal coordinates:

$$\delta\mathbf{x} = \exp(\Phi(\delta\boldsymbol{\chi})) \qquad (2)$$

$$\delta\boldsymbol{\chi} = \Phi^{-1}(\log(\delta\mathbf{x})) \qquad (3)$$

Concretely, we use the minimal (3D) axis-angle perturbation of orientation $\delta\boldsymbol{\alpha} \in \mathbb{R}^3$ which can be converted into its quaternion equivalent $\delta\mathbf{q}$ via the exponential map:

$$\delta\mathbf{q} := \exp\left(\begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\alpha} \\ 0 \end{bmatrix}\right) = \begin{bmatrix} \mathrm{sinc}\left\|\frac{\delta\boldsymbol{\alpha}}{2}\right\| \frac{\delta\boldsymbol{\alpha}}{2} \\ \cos\left\|\frac{\delta\boldsymbol{\alpha}}{2}\right\| \end{bmatrix} \qquad (4)$$

Therefore, using the group operator $\otimes$, we write $\mathbf{q}_{WS} = \delta\mathbf{q} \otimes \overline{\mathbf{q}}_{WS}$. Note that linearization of the exponential map around $\delta\boldsymbol{\alpha} = \mathbf{0}$ yields

$$\delta\mathbf{q} \approx \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\alpha} \\ 1 \end{bmatrix} = \boldsymbol{\iota} + \frac{1}{2}\begin{bmatrix} \mathbf{I}_3 \\ \mathbf{0}_{1\times 3} \end{bmatrix}\delta\boldsymbol{\alpha} \qquad (5)$$

where $\boldsymbol{\iota}$ denotes the identity quaternion. We obtain the minimal robot error state vector

$$\delta\boldsymbol{\chi}_{\mathrm{R}} = \left[\delta\mathbf{p}^{\mathrm{T}}, \delta\boldsymbol{\alpha}^{\mathrm{T}}, \delta\mathbf{v}^{\mathrm{T}}, \delta\mathbf{b}_{\mathrm{g}}^{\mathrm{T}}, \delta\mathbf{b}_{\mathrm{a}}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{15} \qquad (6)$$

Analogously to the robot state decomposition $\mathbf{x}_{\mathrm{T}}$ and $\mathbf{x}_{\mathrm{sb}}$, we use the pose error state $\delta\boldsymbol{\chi}_{\mathrm{T}} := [\delta\mathbf{p}^{\mathrm{T}}, \delta\boldsymbol{\alpha}^{\mathrm{T}}]^{\mathrm{T}}$ and the speed/bias error state $\delta\boldsymbol{\chi}_{\mathrm{sb}} := [\delta\mathbf{v}^{\mathrm{T}}, \delta\mathbf{b}_{\mathrm{g}}^{\mathrm{T}}, \delta\mathbf{b}_{\mathrm{a}}^{\mathrm{T}}]^{\mathrm{T}}$.

As a landmark perturbation, we use a simple Euclidean version $\delta\boldsymbol{\beta} \in \mathbb{R}^3$ that is applied as $\delta\mathbf{l} := [\delta\boldsymbol{\beta}^{\mathrm{T}}, 0]^{\mathrm{T}}$ by addition.

4. Batch visual SLAM with inertial terms

In this section, we present our approach of incorporating inertial measurements into batch visual SLAM. In visual

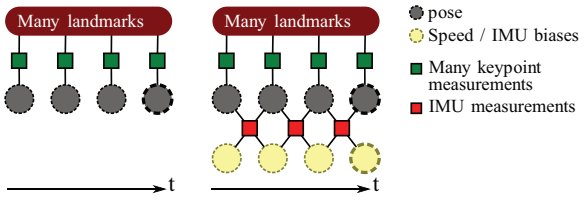

Fig. 3. Graphs of the state variables and measurements involved in the visual SLAM problem (left) versus visual–inertial SLAM (right): incorporating inertial measurements introduces temporal constraints, and necessitates a state augmentation by the robot speed as well as IMU biases.

odometry and SLAM, a nonlinear optimization is formulated to find the camera poses and landmark positions by minimizing the reprojection error of landmarks observed in camera frames. Figure 3 shows the respective graph representation inspired by Thrun and Montemerlo (2006): it displays measurements as edges with square boxes and estimated quantities as round nodes.

As soon as inertial measurements are introduced, they not only create temporal constraints between successive poses, but also between successive speed and IMU bias estimates of both accelerometers and gyroscopes by which the robot state vector is augmented.

We seek to formulate the visual–inertial localization and mapping problem as one joint optimization of a cost function $J(\mathbf{x})$ containing both the (weighted) reprojection errors $\mathbf{e}_\mathrm{r}$ and the (weighted) temporal error term from the IMU $\mathbf{e}_\mathrm{s}$:

$$J(\mathbf{x}) := \underbrace{\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j\in\mathcal{J}(i,k)} \mathbf{e}_\mathrm{r}^{i,j,k\,\mathrm{T}}\mathbf{W}_\mathrm{r}^{i,j,k}\mathbf{e}_\mathrm{r}^{i,j,k}}_{\text{visual}} + \underbrace{\sum_{k=1}^{K-1}\mathbf{e}_\mathrm{s}^{k\,\mathrm{T}}\mathbf{W}_\mathrm{s}^{k}\mathbf{e}_\mathrm{s}^{k}}_{\text{inertial}} \tag{7}$$

where i is the camera index of the assembly, k denotes the camera frame index, and j denotes the landmark index. The indices of landmarks visible in the kth frame and the ith camera are written as the set $\mathcal{J}(i,k)$. Furthermore, $\mathbf{W}_\mathrm{r}^{i,j,k}$ represents the information matrix of the respective landmark measurement, and $\mathbf{W}_\mathrm{s}^{k}$ the information of the kth IMU error.

Throughout our work, we employ the Google Ceres optimizer (see Agarwal et al., 2010) integrated with our real-time capable C++ software infrastructure.

In the following, we will present the reprojection error formulation. Afterwards, an overview on IMU kinematics combined with bias term modeling is given, upon which we base the IMU error term.

4.1. Reprojection error formulation

We use a rather standard formulation of the reprojection error adapted with minor modifications from Furgale (2011):

$$\mathbf{e}_\mathrm{r}^{i,j,k} = \mathbf{z}^{i,j,k} - \mathbf{h}_i\left(\boldsymbol{T}_{CiS}^{k}\,\boldsymbol{T}_{SW}^{k}\,{}_W\boldsymbol{l}^{j}\right) \tag{8}$$

Hereby $\mathbf{h}_i(\cdot)$ denotes the camera projection model (which may include distortion) and $\mathbf{z}^{i,j,k}$ stands for the measurement image coordinates. We also provide the Jacobians here, since they are not only needed for efficient solving, but also play a central role in the marginalization step explained in Section 6:

$$\frac{\partial\mathbf{e}_\mathrm{r}^{i,j,k}}{\partial\delta\boldsymbol{\chi}_\mathrm{T}^{k}} = \mathbf{J}_{\mathrm{r},i}\,\bar{\boldsymbol{T}}_{C_iS}^{k}\begin{bmatrix}\bar{\mathbf{C}}_{SW}^{k}\,{}_W\bar{\boldsymbol{l}}_4^{j} & \bar{\mathbf{C}}_{SW}^{k}\left[{}_W\bar{\boldsymbol{l}}_{1:3}^{j} - {}_W\mathbf{r}_{SW}^{k}\,{}_W\bar{\boldsymbol{l}}_4^{j}\right]^\times \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3}\end{bmatrix} \tag{9}$$

$$\frac{\partial\mathbf{e}_\mathrm{r}^{i,j,k}}{\partial\delta\boldsymbol{\chi}_\mathrm{L}^{j}} = -\,\mathbf{J}_{\mathrm{r},i}\,\bar{\boldsymbol{T}}_{C_iS}^{k}\begin{bmatrix}\bar{\mathbf{C}}_{SW}^{k} \\ \mathbf{0}_{1\times3}\end{bmatrix} \tag{10}$$

$$\frac{\partial\mathbf{e}_\mathrm{r}^{i,j,k}}{\partial\delta\boldsymbol{\chi}_\mathrm{Ci}^{k}} = \mathbf{J}_{\mathrm{r},i}\begin{bmatrix}\mathbf{C}_{C_iS}^{k}\,{}_S\bar{\boldsymbol{l}}_4^{j}\mathbf{C}_{C_iS}^{k}\left[{}_S\bar{\boldsymbol{l}}_{1:3}^{j} - {}_S\mathbf{r}_{C_iS}^{k}\,{}_S\bar{\boldsymbol{l}}_4^{j}\right]^\times & \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3}\end{bmatrix} \tag{11}$$

where $\mathbf{J}_{\mathrm{r},i}$ denotes the Jacobian matrix of the projection $\mathbf{h}_i(\cdot)$ into the ith camera (including distortion) with respect to a landmark in homogeneous coordinates and variables with an overbar represent our current guess. Our framework currently supports radial-tangential as well as equidistant distortion models.

4.2. IMU kinematics and bias model

Before being able to formulate the nonlinear IMU term, we overview the differential equations that describe IMU kinematics and bias evolution. The model is commonly used in estimation with IMUs originating from Savage (1998) using similar simplifications for microelectromechanical system (MEMS)-IMUs as in Shin and El-Sheimy (2004).

4.2.1. Nonlinear model. Under the assumption that the measured effects of the Earth's rotation are small compared with the gyroscope accuracy, we can write the IMU kinematics combined with simple dynamic bias models as

$$\begin{aligned}
{}_W\dot{\mathbf{r}}_S &= \mathbf{C}_{WS}\,{}_S\mathbf{v} \\
\dot{\mathbf{q}}_{WS} &= \frac{1}{2}\boldsymbol{\Omega}({}_S\boldsymbol{\omega})\mathbf{q}_{WS} \\
{}_S\dot{\mathbf{v}} &= {}_S\widetilde{\mathbf{a}} + \mathbf{w}_\mathrm{a} - \mathbf{b}_\mathrm{a} + \mathbf{C}_{SW}\,{}_W\mathbf{g} - ({}_S\boldsymbol{\omega})\times{}_S\mathbf{v} \\
\dot{\mathbf{b}}_\mathrm{g} &= \mathbf{w}_{b_\mathrm{g}} \\
\dot{\mathbf{b}}_\mathrm{a} &= -\frac{1}{\tau}\mathbf{b}_\mathrm{a} + \mathbf{w}_{b_\mathrm{a}}
\end{aligned} \tag{12}$$

where the elements of $\mathbf{w} := [\mathbf{w}_\mathrm{g}^\mathrm{T}, \mathbf{w}_\mathrm{a}^\mathrm{T}, \mathbf{w}_{b_\mathrm{g}}^\mathrm{T}, \mathbf{w}_{b_\mathrm{a}}^\mathrm{T}]^\mathrm{T}$ are each uncorrelated zero-mean Gaussian white noise processes. Here ${}_S\widetilde{\mathbf{a}}$ are accelerometer measurements and ${}_W\mathbf{g}$ represents the Earth's gravitational acceleration vector. The gyro bias, is modeled as random walk, and in contrast, the accelerometer bias is modeled as a bounded random walk with time constant $\tau > 0$. The matrix $\boldsymbol{\Omega}$ is formed from the estimated angular rate ${}_S\boldsymbol{\omega} = {}_S\tilde{\boldsymbol{\omega}} + \mathbf{w}_\mathrm{g} - \mathbf{b}_\mathrm{g}$, with gyro measurement ${}_S\tilde{\boldsymbol{\omega}}$:

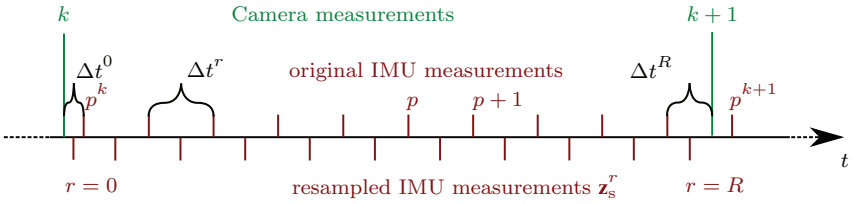

Fig. 4. Different rates of IMU and camera: one IMU term uses all accelerometer and gyro readings between successive camera measurements.

$$\mathbf{\Omega}\left({}_S\boldsymbol{\omega}\right) := \begin{bmatrix} -{}_S\boldsymbol{\omega} \\ 0 \end{bmatrix}^{\oplus} \tag{13}$$

4.2.2. Linearized model of the error states. The linearized version of the above equation around $\overline{\mathbf{x}}_R$ will play a major role in the marginalization step. We therefore review it here briefly: the error dynamics take the form

$$\delta\dot{\boldsymbol{\chi}}_{\mathrm{R}} \approx \mathbf{F}_c(\overline{\mathbf{x}}_{\mathrm{R}})\delta\boldsymbol{\chi}_{\mathrm{R}} + \mathbf{G}(\overline{\mathbf{x}}_{\mathrm{R}})\mathbf{w} \tag{14}$$

where $\mathbf{G}$ is straightforward to derive and

$$\mathbf{F}_c = \begin{bmatrix} \mathbf{0}_{3\times3} & \left[\overline{\mathbf{C}}_{WS}\overline{\mathbf{v}}\right]^{\times} & \overline{\mathbf{C}}_{WS} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \overline{\mathbf{C}}_{WS} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\overline{\mathbf{C}}_{WS}[{}_W\mathbf{g}]^{\times} & -[{}_S\overline{\boldsymbol{\omega}}]^{\times} & -[{}_S\overline{\mathbf{v}}]^{\times} & -\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\frac{1}{\tau}\mathbf{I}_3 \end{bmatrix} \tag{15}$$

where $[.]^{\times}$ denotes the skew-symmetric cross-product matrix associated with a vector. Overbars generally stand for evaluation of the respective symbols with current estimates.

4.3. Formulation of the IMU measurement error term

Figure 4 illustrates the difference in measurement rates with camera measurements taken at time steps k and $k+1$, as well as faster IMU measurements that are not necessarily synchronized with the camera measurements. Note also the introduction of a local time index $r = 0,\ldots,R$ between camera measurements, along with respective time increments Δt^r.

We need the IMU error term $\mathbf{e}_{\mathrm{s}}^k(\mathbf{x}_{\mathrm{R}}^k, \mathbf{x}_{\mathrm{R}}^{k+1}, \mathbf{z}_{\mathrm{s}}^k)$ to be a function of robot states at steps k and $k+1$ as well as of all the IMU measurements in-between these time instances (comprising accelerometer and gyro readings) summarized as $\mathbf{z}_{\mathrm{s}}^k$. Hereby we have to assume an approximate normal conditional probability density f for given robot states at camera measurements k and $k+1$:

$$f\left(\mathbf{e}_{\mathrm{s}}^k | \mathbf{x}_{\mathrm{R}}^k, \mathbf{x}_{\mathrm{R}}^{k+1}\right) \approx \mathcal{N}\left(\mathbf{0}, \mathbf{R}_{\mathrm{s}}^k\right) \tag{16}$$

We are employing the propagation equations above to formulate a prediction $\hat{\mathbf{x}}_{\mathrm{R}}^{k+1}\left(\mathbf{x}_{\mathrm{R}}^k, \mathbf{z}_{\mathrm{s}}^k\right)$ with associated conditional covariance $\mathbf{P}\left(\delta\hat{\mathbf{x}}_{\mathrm{R}}^{k+1} | \mathbf{x}_{\mathrm{R}}^k, \mathbf{z}_{\mathrm{s}}^k\right)$. The respective computation requires numeric integration. As is common in related literature (Mourikis and Roumeliotis, 2007), we applied the classical Runge–Kutta method, in order to obtain discrete time nonlinear state transition equations $\mathbf{f}_{\mathrm{d}}(\overline{\mathbf{x}}_{\mathrm{R}}^k)$ and the error state transition matrix $\mathbf{F}_{\mathrm{d}}(\overline{\mathbf{x}}_{\mathrm{R}}^k)$. The latter is found by integrating $\delta\dot{\boldsymbol{\chi}}_{\mathrm{R}} = \mathbf{F}_c(\overline{\mathbf{x}}_{\mathrm{R}})\delta\boldsymbol{\chi}_{\mathrm{R}}$ over Δt^r keeping $\delta\boldsymbol{\chi}_{\mathrm{R}}$ symbolic.

Using the prediction, we can now formulate the IMU error term as

$$\mathbf{e}_{\mathrm{s}}^k\left(\mathbf{x}_{\mathrm{R}}^k, \mathbf{x}_{\mathrm{R}}^{k+1}, \mathbf{z}_{\mathrm{s}}^k\right) = \begin{bmatrix} {}_W\hat{\mathbf{r}}_S^{k+1} - {}_W\mathbf{r}_S^{k+1} \\ 2\left[\hat{\mathbf{q}}_{WS}^{k+1} \otimes \mathbf{q}_{WS}^{k+1\,-1}\right]_{1:3} \\ \hat{\mathbf{x}}_{\mathrm{sb}}^{k+1} - \mathbf{x}_{\mathrm{sb}}^{k+1} \end{bmatrix} \in \mathbb{R}^{15} \tag{17}$$

This is simply the difference between the prediction based on the previous state and the actual state, except for orientation, where we use a simple multiplicative minimal error.

Next, upon application of the error propagation law, the associated information matrix $\mathbf{W}_{\mathrm{s}}^k$ is found as

$$\mathbf{W}_{\mathrm{s}}^k = \mathbf{R}_{\mathrm{s}}^{k^{-1}} = \left(\frac{\partial\mathbf{e}_{\mathrm{s}}^k}{\partial\delta\hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1}}\mathbf{P}\left(\delta\hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1} | \mathbf{x}_{\mathrm{R}}^k, \mathbf{z}_{\mathrm{s}}^k\right)\frac{\partial\mathbf{e}_{\mathrm{s}}^k}{\partial\delta\hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1}}^{\mathrm{T}}\right)^{-1} \tag{18}$$

The Jacobian $\frac{\partial\mathbf{e}_{\mathrm{s}}^k}{\partial\delta\hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1}}$ is straightforward to obtain but non-trivial, since the orientation error will be non-zero in general:

$$\frac{\partial\mathbf{e}_{\mathrm{s}}^k}{\partial\delta\hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1}} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times9} \\ \mathbf{0}_{3\times3} & \left[\hat{\mathbf{q}}_{WS}^{k+1} \otimes \mathbf{q}_{WS}^{k+1\,-1}\right]_{1:3,\,1:3}^{\oplus} & \mathbf{0}_{3\times9} \\ \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} & \mathbf{I}_9 \end{bmatrix} \tag{19}$$

Finally, the Jacobians with respect to $\delta\boldsymbol{\chi}_{\mathrm{R}}^k$ and $\delta\boldsymbol{\chi}_{\mathrm{R}}^{k+1}$ will be needed for efficient solution of the optimization problem. While differentiating with respect to $\delta\boldsymbol{\chi}_{\mathrm{R}}^{k+1}$ is straightforward (but non-trivial), some attention is given to the other Jacobian. Recall that the IMU error term (17) is calculated by iteratively applying the prediction. Differentiation with respect to the state $\delta\boldsymbol{\chi}_{\mathrm{R}}^k$, thus leads to application of the chain rule, yielding

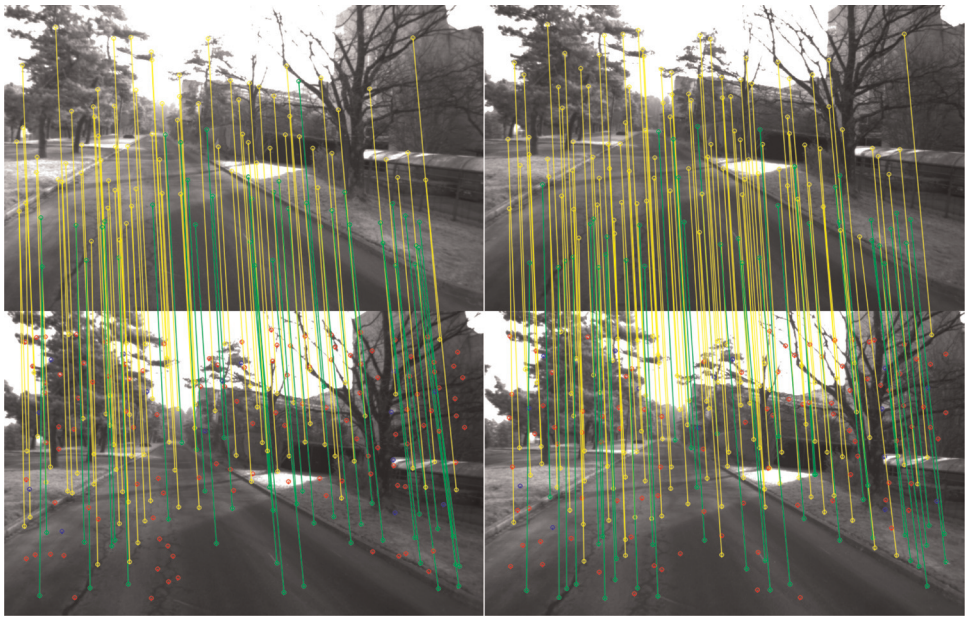

Fig. 5. Visualization of typical data association on a bicycle dataset: current stereo image pair (bottom) with match lines to the newest keyframe (top). Green stands for a 3D–2D match, yellow for 2D–2D match, blue for keypoints with left–right stereo match only, and red keypoints are unmatched.

$$\frac{\partial \mathbf{e}_{\mathrm{s}}^{k}}{\partial \delta \boldsymbol{\chi}_{\mathrm{R}}^{k}} = \frac{\partial \mathbf{e}_{\mathrm{s}}^{k}}{\partial \delta \hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1}} \mathbf{F}_{\mathrm{d}}(\overline{\widehat{\mathbf{x}}}_{\mathrm{R}}^{R}, \Delta t^{R}) \mathbf{F}_{\mathrm{d}}(\overline{\widehat{\mathbf{x}}}_{\mathrm{R}}^{R-1}, \Delta t^{R-1}) \ldots \mathbf{F}_{\mathrm{d}}(\overline{\widehat{\mathbf{x}}}_{\mathrm{R}}^{1}, \Delta t^{1}) \mathbf{F}_{\mathrm{d}}(\overline{\mathbf{x}}_{\mathrm{R}}^{k}, \Delta t^{0}) \tag{20}$$

5. Frontend overview

This section overviews the image processing steps and data association along with outlier detection and initialization of landmarks and states.

5.1. Keypoint detection, matching, and variable initialization

Our processing pipeline employs a customized multi-scale SSE-optimized Harris corner detector (Harris and Stephens, 1988) followed by BRISK descriptor extraction (Leutenegger et al., 2011). The detection scheme favors a uniform keypoint distribution in the image by gradually suppressing corners with weaker corner response close to a stronger corner. BRISK would allow automatic orientation detection; however, better matching results are obtained by extracting descriptors oriented along the gravity direction that is projected into the image. This direction is globally observable thanks to IMU fusion.

As a first step to initialization and matching, we propagate the last pose using acquired IMU measurements in order to obtain a preliminary uncertain estimate of the states.

Assume a set of past frames (including keyframes) as well as a local map consisting of landmarks with sufficiently well-known 3D position is available at this point (see Section 5.2 for details). As a first stage of establishing correspondences, we perform a 3D–2D matching step. Given the current pose prediction, all landmarks that should be visible are considered for brute-force descriptor matching. Outliers are only rejected afterwards. This scheme may seem illogical to the reader who might intuitively want to apply the inverse order in the sense of a guided matching strategy; however, owing to the super-fast matching of binary descriptors, it would actually be more expensive to first look at image-space consistency. The outlier rejection consists of two steps: first of all, we use the uncertain pose predictions in order to perform a Mahalanobis test in image coordinates. Second, an absolute pose random sample consensus (RANSAC) provided in OpenGV (Kneip and Furgale, 2014) is applied.

Next, 2D–2D matching is performed in order to associate keypoints without 3D landmark correspondences. Again, we use brute-force matching first, followed by triangulation, in order to initialize landmark positions and as a first step to rejecting outlier pairings. Both stereo-triangulation across stereo image pairs (in the non-mono case) is performed, as well as between the current frame and any previous frame available. Only triangulations with sufficiently low depth uncertainty are labeled to be initialized, the rest will be treated as 2D measurements in subsequent matching. Finally, a relative RANSAC step (Kneip and Furgale, 2014) is performed between the current frame and the newest keyframe. The respective pose guess is furthermore used for bootstrapping in the very beginning.

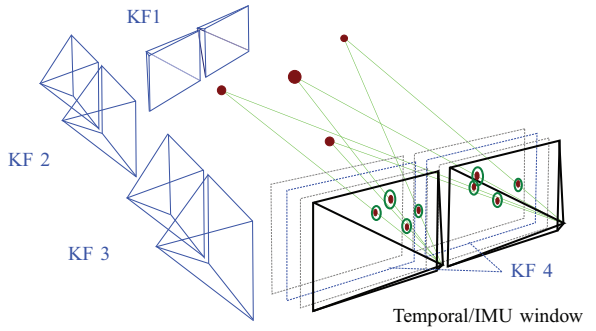

Fig. 6. Frames kept for matching and subsequent optimization in the stereo case: in this example, $M = 3$ keyframes and $S = 4$ most current frames are used.

Figure 5 illustrates a typical detection and matching result in the stereo case. Note the challenging illumination with overexposed sky due to facing towards the Sun.

5.2. Keyframe selection

For the subsequent optimization, a bounded set of camera frames is maintained, i.e. poses with associated image(s) taken at that time instant; all landmarks co-visible in these images are kept in the local map. As illustrated in Figure 6, we distinguish two kinds of frames: we introduce a temporal window of the S most recent frames including the current frame; and we use a number of M keyframes that may have been taken far in the past. For keyframe selection, we use a simple heuristic: if the hull of the projected and matched landmarks covers less than some percentage of the image (we use around 50%), or if the ratio of matched versus detected keypoints is small (below around 20%), the frame is inserted as a keyframe.

6. Keyframes and marginalization

In contrast to the vision-only case, it is not obvious how nonlinear temporal constraints from the IMU can reside in a bounded optimization window containing keyframes that may be arbitrarily far spaced in time. In the following, we first provide the mathematical foundations for marginalization, i.e. elimination of states in nonlinear optimization, and apply them to visual–inertial odometry.

6.1. Mathematical formulation of marginalization in nonlinear optimization

A Gauss–Newton system of equations is constructed from all of the error terms, Jacobians and information matrices, taking the form $\mathbf{H}\delta\boldsymbol{\chi} = \mathbf{b}$. Let us consider a set of states to be marginalized out, $\mathbf{x}_\mu$, the set of all states related to those by error terms, $\mathbf{x}_\lambda$, and the set of remaining states, $\mathbf{x}_\rho$. Due to conditional independence, we can simplify the marginalization step and only apply it to a sub-problem:

$$\begin{bmatrix} \mathbf{H}_{\mu\mu} & \mathbf{H}_{\mu\lambda_1} \\ \mathbf{H}_{\lambda_1\mu} & \mathbf{H}_{\lambda_1\lambda_1} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\chi}_\mu \\ \delta\boldsymbol{\chi}_\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{b}_\mu \\ \mathbf{b}_{\lambda_1} \end{bmatrix} \tag{21}$$

Application of the Schur complement operation yields

$$\mathbf{H}^*_{\lambda_1\lambda_1} := \mathbf{H}_{\lambda_1\lambda_1} - \mathbf{H}_{\lambda_1\mu}\mathbf{H}^{-1}_{\mu\mu}\mathbf{H}_{\mu\lambda_1} \tag{22a}$$

$$\mathbf{b}^*_{\lambda_1} := \mathbf{b}_{\lambda_1} - \mathbf{H}_{\lambda_1\mu}\mathbf{H}^{-1}_{\mu\mu}\mathbf{b}_\mu \tag{22b}$$

where $\mathbf{b}^*_{\lambda_1}$ and $\mathbf{H}^*_{\lambda_1\lambda_1}$ are nonlinear functions of $\mathbf{x}_\lambda$ and $\mathbf{x}_\mu$.

The equations in (22) describe a single step of marginalization. In our keyframe-based approach, we must apply the marginalization step repeatedly and incorporate the resulting information as a prior in our optimization while our state estimate continues to change. Hence, we fix the linearization point around $\mathbf{x}_0$, the value of $\mathbf{x}$ at the time of marginalization. The finite deviation $\Delta\boldsymbol{\chi} := \Phi^{-1}(\log(\overline{\mathbf{x}} \boxplus \mathbf{x}_0^{-1}))$ represents state updates that occur after marginalization, where $\overline{\mathbf{x}}$ is our current estimate for $\mathbf{x}$. In other words, $\mathbf{x}$ is composed as

$$\mathbf{x} = \exp(\Phi(\delta\boldsymbol{\chi})) \boxplus \underbrace{\exp(\Phi(\Delta\boldsymbol{\chi})) \boxplus \mathbf{x}_0}_{=\overline{\mathbf{x}}} \tag{23}$$

This generic formulation allows us to apply prior information on minimal coordinates to any of our state variables, including unit length quaternions. Introducing $\Delta\boldsymbol{\chi}$ allows the right-hand side to be approximated (to first order) as

$$\mathbf{b} + \left.\frac{\partial \mathbf{b}}{\partial \Delta\boldsymbol{\chi}}\right|_{\mathbf{x}_0} \Delta\boldsymbol{\chi} = \mathbf{b} - \mathbf{H}\Delta\boldsymbol{\chi} \tag{24}$$

Again using the partition into $\delta\boldsymbol{\chi}_\mu$ and $\delta\boldsymbol{\chi}_\lambda$, we can now write (24) as the right-hand side of the Gauss–Newton system (22b) as

$$\begin{bmatrix} \mathbf{b}_\mu \\ \mathbf{b}_{\lambda_1} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{\mu,0} \\ \mathbf{b}_{\lambda_1,0} \end{bmatrix} - \begin{bmatrix} \mathbf{H}_{\mu\mu} & \mathbf{H}_{\mu\lambda_1} \\ \mathbf{H}_{\lambda_1\mu} & \mathbf{H}_{\lambda_1\lambda_1} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\chi}_\mu \\ \Delta\boldsymbol{\chi}_\lambda \end{bmatrix} \tag{25}$$

In this form, i.e. plugging in (25), the right-hand side (22) becomes

$$\mathbf{b}^*_{\lambda_1} = \underbrace{\mathbf{b}_{\lambda_1,0} - \mathbf{H}_{\lambda_1\mu}\mathbf{H}^{-1}_{\mu\mu}\mathbf{b}_{\mu,0}}_{\mathbf{b}^*_{\lambda_1,0}} - \mathbf{H}^*_{\lambda_1\lambda_1}\Delta\boldsymbol{\chi}_{\lambda_1} \tag{26}$$

The marginalization procedure thus consists of applying (22a) and (26).

In the case where marginalized nodes comprise landmarks at infinity (or sufficiently close to infinity), or landmarks visible only in one camera from a single pose, the Hessian blocks associated with those landmarks will be (numerically) rank-deficient. We thus employ the pseudo-inverse $\mathbf{H}^+_{\mu\mu}$, which provides a solution for $\delta\boldsymbol{\chi}_\mu$ given $\delta\boldsymbol{\chi}_\lambda$ with a zero-component into null space direction.

The formulation described above introduces a fixed linearization point for both the states that are marginalized $\mathbf{x}_\mu$, as well as the remaining states $\mathbf{x}_\lambda$. This will also be used as

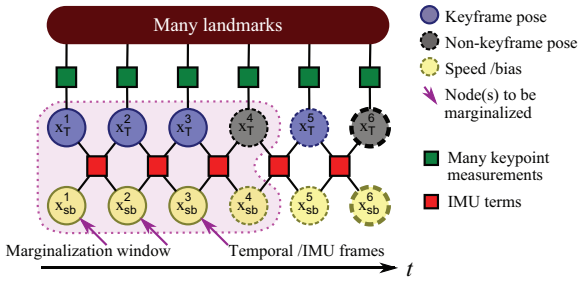

Fig. 7. Graph showing the initial marginalization on the first $M+1$ frames: speed and bias states outside the temporal window of size $S = 3$ are marginalized out.

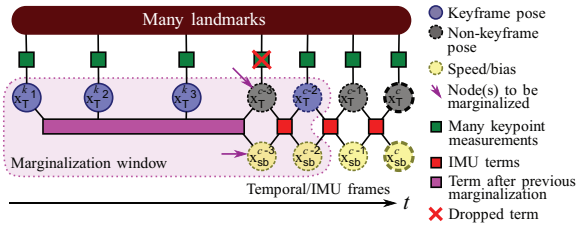

Fig. 8. Graph illustration with $M = 3$ keyframes and an IMU/temporal node size $S = 3$. A regular frame is slipping out of the temporal window. All corresponding keypoint measurements are dropped and the pose as well as speed and bias states are subsequently marginalized out.

a point of reference for *all future* linearizations of terms involving these states: this procedure is referred to as using "first estimate Jacobians" and was applied by Dong-Si and Mourikis (2011), with the aim of minimizing erroneous accumulation of information. After application of (22), we can remove the nonlinear terms consumed and add the marginalized $\mathbf{H}^*_{\lambda_1\lambda_1}$ and $\mathbf{b}^*_{\lambda_1}$ as summands to construct the overall Gauss–Newton system. The contribution to the chi-square error may be written as $\chi^2_{\lambda_1} = \mathbf{b}^{*T}_{\lambda_1}\mathbf{H}^{*+}_{\lambda_1\lambda_1}\mathbf{b}^*_{\lambda_1}$.

6.2. Marginalization applied to keyframe-based visual–inertial SLAM

The initially marginalized error term is constructed from the first $M + 1$ frames $\mathbf{x}^k_{\mathrm{T}}$, $k = 1, \ldots, M+1$ with respective speed and bias states as visualized graphically in Figure 7. The M first frames will all be interpreted as keyframes and the marginalization step consists of eliminating the corresponding speed and bias states. Note that before marginalization, we transform all error terms relating variables to be marginalized into one linear error term according to (25), which will persist and form a part of any subsequent marginalization step.

When a new frame $\mathbf{x}^c_{\mathrm{T}}$ (current frame, index c) is inserted into the optimization window, we apply a marginalization operation. In the case where the oldest frame in the temporal window ($\mathbf{x}^{c-S}_{\mathrm{T}}$) is not a keyframe, we will drop all of its landmark measurements and then marginalize it out together with the oldest speed and bias states. In other words, all states are marginalized out, but no landmarks. Figure 8 illustrates this process. Dropping landmark measurements is suboptimal; however, it keeps the problem sparse for efficient solutions. In fact, visual SLAM with keyframes successfully proceeds analogously: it drops entire frames with their landmark measurements.

In the case of $\mathbf{x}^{c-S}_{\mathrm{T}}$ being a keyframe, the information loss of simply dropping all keypoint measurements would be more significant: all relative pose information between the oldest two keyframes encoded in the common landmark observations would be lost. Therefore, we additionally marginalize out the landmarks that are visible in $\mathbf{x}^{k_1}_{\mathrm{T}}$ but not in the most recent keyframe or newer frames. This means,

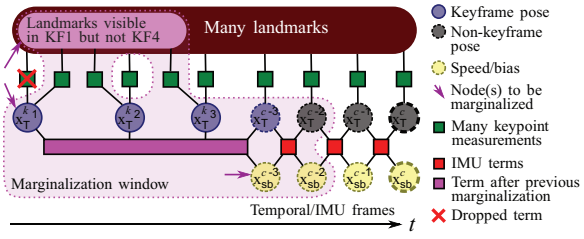

Fig. 9. Graph for marginalization of $\mathbf{x}^{c-3}_{\mathrm{T}}$ being a keyframe: the first (oldest) keyframe ($\mathbf{x}^{k_1}_{\mathrm{T}}$) will be marginalized out. Hereby, landmarks visible exclusively in $\mathbf{x}^{k_1}_{\mathrm{T}}$ to $\mathbf{x}^{k_{M-1}}_{\mathrm{T}}$ will be marginalized as well.

respective landmark observations in the keyframes $k_1, \ldots k_{M-1}$ are included in the linearized error term prior to landmark marginalization. Figure 9 depicts this procedure graphically.

The sparsity of the problem is again preserved; we show the sparsity of the Hessian matrix in Figure 10, along with further explanations on how measurements are dropped and landmarks may be marginalized.

The above explanations did not include extrinsics calibration nodes. The framework, owing to its generic nature, is nevertheless extended to handle this case in a straightforward manner: in fact, the extrinsics poses will also be added to the linear term, as soon as landmarks are marginalized out. In Section 7, we present results on online estimation of temporally static extrinsics $\mathbf{x}_{\mathrm{C}_i}$, $i \in \{1, 2\}$; treating them as states, i.e. instances inserted at every frame, is equally possible. A temporal relative pose error has to be inserted in this case to model allowed changes of the extrinsics over time. Furthermore, marginalization of extrinsics nodes along with poses will be required in this case.

6.3. Priors and fixation of variables

As described previously, our framework allows for a purely relative representation of information that applies to the optimization window. This formulation constitutes a fundamental advantage over classical filtering approaches, where uncertainty is kept track of in an absolute manner, i.e. in a

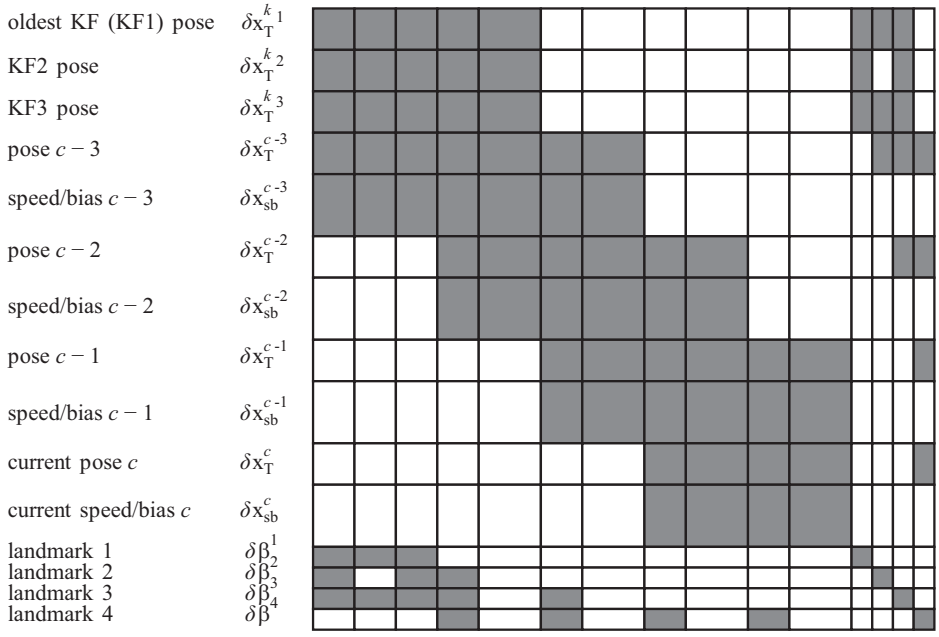

Fig. 10. Example sparsity pattern of the Hessian matrix (gray signifies a non-zero block) for a simple case with only four landmarks. In case of marginalization of frame $c-3$, the observations of landmarks 2,3, and 4 in frame $c-3$ would be removed prior to marginalization, in order to prevent fill-in. In case of marginalization of the oldest keyframe KF1, the proposed strategy would marginalize out landmark 1, remove the observations of landmarks 2 and 3 in KF1, and leave landmark 4, since it is not observed in KF1.

global frame of reference: with the absolute formulation, naturally, uncertainty will grow and increasingly incorrectly be represented through some form of linear propagation, leading to inconsistencies if not specifically addressed.

Furthermore, a filter will always need priors for all states when initializing, where they might be completely unknown and potentially bias the estimate. Our presented framework does conceptually not need any priors. For more robust initialization particularly of the monocular version, however, we actually apply (rather weak) zero-mean uncorrelated priors to speed and biases. For speed we use a standard deviation of 3 m/s, which is tailored to the setups as presented in the results. For gyro bias, we applied a prior with standard deviation 0.1 rad/s and for accelerometer bias 0.2 m/s^2, which relates to the IMU parameters described in Section 7.1.1.

Inherently, the vision-only problem has six DoFs that are unobservable and need to be held fixed during optimization, i.e. the absolute pose. The combined visual–inertial problem has only four unobservable DoFs, since gravity renders two rotational DoF observable.

In contrast to our previously published results (Leutenegger et al., 2013), we forgo fixation of absolute yaw and position: underlying optimization algorithms such as Levenberg–Marquardt will automatically cater to not taking steps along unobservable directions. Forced fixation of yaw may introduce errors, in case the orientation is not very accurately estimated.

Due to numeric noise, positive-semidefiniteness of the left-hand side linearized subpart $\mathbf{H}^*_{\lambda_1\lambda_1}$ has to be enforced at all times. To ensure this, we apply an Eigen-decomposition $\mathbf{H}^*_{\lambda_1\lambda_1} = \mathbf{U}\Lambda\mathbf{U}^{\mathrm{T}}$ before optimization, and reconstruct $\mathbf{H}^*_{\lambda_1\lambda_1}$ as $\mathbf{H}^*_{\lambda_1\lambda_1}{}' = \mathbf{U}\Lambda'\mathbf{U}^{\mathrm{T}}$, where $\mathbf{\Lambda}'$ is obtained from $\mathbf{\Lambda}$ by setting all Eigenvalues below a threshold to zero.

7. Results

Throughout the literature, a plethora of motion tracking algorithms has been suggested; how they perform in relation to each other, however, is often unclear, since results are typically shown on individual datasets with differing motion characteristics as well as sensor qualities. In order to make a strong argument for our presented work, we will thus compare it with a state-of-the-art visual–inertial stochastic cloning sliding-window filter which follows the MSCKF derivation of Mourikis et al. (2009).

7.1. Evaluation setup

In the following, we provide a short overview of the hardware and settings used for dataset acquisition, as well as of the hardware and algorithms used for evaluation.

7.1.1. Sensor unit overview. The custom-built visual–inertial sensor is described in detail by Nikolic et al. (2014). In essence, the assembly as shown in Figure 11 consists of an ADIS16448 MEMS IMU and two embedded WVGA monochrome cameras with an 11 cm baseline that are all rigidly connected by an aluminum frame. An FPGA board

Table 1. IMU characteristics.

Rate gyros			Accelerometers		
σ_g	1.2×10^{-3}	$\mathrm{rad}/(\mathrm{s}\sqrt{\mathrm{Hz}})$	σ_a	8.0×10^{-3}	$\mathrm{m}/(\mathrm{s}^2\sqrt{\mathrm{Hz}})$
σ_{b_g}	2.0×10^{-5}	$\mathrm{rad}/(\mathrm{s}^2\sqrt{\mathrm{Hz}})$	σ_{b_a}	5.5×10^{-5}	$\mathrm{m}/(\mathrm{s}^3\sqrt{\mathrm{Hz}})$
			τ	∞	s

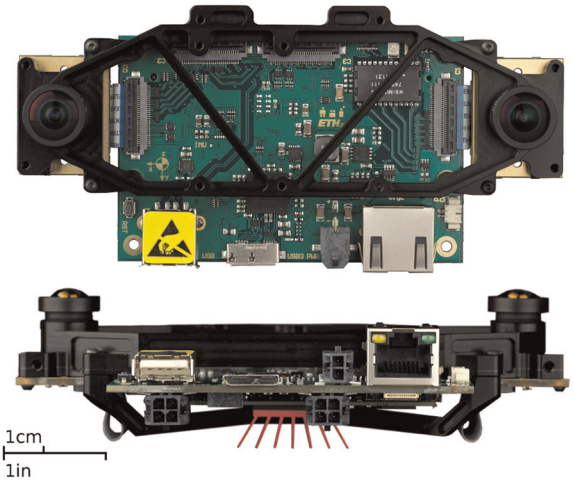

Fig. 11. Visual–inertial sensor front and side view. Stereo imagery is hardware synchronized with the IMU measurements and transmitted to a host computer via gigabit Ethernet.

performs hardware synchronization between imagery and IMU up to the level of pre-triggering the cameras according to the variable shutter opening times. Furthermore, the FPGA may perform keypoint detection, in order to save CPU usage for subsequent algorithms. The data is streamed to a host computer via Gigabit Ethernet. The datasets used in this work were collected at an IMU rate of 800 Hz, while the camera frame rate was set to 20 Hz (although the hardware would allow up to 60 Hz).

7.1.2. Sensor characteristics. We have taken the IMU noise parameters from the ADIS16448 datasheet (available at http://www.analog.com/en/mems-sensors/mems-inertial-measurement-units/adis16448/products/product.html as of March 2014) and verified them in stand-still. In order to account for unmodeled and dynamic effects, slightly more conservative numbers as listed in Table 1 were used.

Concerning image keypoints, we applied a detection standard deviation of 0.8 pixels. Note that the keypoints were extracted on the highest-resolution image of the pyramid only. Again, this is slightly higher than what error statistics of our sub-pixel-resolution Harris corner detector would suggest.

An intrinsics and extrinsics calibration (distortion coefficients and $\boldsymbol{T}_{SC_i}$) was obtained using the method described by Furgale et al. (2013).

7.1.3. Quantitative evaluation procedures. Defining the system boundaries of a specific algorithm along with its inputs and outputs poses some trade-offs. We chose to feed all algorithms with the same correspondences (i.e. keypoint measurements with landmark IDs per image) as they were generated by our stereo algorithm. Each algorithm evaluated from there was left with the freedom to apply its own outlier rejection and landmark triangulation. Obviously, all algorithms were provided with the same IMU measurements. To ensure fairness, we furthermore apply the same keypoint detection uncertainty as well as IMU noise densities, and gravity acceleration across all algorithms. Note that all parameters were left unchanged throughout all datasets and for all algorithms, including keypoint detection and matching thresholds.

We adopt the evaluation scheme of Geiger et al. (2012): for many starting times, the ground truth and estimated trajectories are aligned and the error is evaluated for increasing distances traveled from there.

Consider the ground-truth trajectory $\boldsymbol{T}_{GV}^{p}$ and estimated trajectory $\bar{\boldsymbol{T}}_{WS}^{p}$ both resampled at the same rate indexed by p, where $\underrightarrow{\mathcal{F}}_{V}$ denotes the body frame of the ground truth trajectory. Let d^p be the (scalar) distance traveled since start-up; in order to obtain a statistical characterization we choose many starting pose indices p_s such that they are spaced by a specific traveled distance. Relative to those starting poses, we define the error transformation as

$$\Delta\boldsymbol{T}(\Delta d) = \bar{\boldsymbol{T}}_{WS}^{p}\boldsymbol{T}_{SV}\boldsymbol{T}_{GV}^{p\,-1}\boldsymbol{T}_{GW}^{p_s}, \forall p > p_s \qquad (27)$$

where $\Delta d = d^p - d^{p_s}$. The many errors $\Delta\boldsymbol{T}(\Delta d)$ can now be accumulated in bins of distance traveled, in order to obtain error statistics.

At this point, we have to distinguish between the availability of 6D ground truth from the indoor Vicon motion tracking system (Vicon motion tracking system, see http://www.vicon.com/ as of March 2014) or 3D (DGPS) obtained using a Leica Viva GS14 (Leica Viva GS14 GNSS recorder http://www.leica-geosystems.com/en/Leica-Viva-GS14_102200.htm as of March 2014) ground truth.

In the 6D case, we first have to estimate the transformation between the tracked body frame $\underrightarrow{\mathcal{F}}_{V}$ and the estimation body frame $\underrightarrow{\mathcal{F}}_{S}$. The alignment of estimator world frame $\underrightarrow{\mathcal{F}}_{W}$ and ground-truth world frame $\underrightarrow{\mathcal{F}}_{G}$ at starting time index p_s becomes trivially $\boldsymbol{T}_{GW}^{p_s} = \boldsymbol{T}_{GV}^{p_s}\boldsymbol{T}_{SV}^{-1}\bar{\boldsymbol{T}}_{\boldsymbol{WS}}^{\boldsymbol{p_s}\,-1}$.

Table 2. Dataset characteristics.

Name	Length	Duration	Max. Speed	Ground Truth
Vicon Loops	1200 m	14 min	2.0 m/s	Vicon 6D @200 Hz
Bicycle Trajectory	7940 m	23 min	13.1 m/s	DGPS 3D @1 Hz
Handheld around ETH Main Building	620 m	6:40 min	2.2 m/s	DGPS 3D @1 Hz

In the 3D ground-truth case, however, we have to set $\mathbf{C}_{GV} = \mathbf{I}$. We furthermore neglect the offset between GPS antenna and IMU center (this is in the order of centimeters) and set $\boldsymbol{T}_{SV}^{-1} = \mathbf{I}$. Now the alignment of the world frames $\boldsymbol{T}_{GW}^{p_s}$ is solved for as an SVD-based trajectory alignment, where the (small but large enough) segment used in this process is obviously discarded for evaluation.

7.2. Evaluation on complementary datasets

In the following, we will present evaluation results on three datasets. In order to cover different conditions, care was taken to record datasets with different lengths, distances to structure, speeds, dynamics, as well as with differences in illumination and number of moving objects. The main characteristics are summarized in Table 2. We compare both our monocular version (*aslam-mono*) and the stereo variant (*aslam*) to the (monocular) sliding window filter reference implementation (*msckf-mono*). Our algorithm used $M = 7$ keyframes and $S = 3$ most current frames in all datasets, while the sliding window filter was set up to maintain five pose clones.

Note that the following trajectory reconstructions do not include sigma bounds; this is related to the fact that the presented framework only uses relative information and thus global uncertainty is not represented. We do not use priors or a global fixation on the unobservable subspace (e.g. very first position and yaw angle). While we consider this formulation a major advantage of our approach, it comes at the cost of not being able to simply plot uncertainty bounds. In fact, the global uncertainty could be recovered by linking the relative information to a pose graph containing *all* keyframe poses ever recorded; the first position and yaw angle would be fixed or given a prior. Owing to uncertainty growth over time, the current global pose covariance, however, loses its meaning, since linear error propagation trough a transformation chain will be increasingly inaccurate.

7.2.1. Vicon loops. A trajectory was recorded with the handheld sensor inside our Vicon room. Consequently, the path is very much limited in spatial extent, while only close structure is observed. Full 6D ground truth is available from external motion tracking at 200 Hz. The sequence lasts almost 14 minutes, walking mostly in circles no faster than 2.0 m/s. We show the overhead plot, altitude profile along with absolute errors as a function of distance traveled in Figure 12.

Note that all algorithms achieve below 0.1% of median position error per distance traveled at the end of the 1200 m long path; this is, however, not only caused by the high accuracy of the algorithms. In fact, yaw drift, which is clearly present, does not become manifest as much in position error as in the other datasets that cover larger distances. The differences between the algorithms do not show extreme differences, but some subtleties may nonetheless be identified: while all manage to estimate the World z-direction (aligned with acceleration due to gravity), the computationally more expensive algorithms proposed in this work expectedly show a slightly better performance in terms of yaw drift. We furthermore provide the plots of both gyro and accelerometer biases in Figure 13. Note that despite their different natures, all algorithms converge to tracking very similar values.

7.2.2. Bicycle trajectory. The sensor was mounted onto a helmet and worn for a bicycle ride of 7.9 km from ETH Hönggerberg into the city of Zurich and back to the starting point. Figure 14 illustrates the setup. Speeds up to 13 m/s were reached during the 23-minute-long course. Post-processed DGPS ground truth is available at 1 Hz, and all measurements with a position uncertainty beyond 1 m were discarded. Figure 15 displays reconstructed trajectories as compared to ground-truth and reports the statistics on the position error normalized by distance traveled.

As expected, the stereo version shows a notably better performance than the monocular one. Both outperform the reference implementation of the MSCKF, which clearly suffers from more yaw drift. It is furthermore worth mentioning that *aslam-mono* and *aslam* accumulate less drift in altitude, where a clear advantage of the stereo algorithm becomes visible.

7.2.3. Handheld around ETH main building. This second outdoor loop was recorded with the handheld sensor while walking around the main building of central ETH (no faster than 2.2 m/s). The path length amounts to 620 m. The imagery is characterized by varying depth of the observed structure, and includes some pedestrians walking by. Figure 16 summarizes the results.

Again, both our approaches outperform the reference implementation of the MSCKF. Qualitatively, the yaw error seems to contribute the least to the position error; rather the position drift appears to originate from (locally) badly estimated scale. Interestingly, the stereo version of our algorithm seems to perform slightly worse in this respect.

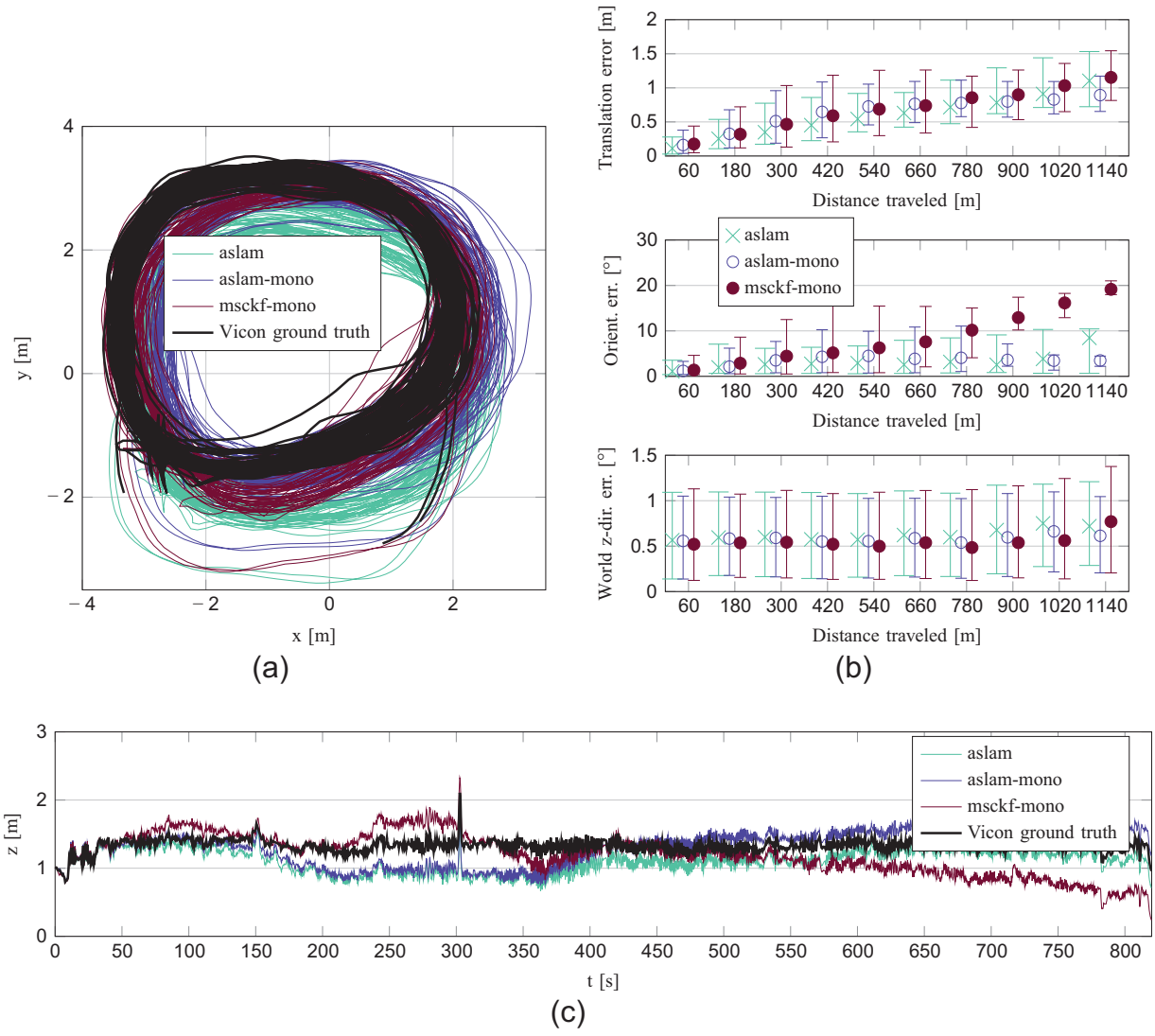

Fig. 12. Vicon dataset evaluation: while differences in position errors between the different algorithms and variants are not very significant, yaw drift of the *msckf* is clearly higher. (a) Overhead plot of the vicon dataset. (b) Error statistics in terms of median, 5th, and 95th percentiles: norm of position error (top), norm of error axis angle vector (middle), and angle between ground-truth down axis and estimator down axes (bottom). (c) Altitude profiles, i.e. z -components of estimator outputs represented in $\overset{\mathcal{F}}{\rightarrow} G$.

We suspect small errors in the stereo calibration to cause this behavior, concretely a slight mismatch of relative camera orientation. This issue is further investigated in Section 7.3.1.

7.3. Parameter studies

With focus on the proposed algorithm, studies are provided that investigate the sensitivity on the performance with respect to a selected set of parameters.

7.3.1. Online extrinsics calibration. In the following experiment, we assess the performance of our online IMU to camera extrinsics estimation scheme. We assume a starting point of calibrated intrinsics, for which off-the-shelf tools exist; and we furthermore take rough extrinsics as available in a straightforward manner, e.g. from a Computer Aided Design (CAD) software. For the problem to be best constrained, we run the stereo algorithm.

In order to obtain a well-defined optimization problem, we apply a weak prior to all extrinsic translations (10 mm standard deviation) as well as orientation (0.6° standard deviation).

Using the ETH Main Building dataset as introduced above, Figure 17 displays a respective comparison in terms of estimation accuracy of pre-calibration, online-calibration and post-calibration to the result as shown above with the original calibration. Remarkably, the very rough extrinsics guess generates mostly scale mismatch, while the orientation seems consistent. In fact, the scale is not simply wrong due to incorrect baseline setting: since the baseline is

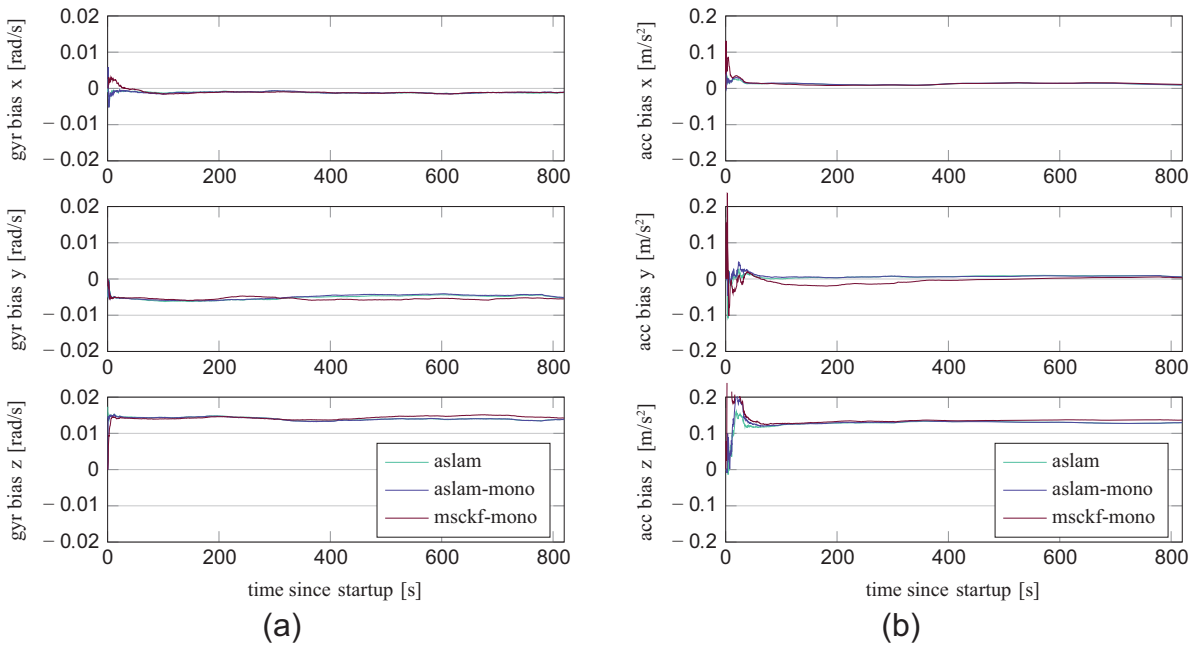

Fig. 13. Evolution of bias estimates by the different algorithms: despite the different characteristics of the algorithms, they converge to and track similar values. (a) Gyro bias estimates. (b) Accelerometer bias estimates.

Fig. 14. Simon Lynen ready for dataset collection on a bicycle at ETH with sensor and GNSS ground-truth recorder mounted to a helmet.

known and set to a much higher accuracy than the 10% scale error. Interestingly, the estimates during online calibration are significantly more accurate than with fixed original calibration. In fact, the scale mismatch is completely removed. Moreover, taking the frozen final online estimates and re-running the process results in no significant change as compared with the online estimation, suggesting that online extrinsics calibration may be safely left switched on, at least in the stereo case. In such a case, however, noise would need to be injected into the estimation process, in order to allow for the extrinsics to (slowly) drift.

Figure 18 addresses the remaining question of whether or not the extrinsics converge, and if the respective estimates correspond to the original calibration.

Clearly, the estimates of the IMU to camera orientations converge fast to a stable estimate, while the camera positions with respect to the IMU that are less well observable are subjected to more mobility.

7.3.2. Influence of keyframe and keypoint numbers. We claim that the proposed algorithm offers scalability in terms of tailoring it according to the trade-off between accuracy and processing constraints. In this context, we analyze the influence of two main parameters on the performance: on the one hand, we can play with the number of pose variables to be estimated, and on the other hand we have the choice of average number of landmark observations per image by adjusting the keypoint detection threshold.

Taking the ETH Main Building dataset and running the mono-version of our algorithm, we show the quantitative results for different keyframe number M settings in Figure 19. In the same comparison, we furthermore varied the number of frames connected by nonlinear IMU terms S

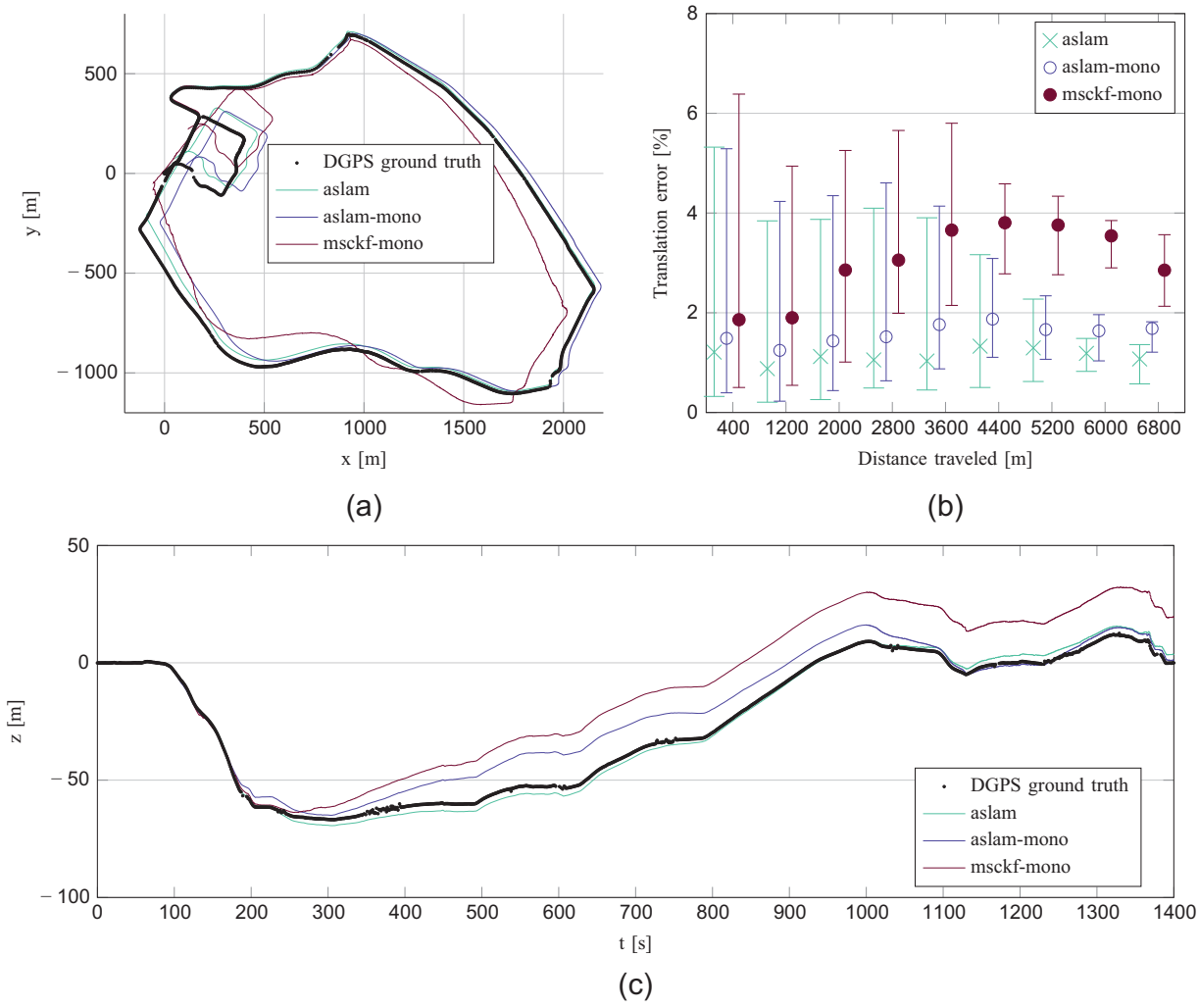

Fig. 15. Trajectories and evaluation of the Bicycle Trajectory dataset: the filtering approach *msckf-mono* accumulates the largest yaw error that becomes manifest also in position error. As expected, our stereo variant performs the best, which is also apparent in the altitude evolution. (a) Overhead plot of the reconstructed bicycle ride trajectories. (b) Relative position error statistics: median, 5th, and 95th percentiles. (c) Altitude profile of the bicycle ride.

and provide the full-batch solution as a gold standard reference. Note that the full-batch problem was initialized with the original version of our algorithm and run to convergence.

At the low end of the frame number ($M = 4$), we see a clear performance drawback, whereas large numbers of keyframes $M = 12$ do not seem to increase accuracy. Another interesting finding is that increasing S to account for more nonlinear error terms does not become manifest in less error. Also note that the full-batch optimization does not increase the overall accuracy of the solution, indicating that the approximations in terms of linearization, marginalization, as well as measurement dropping as described here are forming a suite of reasonable choices.

Note that in the overall complexity of the algorithm, the number of keyframes M contributes with $\mathcal{O}(M^3)$ when it comes to solving the respective dense part of the linear system of equations.

Finally, we also investigate the influence of the keypoint detection threshold u that directly affects keypoint density in image space. Figure 20 summarizes the respective quantitative results, again processing the ETH Main Building dataset with the monocular version of our algorithm.

Interestingly, all versions perform similarly on shorter ranges, despite the large variety in average keypoints per image, i.e. 45.3, 110.3, and 239.2. A slight trend suggesting that more keypoints result in better pose estimates is only visible for longer traveled distances. Note that increasing the detection threshold inevitably not only decreases execution time, but also comes at the expense of environment representation richness in terms of landmark density.

8. Conclusion

We have introduced a framework of tightly coupled fusion of inertial measurements and image keypoints in a

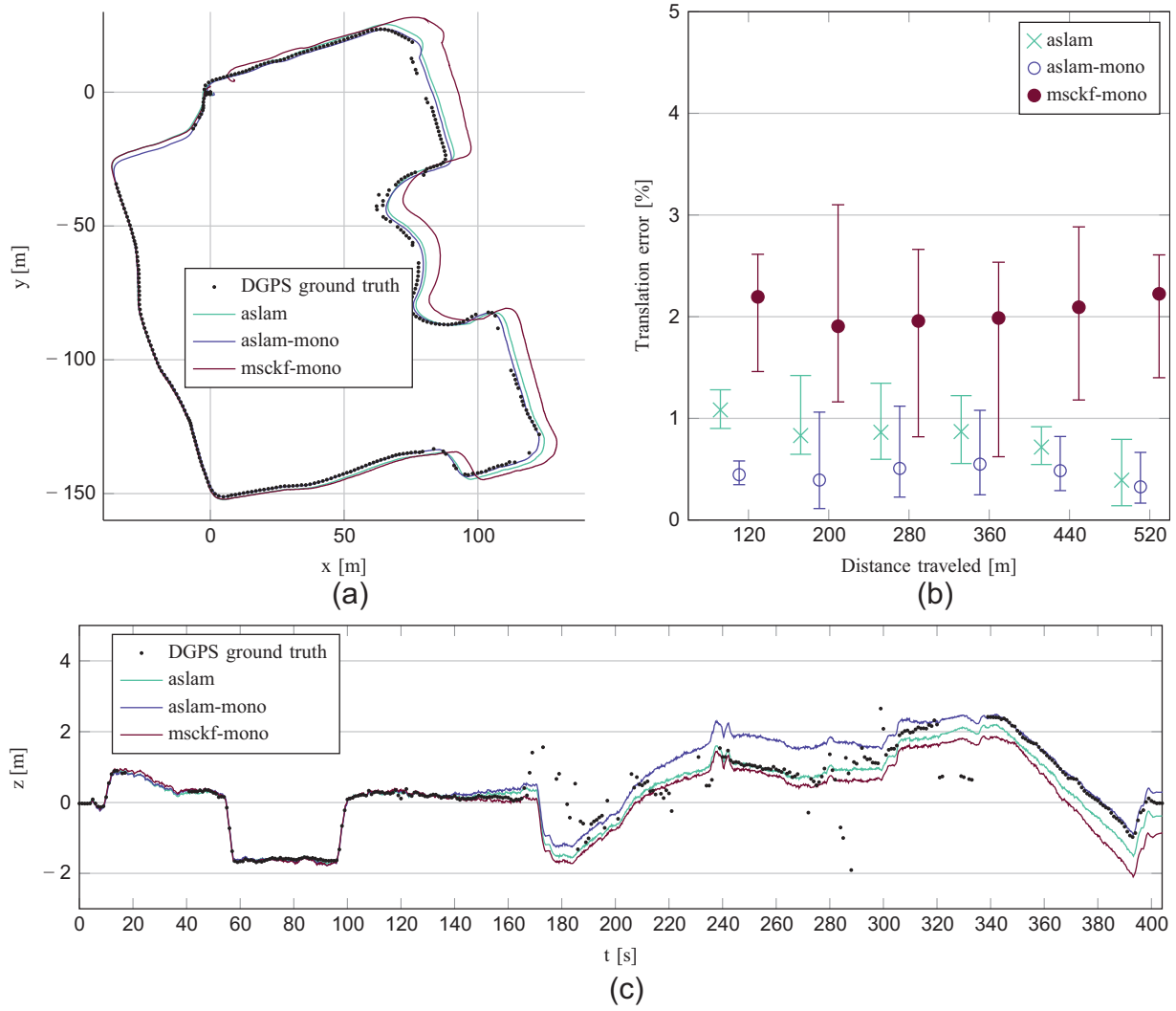

Fig. 16. Evaluation results for the ETH Main Building dataset: interestingly, the stereo version of our algorithm is outperformed by the mono variant. The cause is further investigated in Section 7.3.1. (a) Overhead plot of the reconstructed trajectories. (b) Relative position error statistics: median, 5th and 95th percentiles. (c) Altitude profile.

nonlinear optimization problem that applies linearization and marginalization in order to achieve keyframing. As an output, we obtain poses, velocities, and IMU biases as a time series, as well as a 3D map of sparse landmarks. The proposed algorithm is bounded in complexity, as the optimization includes a fixed number of poses. The keyframing, in contrast to a fixed-lag smoother, allows for arbitrary temporal extent of estimated camera poses jointly observing landmarks. As a result, our framework achieves high accuracy, while still being able to operate in real-time. We showed extensive evaluation of both a stereo and a mono version of the proposed algorithm on complementary datasets with varied type of motion, lighting conditions, and distance to structure. In this respect, we made the effort to compare our results with the output of a state-of-the-art visual–inertial sliding-window filter, which follows the MSCKF algorithm and is fed with the same IMU data and keypoints with landmark associations. While admittedly being computationally more demanding, our approach consistently outperforms the filter. In further studies we showed how online calibration of camera extrinsics can be incorporated into our framework: results on the stereo version indicate how slight miscalibration can become manifest in scale error; online calibration, even starting from a very rough initial guess, removes this effect. Finally, we also address scalability of the proposed method in the sense of tailoring to hardware characteristics, and how the setting of number of frames as well as detected keypoints affect accuracy. Interestingly, employing larger numbers of keyframes, such as 12, does not show a significant advantage over the standard setting of 7, at least in exploratory motion mode. Furthermore, we do not observe a dramatic performance decrease when reducing average numbers of keypoints per image from 240 to 45.

The way is paved for deployment of our algorithm on various robotic platforms such as unmanned aerial systems.

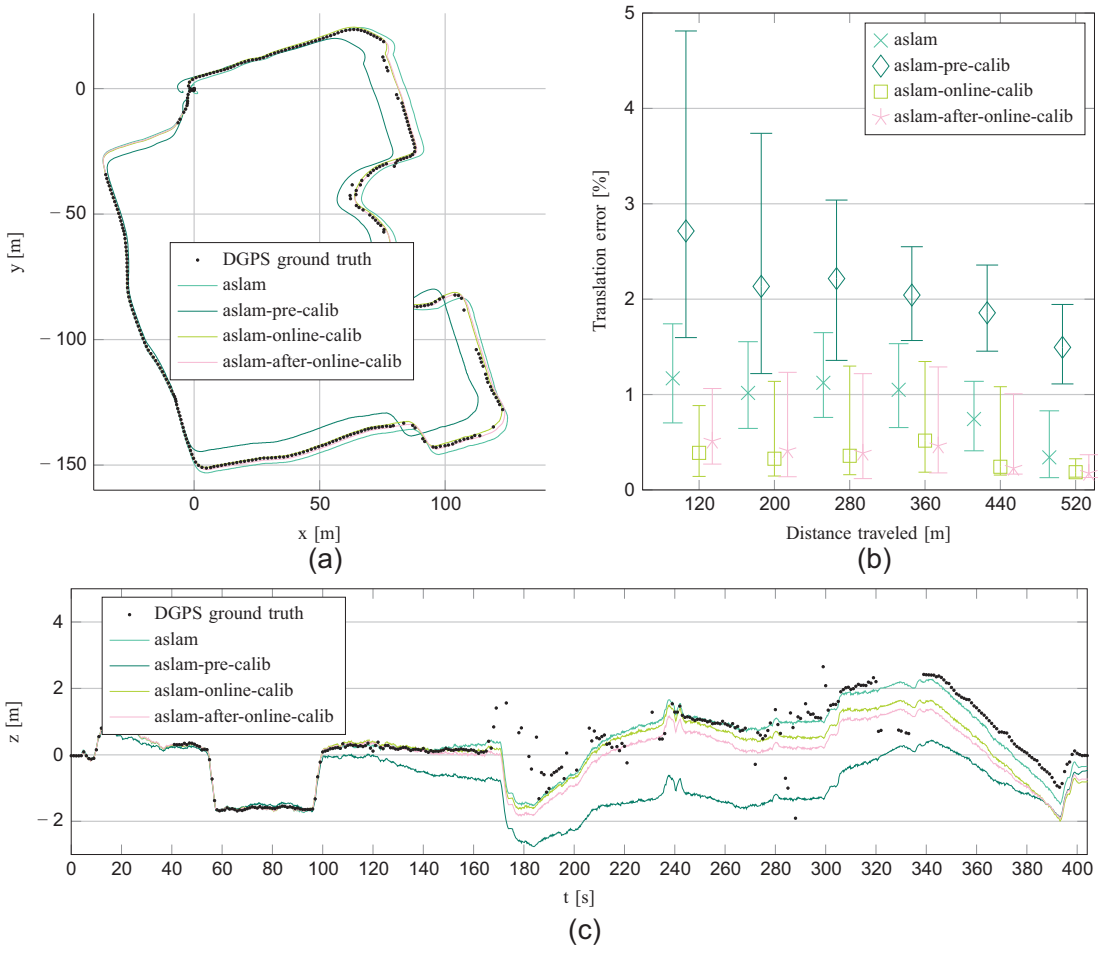

Fig. 17. Evaluation results for the extrinsics calibration. Compared to the original estimate (*aslam*), a reconstruction with fixed roughly aligned extrinsics (*aslam-pre-calib*) yields expectedly rather poor results. Estimates during online calibration (*aslam-online-calib*) that use the rough alignment as starting guess manages to even outperform the original result. Freezing the online estimates to the final values and re-running the process (*aslam-after-online-calib*) results in equivalent performance as with online calibration turned on. (a) Overhead plot of the extrinsics calibration. (b) Relative error statistics of the extrinsics calibration: median, 5th and 95th percentiles. (c) Altitude profile.

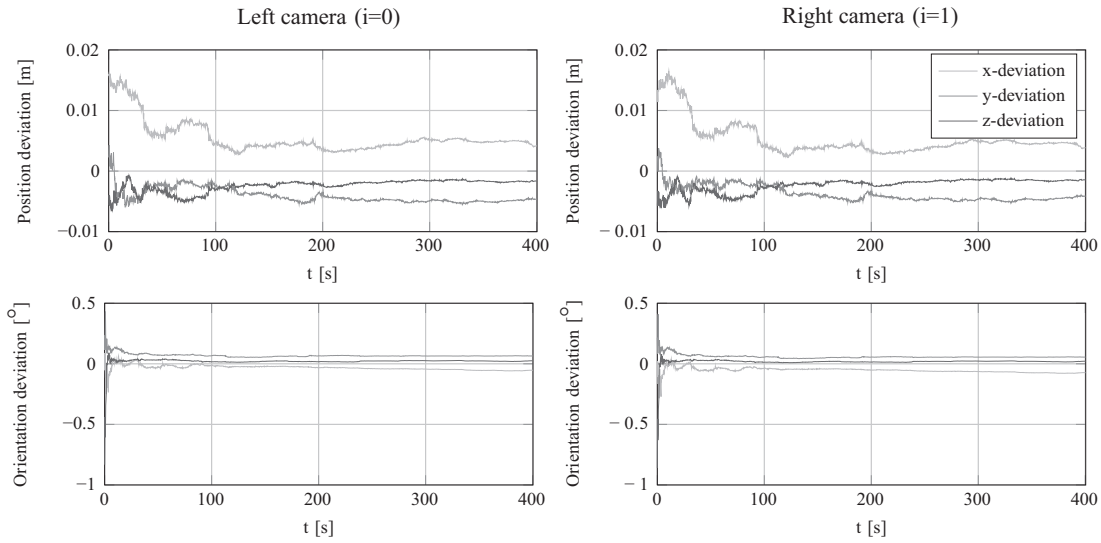

Fig. 18. Position differences of online calibrated extrinsics translation with respect to original (top) and axis-angle difference of online calibrated extrinsics orientation with respect to original (bottom) for left and right cameras.

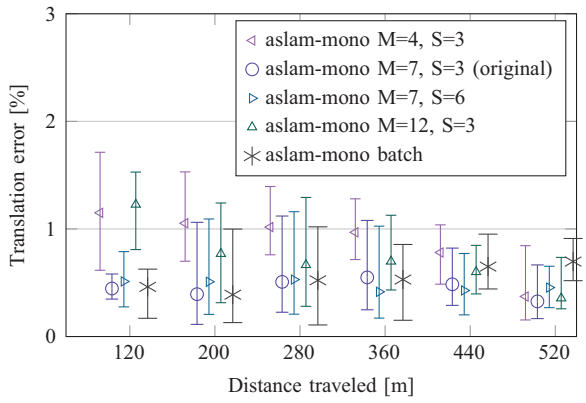

Fig. 19. Comparison of different frame number (N keyframes and S IMU frames) settings with respect to the full batch solution (initialized with the result from $N = 7$, $S = 3$).

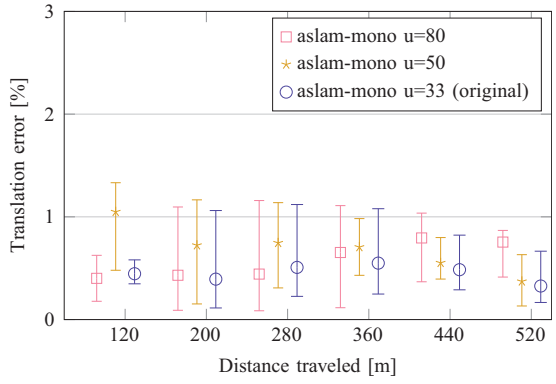

Fig. 20. Comparison of different keypoint detection thresholds $u \in \{80, 50, 33\}$ settings. The corresponding mean number of keypoints per image are 45.3, 110.3, and 239.2 in this dataset.

In this respect, we are planning to release our proposed framework as an open-source software package. Moreover, we will explore inclusion of other, platform-specific sensor feeds, such as wheel odometry, GPS, magnetometer or pressure measurements, with the aim of increasing accuracy and robustness of the estimation process, both primary requirements for successful deployment of robots in challenging real environments.

Acknowledgements

The authors would like to thank Janosch Nikolic, Pascal Gohl, Michael Burri, and Joern Rehder from ASL/ETH for their invaluable support with hardware and dataset recording and calibration efforts. Finally, special thanks go to Vincent Rabaud and Kurt Konolige from Willow Garage (at the time) for their valuable inputs.

Funding

The research leading to these results has received funding from the European Commission's Seventh Framework Programme (FP7/2007–2013; grant agreement numbers 285417 (ICARUS), 600958 (SHERPA) and 269916 (V-charge)). This work was also partly sponsored by Willow Garage.

References

Agarwal S, Mierle K, et al. (2010) Ceres Solver. Available at: http://ceres-solver.org (accessed 27 November 2014).

Barfoot T, Forbes JR and Furgale PT (2011) Pose estimation using linearized rotations and quaternion algebra. *Acta Astronautica* 68(1–2): 101–112.

Bayard DS and Brugarolas PB (2005) An estimation algorithm for vision-based exploration of small bodies in space. In: *Proceedings of the 2005 American control conference*. IEEE, pp. 4589–4595.

Bryson M, Johnson-Roberson M and Sukkarieh S (2009) Airborne smoothing and mapping using vision and inertial sensors. In: *IEEE international conference on robotics and automation, 2009 (ICRA'09)*. IEEE, pp. 2037–2042.

Chai L, Hoff WA and Vincent T (2002) Three-dimensional motion and structure estimation using inertial sensors and computer vision for augmented reality. *Presence: Teleoperators and Virtual Environments* 11(5): 474–492.

Davison A (2003) Real-time simultaneous localisation and mapping with a single camera. In: *Proceedings of the ninth IEEE international conference on computer vision*, 2003, vol. 2, pp. 1403–1410.

Dong-Si TC and Mourikis AI (2011) Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis. In: *IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 5655–5662.

Dong-Si TC and Mourikis AI (2012) Estimator initialization in vision-aided inertial navigation with unknown camera-imu calibration. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 1064–1071.

Ebcin S and Veth M (2007) Tightly-coupled image-aided inertial navigation using the unscented kalman filter. Technical report, DTIC Document.

Furgale P (2011) *Extensions to the Visual Odometry Pipeline for the Exploration of Planetary Surfaces*. PhD Thesis, University of Toronto, Institute for Aerospace Studies.

Furgale P, Rehder J and Siegwart R (2013) Unified temporal and spatial calibration for multi-sensor systems. In: *IEEE/RSJ international conference on robots and systems (IROS)*. pp. 1280–1286.

Geiger A, Lenz P and Urtasun R (2012) Are we ready for autonomous driving? The KITTI vision benchmark suite. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Harris C and Stephens M (1988) A combined corner and edge detector. In: *Proceedings of the 4th Alvey vision conference*, pp. 147–151.

Hesch JA, Kottas DG, Bowman SL and Roumeliotis SI (2012a) *Observability-constrained vision-aided inertial navigation*. Technical Report 1, University of Minnesota, Department of Computer Science and Engineering, MARS Lab.

Hesch JA, Kottas DG, Bowman SL and Roumeliotis SI (2012b) Towards consistent vision-aided inertial navigation. In: *Proceedings of the international workshop on the algorithmic foundations of robotics (WAFR)*.

Hesch JA, Kottas DG, Bowman SL and Roumeliotis SI (2013) Towards consistent vision-aided inertial navigation. In: *Algorithmic Foundations of Robotics X*. Springer, pp. 559–574.

Huang G, Kaess M and Leonard JJ (2013) Towards consistent visual-inertial navigation. *Technical Report* Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

Huang GP, Mourikis AI and Roumeliotis SI (2009) A first-estimates Jacobian EKF for improving SLAM consistency. In: *Experimental Robotics*. Springer, pp. 373–382.

Huang GP, Mourikis AI and Roumeliotis SI (2011) An observability-constrained sliding window filter for SLAM. In: *2011 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 65–72.

Indelman V, Williams S, Kaess M and Dellaert F (2012) Factor graph based incremental smoothing in inertial navigation systems. In: *International conference on information fusion (FUSION)*.

Jia C and Evans BL (2012) Probabilistic 3-D motion estimation for rolling shutter video rectification from visual and inertial measurements. In: *MMSP*, pp. 203–208.

Jones ES and Soatto S (2011) Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research* 30(4): 407–430.

Jung SH and Taylor CJ (2001) Camera trajectory estimation using inertial sensor measurements and structure from motion results. In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition (CVPR 2001)*, volume 2. IEEE, pp. II–732.

Kaess M, Johannsson H, Roberts R, Ila V, Leonard JJ and Dellaert F (2012) ISAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research* 31(2): 216–235.

Kelly J and Sukhatme GS (2011) Visual–inertial sensor fusion: localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research* 30(1): 56–79.

Kim J and Sukkarieh S (2007) Real-time implementation of airborne inertial-SLAM. *Robotics and Autonomous Systems* 55(1): 62–71.

Klein G and Murray D (2007) Parallel tracking and mapping for small AR workspaces. In: *6th IEEE and ACM international symposium on mixed and augmented reality, 2007 (ISMAR 2007)*. IEEE, pp. 225–234.

Kneip L and Furgale PT (2014) OpenGV: A unified and generalized approach to real-time calibrated geometric vision. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.

Konolige K, Agrawal M and Sola J (2011) Large-scale visual odometry for rough terrain. In: *Robotics Research*. Springer, pp. 201–212.

Kottas DG, Hesch JA, Bowman SL and Roumeliotis SI (2012) On the consistency of vision-aided inertial navigation. In: *Proceedings of the international symposium on experimental robotics (ISER)*.

Leutenegger S, Chli M and Siegwart R (2011) BRISK: Binary robust invariant scalable keypoints. In: *Proceedings of the IEEE international conference on computer vision (ICCV)*.

Leutenegger S, Furgale P, Rabaud V, Chli M, Konolige K and Siegwart R (2013) Keyframe-based visual-inertial SLAM using nonlinear optimization. In: *Proceedings of robotics: science and systems (RSS)*.

Li M, Kim BH and Mourikis AI (2013) Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera. In: *2013 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 4712–4719.

Li M and Mourikis AI (2012a) Improving the accuracy of EKF-based visual–inertial odometry. In: *2012 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 828–835.

Li M and Mourikis AI (2012b) Improving the accuracy of EKF-based visual–inertial odometry. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.

Li M and Mourikis AI (2012c) Vision-aided inertial navigation for resource-constrained systems. In: *2012 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 1057–1063.

Li M and Mourikis AI (2013) Optimization-based estimator design for vision-aided inertial navigation. *Robotics: Science and Systems*, pp. 241–248.

Lobo J and Dias J (2007) Relative pose calibration between visual and inertial sensors. *The International Journal of Robotics Research* 26(6): 561–575.

Lynen S, Achtelik MW, Weiss S, Chli M and Siegwart R (2013) A robust and modular multi-sensor fusion approach applied to MAV navigation. In: *2013 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 3923–3929.

Martinelli A (2011) State estimation based on the concept of continuous symmetry and observability analysis: the case of calibration. *IEEE Transactions on Robotics* 27(2): 239–255.

Martinelli A (2014) Visual-inertial structure from motion: observability vs minimum number of sensors. In: *Proceedings of the IEEE international conference on robotics and automation*.

Mei C, Sibley G, Cummins M, Newman PM and Reid ID (2011) RSLAM: a system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision* 94(2): 198–214.

Mirzaei FM and Roumeliotis SI (2007) *IMU–camera Calibration: Bundle Adjustment Implementation*. Technical report, Department of Computer Science and Engineering, University of Minnesota.

Mirzaei FM and Roumeliotis SI (2008) A Kalman filter-based algorithm for IMU–camera calibration: observability analysis and performance evaluation. *IEEE Transactions on Robotics* 24(5): 1143–1156.

Mourikis AI and Roumeliotis SI (2007) A multi-state constraint Kalman filter for vision-aided inertial navigation. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.

Mourikis AI, Trawny N, Roumeliotis SI, Johnson AE, Ansar A and Matthies L (2009) Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics* 25(2): 264–280.

Nerurkar ED, Wu KJ and Roumeliotis SI (2013) C-KLAM: constrained keyframe-based localization and mapping. In: *Proceedings of the workshop on "multi-view geometry in robotics" at the robotics: science and systems (RSS)*.

Nikolic J, Rehder J, Michael Burri PG, Leutenegger S, Furgale PT and Siegwart R (2014) A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.

Ranganathan A, Kaess M and Dellaert F (2007) Fast 3D pose estimation with out-of-sequence measurements. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*.

Roumeliotis SI, Johnson AE and Montgomery JF (2002) Augmenting inertial navigation with image-based motion

estimation. In: *Proceedings IEEE international conference on robotics and automation (ICRA'02)*, vol. 4. IEEE, pp. 4326–4333.

Savage PG (1998) Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *Journal of Guidance, Control, and Dynamics* 21(2): 208–221.

Shin EH and El-Sheimy N (2004) An unscented Kalman filter for in-motion alignment of low-cost IMUs. In: *Position Location and Navigation Symposium, 2004 (PLANS 2004)*. IEEE, pp. 273–279.

Sibley G, Matthies L and Sukhatme G (2010) Sliding window filter with application to planetary landing. *Journal of Field Robotics* 27(5): 587–608.

Strasdat H, Montiel JMM and Davison AJ (2010) Real-time monocular SLAM: why filter? In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.

Strelow D and Singh S (2003) *Online motion estimation from image and inertial measurements*. Available at: https://www.frc.ri.cmu.edu/projects/buzzard/publications/inervis03.pdf

Strelow D and Singh S (2004) Motion estimation from image and inertial measurements. *The International Journal of Robotics Research* 23(12): 1157–1195.

Thrun S and Montemerlo M (2006) The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research* 25(5): 403–430.

Weiss S, Achtelik M, Lynen S, Chli M and Siegwart R (2012) Real-time onboard visual–inertial state estimation and self-calibration of MAVs in unknown environments. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.

Weiss SM (2012) *Vision based navigation for micro helicopters*. PhD Thesis, Eidgenössische Technische Hochschule, ETH Zürich, 2012.