Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback

Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, and Roland Siegwart

Abstract

This paper presents a visual-inertial odometry framework that tightly fuses inertial measurements with visual data from one or more cameras, by means of an iterated extended Kalman filter. By employing image patches as landmark descriptors, a photometric error is derived, which is directly integrated as an innovation term in the filter update step. Consequently, the data association is an inherent part of the estimation process and no additional feature extraction or matching processes are required. Furthermore, it enables the tracking of noncorner-shaped features, such as lines, and thereby increases the set of possible landmarks. The filter state is formulated in a fully robocentric fashion, which reduces errors related to nonlinearities. This also includes partitioning of a landmark's location estimate into a bearing vector and distance and thereby allows an undelayed initialization of landmarks. Overall, this results in a compact approach, which exhibits a high level of robustness with respect to low scene texture and motion blur. Furthermore, there is no time-consuming initialization procedure and pose estimates are available starting at the second image frame. We test the filter on different real datasets and compare it with other state-of-the-art visual-inertial frameworks. Experimental results show that robust localization with high accuracy can be achieved with this filter-based framework.

Keywords

Visual-inertial odometry, iterated extended Kalman filter, photometric error, tight information fusion, multiple cameras

1. Introduction

Robust and high-bandwidth estimation of ego-motion is a key factor in enabling the operation of autonomous robots. For dynamically controlled robots, such as aerial vehicles or legged robots, a reliable state estimate is essential; failures of the state estimator can quickly lead to damage of the hardware and its surroundings. Thus, as autonomous robots become more capable and extend their range of applications, it is essential that the corresponding ego-motion estimation can perform well in increasingly difficult environments. The corresponding selection of sensors should be kept as lightweight and low-cost as possible in order to employ them on a wide range of robotic systems. Furthermore, in the context of vision-based estimation, extreme conditions, such as strongly varying lighting, missing textures, fast motion, or dynamic objects may need to be accounted for.

Past research has shown that combining complementary information from an inertial measurement unit (IMU) and visual sensors can be a very capable approach in terms of accuracy and reliability. Consequently, this approach has been successfully applied to robotic systems, such as unmanned aerial robots (Shen et al., 2014; Weiss et al.,

2013) or legged robots (Ma et al., 2016; Stelzer et al., 2012). Since assessing the precision of an algorithm is often simpler than evaluating its robustness, many researchers have focused on optimizing the accuracy of their approaches. The evaluation is typically done by measuring the accumulated position error over given traveled distances. Depending on the experimental setup, state-of-the-art algorithms reduce position errors to 0.1% of the traveled distance (Forster et al., 2017; Leutenegger et al., 2015; Usenko et al., 2016). Such a demonstration of high accuracy can serve as a surrogate for the good function of an approach. However, all odometry frameworks inherently suffer from drift and, if the primary goal is localization accuracy, a back-end framework for global mapping, relocalization, and loop closure will be indispensable (e.g., Lynen et al. (2015)). Furthermore, if the ego-motion estimation is employed within a feedback loop on an autonomous robot,

---

ETH Zurich, Switzerland

Corresponding author:
Michael Bloesch, Autonomous Systems Lab, ETH Zurich, LEE J 225, Leonhardstrasse 21, 8092 Zurich, Switzerland.
Email: bloesch.michael@gmail.com

other aspects, such as reliability and estimation time-delay, become more important.

The well-established Kalman filtering techniques represent sensor fusion frameworks that allow computationally efficient and high-bandwidth state estimation. Owing to the inherent marginalization, the filter states at each timestep can refer to different physical quantities, e.g., a landmark's position can be estimated with respect to the moving sensor frame (and thereby represent a varying quantity over time). This enables the use of a fully robocentric formulation of the state and thereby reduces observability- and nonlinearity-related issues (Castellanos et al., 2004). To mitigate the problem of intrinsic unobservability of a landmark's initial distance from the observer, the landmark position can be parametrized by its bearing vector and distance (Montiel et al., 2006). Consequently a landmark's distance can be initialized with a high uncertainty without affecting its bearing vector estimate (which can be initialized with a low uncertainty). This becomes especially invaluable for scenarios with fast motions and short feature tracks, as it allows a seamless initialization of landmarks and thereby the extraction of visual information from a landmark's second observation onwards. In this context, a sound representation of the filter state is crucial for the applicability of sensor fusion algorithms. An approach that has become increasingly popular for 3D orientations is based on a manifold encapsulation technique (Hertzberg et al., 2011), which, in this work, is also applied to bearing vectors.

The proposed approach combines an iterated extended Kalman filter, a fully robocentric formulation of visual-inertial odometry, and a photometric error model. This is achieved by associating every landmark with a multilevel patch feature, where the innovation term is derived by projecting the patch into the current image and computing the photometric error for every patch pixel. To keep the computational effort tractable, a QR-decomposition based reduction is applied to obtain an *equivalent* 2D innovation term for each observed landmark. This method takes into account the local texture of a landmark and thereby gains more information along the directions where the image gradients are stronger. In addition, this offers the possibility of tracking noncorner-shaped features, such as lines, increasing the set of possible image features; this is beneficial in scenarios with missing texture.

The main contributions of this work lie in the *fully robocentric* formulation of a visual-inertial odometry, as well as in the *tight* integration of the photometric error. The application of manifold encapsulation to bearing vectors can be seen as an enabling technique and allows for a sound robocentric representation of 3D landmarks. Furthermore, in contrast with our previous work (Bloesch et al., 2015), the present approach inherently takes care of landmark tracking by employing an iterated extended Kalman filter. This allows per-landmark iterative updates and thus provides simultaneous landmark tracking and full state refinement,

while considering inertial and visual data. To the best of our knowledge, this *tight* integration of data association and estimation process has no precedent in visual-inertial odometry.

All in all, this paper describes a *fully robocentric* and *direct* visual-inertial odometry framework, which runs in real time on computationally constrained platforms. To increase robustness and usability, we implement multi-camera support (with or without overlapping field of view) and enable online calibration of camera-IMU extrinsics. An in-depth derivation and evaluation of the framework is provided, including experiments on publicly available datasets (Burri et al., 2016). Our framework, which we refer to as Rovio (robust visual-inertial odometry), is implemented in C++ and is available as open-source software (https://github.com/ethz-asl/rovio).

2. Related work

Within the field of computer vision, Davison (2003) proposed one of the first real-time 3D monocular localization and mapping frameworks. Like the work in this paper, Davison (2003) made use of an extended Kalman filter framework to co-estimate the absolute position of 3D landmarks. Since then, various research groups have contributed improvements and proposed further approaches. A key issue is to improve the consistency of the estimation framework, which is affected by its inherent nonlinearity (Castellanos et al., 2004; Julier and Uhlmann, 2001). One approach is to make use of a robocentric representation for the tracked landmarks and thereby significantly reduce the effect of nonlinearities (Castellanos et al., 2004; Civera et al., 2009). As an alternative, Huang et al. (2008) propose the use of a so-called observability constrained extended Kalman filter, whereby inconsistencies can be avoided by using special linearization points while evaluating the system Jacobians.

A somewhat related problem is the choice of the specific representation of a landmark's location. Since the depth of a newly detected landmark is unknown for monocular setups, the initial 3D location estimate exhibits a high uncertainty along the viewing axis. To integrate this landmark from the beginning into the estimation process, Montiel et al. (2006) proposed the use of an inverse-depth parametrization. They parametrize each landmark location by the camera position where the landmark was initially detected and by a bearing vector (parametrized with azimuth and elevation angles), as well as the inverse depth of the landmark. The increase in consistency for the inverse-depth parametrization and other parametrization methods was further analyzed and confirmed by Solà et al. (2012).

While most standard visual odometry approaches are based on detected and tracked point landmarks as sources of visual information, so-called *direct* approaches directly use the image intensities in their estimation framework. Jin et al. (2003) propose modeling the environment as a

collection of planar patches and deriving a corresponding photometric error between camera frames. Their work is similar to ours in that they also embed the photometric error directly into a filtering framework (but they do not use any inertial data, which limits them to slow motions). Molton et al. (2004) also track locally planar image patches in a filter-based SLAM framework. By employing gradient-based image alignment, they also co-estimate surface normals but keep data association separate from the subsequent extended-Kalman-filter-based information fusion. Silveira et al. (2008) also use planar regions and minimize the photometric error with respect to a reference frame in order to estimate the relative motion, as well as other parameters, such as illumination parameters and patch normals. They subsequently merge the output in an extended Kalman filter. More recently, by employing highly optimized single-instruction multiple data implementations, first real-time, CPU-based approaches for semi-dense motion estimation using a monocular camera (Engel et al., 2014; Forster et al., 2014) have been proposed.

Incorporating inertial measurements in the estimation can significantly improve the robustness of the system, provide the estimation process with the notion of gravity, and allow for a more accurate and high-bandwidth estimation of the velocities and rotational rates. By adapting the original extended Kalman filter proposed by Davison (2003), additional IMU measurements can be relatively simply integrated into the ego-motion estimation, whereby calibration parameters can be co-estimated online (Jones and Soatto, 2011; Kelly and Sukhatme, 2011). Leutenegger et al. (2015) describe a *tightly* coupled approach, in which the robot trajectory and sparse 3D landmarks are estimated in a joint optimization problem using inertial error terms, as well as the reprojection error of the tracked landmarks in the camera images. This is done in a windowed bundle adjustment approach over a set of keyframe images and a temporal inertial measurement window. Similarly, Mourikis and Roumeliotis (2007) estimate the trajectory in an IMU-driven filtering framework using the reprojection error of 3D landmarks as measurement updates. Instead of adding the landmarks to the filter state, they marginalize them out using a nullspace decomposition, thus leading to a small filter state size. Since inertial measurements are often obtained at a higher rate than image data, methods for combining inertial measurements are desirable to reduce computational costs. Forster et al. (2017) have presented a concise pre-integration method, in which IMU measurements can be efficiently included in a factor graph framework. Recently, Usenko et al. (2016) have extended their previous work on semi-dense visual odometry (Engel et al., 2014) to integrate inertial measurements. They minimize a joint energy term composed of visual and inertial error terms to estimate the ego-motion of their sensor.

Probably the most comparable work to ours was developed by Tanskanen et al. (2015), who implemented a framework based on an extended Kalman filter for merging patch-based photometric errors with IMU measurements. They parametrize their landmarks by the pose of the camera when the landmark was detected as well as the corresponding bearing vector and inverse depth (analogously to Montiel et al. (2006)). Our work differs in that it uses a fully robocentric formulation of the current state, which has various implications on the filtering and visual processing framework. We also integrate a QR-decomposition based measurement space reduction and perform per-landmark update iterations, which are both key to the efficiency and accuracy of our system.

3. Prerequisites on rotations and unit vectors

3.1. Notation

For better readability and comprehensibility, we give a brief overview of the employed notation and the algebra of 3D rotations and unit vectors. Three different coordinate frames are used throughout the paper: the inertial world coordinate frame, $\mathcal{I}$, the IMU fixed coordinate frame, $\mathcal{B}$, and the camera fixed coordinate frame, $\mathcal{C}$. Only in Section 6, where multi-camera setups are discussed, will the distinction between different camera frames be made. The origin associated with a specific coordinate frame is denoted by the same symbol. In this context, a term of the form $_{\mathcal{I}}r_{\mathcal{BC}}$ denotes the coordinates of a vector from the origin of $\mathcal{B}$ to the origin of $\mathcal{C}$, expressed in the coordinate frame $\mathcal{I}$. Furthermore, $q_{\mathcal{BI}}$ is employed in an abstract manner to represent the rotation between a frame $\mathcal{I}$ and $\mathcal{B}$. A good way to think of a rotation is as a mapping $q_{\mathcal{BI}} : \mathbb{R}^3 \to \mathbb{R}^3$ between the two associated coordinate frames. Given a physical vector $r_{\mathcal{BC}}$, a rotation maps the corresponding coordinates from the right index frame to the left index frame, e.g., $_{\mathcal{B}}r_{\mathcal{BC}} = q_{\mathcal{BI}} (_{\mathcal{I}}r_{\mathcal{BC}})$. We also employ the mapping $C(q) : SO(3) \to \mathbb{R}^{3\times3}$, which is defined such that $q(r) \triangleq C(q)\,r$ and basically returns the 3×3 rotation matrix.

As further abbreviations, we use $v_{\mathcal{B}}$ to denote the absolute velocity of $\mathcal{B}$, and $\omega_{\mathcal{IB}}$ for the vector describing the relative rotational velocity of the coordinate frame $\mathcal{B}$ with respect to the coordinate frame $\mathcal{I}$. In some cases, we use further denotations, such as tildes (measurements) or hats (estimates), if we want to highlight a specific aspect of a quantity. The superscript $^{\times}$ is used to denote the skew symmetric matrix $v^{\times} \in \mathbb{R}^{3\times3}$ of a vector $v \in \mathbb{R}^3$.

3.2. Representation of 3D rotations

The set of 3D rotations, the special orthogonal group $SO(3)$ with group operation $\otimes$, is not a vector space and thus adaptations are required to enable traditional optimization-based methods (e.g., filtering). A mathematically sound and increasingly popular method is to map the region around a selected linearization point to a proper vector space and thereby introduce a local parametrization. There are slight variations in how this concept is formalized; we follow

the approach of Hertzberg et al. (2011), as we found it to provide a useful level of abstraction for modeling.

$SO(3)$ is a Lie group and has a logarithmic and an exponential map, which map to and from a corresponding Lie algebra $\mathbb{R}^3$

$$\log : SO(3) \to \mathbb{R}^3$$
$$q_{\mathcal{BI}} \mapsto \log(q_{\mathcal{BI}}) = \theta_{\mathcal{BI}} \tag{1}$$
$$\exp : \mathbb{R}^3 \to SO(3)$$
$$\theta_{\mathcal{BI}} \mapsto \exp(\theta_{\mathcal{BI}}) = q_{\mathcal{BI}} \tag{2}$$

There is a certain amount of freedom in selecting these maps. Here, we select the exponential and logarithmic maps such that $\theta_{\mathcal{BI}}$ in equations (1) and (2) coincides with the passive rotation vector of the rotation $q_{\mathcal{BI}}$. We can write the following identities (the last identity is known as Rodrigues' formula)

$$\exp(-\theta) = \exp(\theta)^{-1} \tag{3}$$

$$\exp(q(\theta)) = q \otimes \exp(\theta) \otimes q^{-1} \tag{4}$$

$$C(\exp(\theta)) = I - \frac{\sin(\|\theta\|)\,\theta^\times}{\|\theta\|} + \frac{(1 - \cos(\|\theta\|))\,\theta^{\times 2}}{\|\theta\|^2} \tag{5}$$

The exponential and logarithmic maps can be used to introduce a boxplus ($\boxplus$) and a boxminus ($\boxminus$) operator, which adopt the role of addition and subtraction operators for rotations. Using a slightly different notation from Hertzberg et al. (2011), we define

$$\boxplus : SO(3) \times \mathbb{R}^3 \to SO(3)$$
$$q, \theta \mapsto \exp(\theta) \otimes q \tag{6}$$
$$\boxminus : SO(3) \times SO(3) \to \mathbb{R}^3$$
$$q, p \mapsto \log(q \otimes p^{-1}) \tag{7}$$

Like regular addition and subtraction, both operators fulfill the following axioms

$$q \boxplus 0 = q \tag{8}$$
$$(q \boxplus \theta) \boxminus q = \theta \tag{9}$$
$$q \boxplus (p \boxminus q) = p \tag{10}$$

This approach distinguishes between actual rotations, which are on $SO(3)$ (Lie group), and differences of rotations, which lie on $\mathbb{R}^3$ (Lie algebra). These operators take care of appropriately transforming the elements into their respective spaces and allow a smooth embedding of rotational quantities in filtering and optimization frameworks.

The definition of differentials involving rotation can be adapted by replacing the regular plus and minus operators with the boxplus and boxminus operators. For instance, the differential of a mapping $q(x): \mathbb{R} \to SO(3)$ can be defined as

$$\frac{\partial}{\partial x} q(x) := \lim_{\epsilon \to 0} \frac{q(x + \epsilon) \boxminus q(x)}{\epsilon} \tag{11}$$

The same can be done for the other way round, where we have a mapping $x(q): SO(3) \to \mathbb{R}$

$$\frac{\partial}{\partial q} x(q) := \lim_{\epsilon \to 0} \begin{bmatrix} \frac{x(q \boxplus (e_1 \epsilon)) - x(q)}{\epsilon} \\ \frac{x(q \boxplus (e_2 \epsilon)) - x(q)}{\epsilon} \\ \frac{x(q \boxplus (e_3 \epsilon)) - x(q)}{\epsilon} \end{bmatrix}^{\mathrm{T}} \tag{12}$$

where $e_{1/2/3}$ are orthonormal basis vectors. This results in the following frequently-used derivatives (these may vary depending on conventions)

$$\partial/\partial t\,(q_{\mathcal{BI}}(t)) = {}_{\mathcal{B}}\omega_{\mathcal{IB}}(t) \tag{13}$$
$$\partial/\partial q\,(q(r)) = (q(r))^\times \tag{14}$$
$$\partial/\partial q\,(q^{-1}) = -C(q)^{\mathrm{T}} \tag{15}$$
$$\partial/\partial q\,(q \otimes p) = I \tag{16}$$
$$\partial/\partial q\,(p \otimes q) = C(p) \tag{17}$$
$$\partial/\partial \theta\,(\exp(\theta)) = \Gamma(\theta) \tag{18}$$
$$\partial/\partial q\,(\log(q)) = \Gamma^{-1}(\log(q)) \tag{19}$$

The derivative of the exponential map is given by the Jacobian $\Gamma(\theta) \in \mathbb{R}^{3 \times 3}$, which has the following analytical expression

$$\Gamma(\theta) = I - \frac{(1 - \cos(\|\theta\|))\,\theta^\times}{\|\theta\|^2} + \frac{(\|\theta\| - \sin(\|\theta\|))\,\theta^{\times 2}}{\|\theta\|^3} \tag{20}$$

These derivations are independent of the actual numerical parametrization of 3D rotations (e.g., quaternions or rotation matrices) as long as it is lossless and the associated operations adhere to certain rules. A more detailed discussion and derivations can be found in Bloesch et al. (2016). The employed parametrization in the implementation is based on unit quaternions using the Hamilton convention. For a unit quaternion q with real part q_0 and imaginary part $\check{q}$, we employ the following exponential and logarithmic maps

$$\exp(\theta) = (q_0, \check{q}) = \left(\cos(\|\theta\|/2), \sin(\|\theta\|/2)\,\frac{\theta}{\|\theta\|} \right) \tag{21}$$

$$\log(q) = 2\,\mathrm{atan2}(\|\check{q}\|, q_0)\,\frac{\check{q}}{\|\check{q}\|} \tag{22}$$

where $\theta \in \mathbb{R}^3$ can be interpreted as the corresponding rotation vector. Both maps, together with the quaternion multiplication, are the only parametrization-specific operations that are required.

3.3. Representation of 3D unit vectors

While the handling of rotations outlined in Section 3.2 has been used similarly in previous filtering frameworks (Bloesch et al., 2013; Li and Mourikis, 2013), we extend the methodology to 3D unit vectors on the 2-sphere S^2. This is done analogously to Hertzberg et al. (2011), except that we employ a parametrization yielding simple analytical

derivatives and guarantee second-order differentiability. A main issue with 3D unit vectors is the problem of selecting orthonormal vectors for spanning the tangent space such that a suitable difference operator can be defined. Assigning orthonormal vectors to every point on the 2-sphere creates a vector field and, as stated by the "hairy ball theorem", there is no continuous way of doing so over the full 2-sphere. To solve this issue, we employ a rotation, $\mu \in SO(3)$, as an underlying representation for unit vectors and define the following quantities

$$n(\mu) := \mu(e_z) \in S^2 \subset \mathbb{R}^3 \qquad (23)$$

$$N(\mu) := \left[\mu(e_x), \mu(e_y) \right] \in \mathbb{R}^{3 \times 2} \qquad (24)$$

where $e_{x/y/z} \in \mathbb{R}^3$ are the basis vectors of an arbitrary orthonormal coordinate system. The actual unit vector is given by $n(\mu)$, which results when rotating e_z by μ (if the context is clear, we directly refer to the unit vector using μ). The matrix $N(\mu)$ is composed of the rotated e_x and e_y and spans the tangent space. While such a construction of the tangent space is not deterministic, since infinitely many rotations μ provide the same unit vector $n(\mu)$, we have the advantage that smooth transformations of the rotation μ induce smooth transformations of the associated tangent space.

The tangent space can be used to define the following boxplus and boxminus operators

$$\boxplus : SO(3) \times \mathbb{R}^2 \to SO(3)$$
$$\mu, u \mapsto \exp(N(\mu) u) \otimes \mu \qquad (25)$$
$$\boxminus : SO(3) \times SO(3) \to \mathbb{R}^2$$
$$v, \mu \mapsto N(\mu)^\mathrm{T} \theta(\mu, v) \qquad (26)$$

where θ maps two unit vectors to the minimal rotation vector between them

$$\theta(n(\mu), n(v)) = \frac{\mathrm{acos}(n(v)^\mathrm{T} n(\mu))}{\| n(v) \times n(\mu) \|} n(v) \times n(\mu) \qquad (27)$$

A visualization of the 2-sphere and the tangent space for a specific μ is given in Figure 1.

The concept is slightly more complicated than in the case of 3D rotations, since we truly over-parametrize a 3D unit vector (no constraint is imposed on the underlying rotation). To overcome this, we use a different notion of equivalence, where we define that two unit vector parametrizations μ and v are equivalent ($\mu \sim v$) *iff* $n(\mu) = n(v)$. With this, the axioms proposed by Hertzberg et al. (2011) are fulfilled

$$\mu \boxplus 0 = \mu \qquad (28)$$
$$(\mu \boxplus u) \boxminus \mu = u \qquad (29)$$
$$\mu \boxplus (v \boxminus \mu) \sim v \qquad (30)$$

A technical detail with this parametrization is that whenever representing a difference, $u \in \mathbb{R}^2$, we have to keep track of the corresponding tangent space. Mathematically, if we

have two rotations μ and v with $\mu \sim v$, it does not follow that $(\mu \boxplus u) \sim (v \boxplus u)$.

Similarly to the derivatives given in Section 3.2, the most commonly used derivatives for terms involving 3D unit vectors are given by

$$\partial / \partial t\, (\mu(t)) = -N(\mu(t))^\mathrm{T}\, n(\mu(t))^\times$$
$$\cdot\, \partial / \partial t\, (n(\mu(t))) \qquad (31)$$
$$\partial / \partial \mu\, (n(\mu)) = n(\mu)^\times N(\mu) \qquad (32)$$
$$\partial / \partial \mu\, \left(N(\mu)^\mathrm{T}\, r \right) = -N(\mu)^\mathrm{T}\, r^\times N(\mu) \qquad (33)$$

The first identity relates the time derivative of a 3D unit vector on its manifold to its time derivative in the 3D vector space. The second expression is the derivative of the unit vector in 3D with respect to its minimal 2D representation. These identities can be very useful when computing Jacobians, whereby the chain rule can be applied to compute the derivatives of more complex terms. An example will be provided when discussing the process model of the bearing vector state of 3D landmarks (see Section 5.3 and the appendix).

All in all, the proposed unit vector parametrization yields analogous advantages, as obtained when employing the well-established minimal 3D rotation parametrization. This includes a singularity-free parametrization, which comes with relatively simple differentials. Furthermore, the parametrization of the tangent space is orthogonal and the direction of the boxminus operation is in accordance with the shortest path between two given unit vectors (taking a step along $v \boxminus \mu$ is optimal for going from μ to v, see Figure 1). Other parametrizations, such as azimuth and elevation angles, do not meet these properties and often exhibit singular configurations.

4. Multilevel patches and photometric error

4.1. Multilevel patch features

Along the lines of other landmark-based visual odometry approaches (Davison, 2003) we model landmarks as distinguished stationary 3D locations in the environment. Each landmark is associated with a multilevel patch feature $P = \{P_0, \ldots, P_L\}$, which is composed of multiple $n \times n$ image patches, P_l, extracted at the projected landmark location on image level l. In the current default implementation, we extract 6×6 image patches at the second and third pyramid level (down-sampling factor of 2). These parameters can and should be adapted to the actual hardware setup and the scenario. An example is given in Figure 2. The simultaneous use of multiple pyramid levels leads to cross-correlations between pixel intensities. These are not explicitly modeled but can be handled to a certain extent by tuning the corresponding error weighting.

By contrast with a standard feature descriptor, a patch-based descriptor allows computation of a photometric error and thereby enables the use of reprojection errors to be
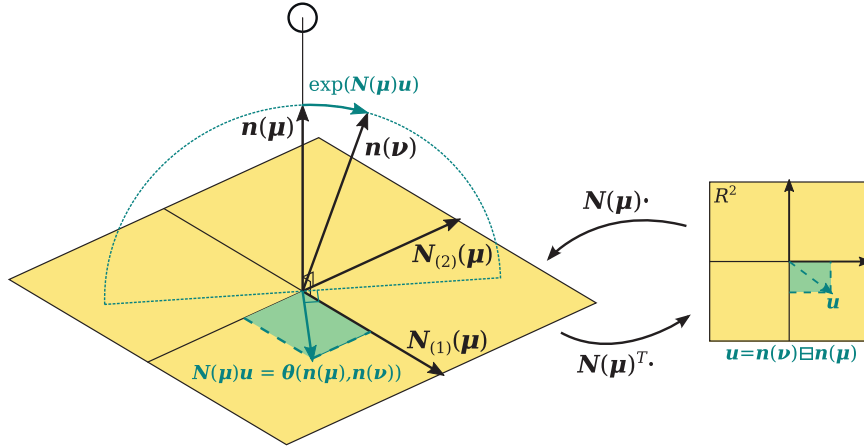
Fig. 1. Representation of 3D unit vectors. The 3D unit vector $n(\mu)$ is represented as the result of applying the rotation μ onto the *z*-axis of an arbitrary inertial coordinate system. The images of the *x*- and *y*-axes are used to define an orthonormal plane to the unit vector. This plane then represents the tangent space used for the construction of the boxplus and boxminus operations. The boxminus operator takes two 3D unit vectors and represents their difference in $\mathbb{R}^2$. Conversely, the boxplus operator takes an element from $\mathbb{R}^2$ and applies it on a 3D unit vector.
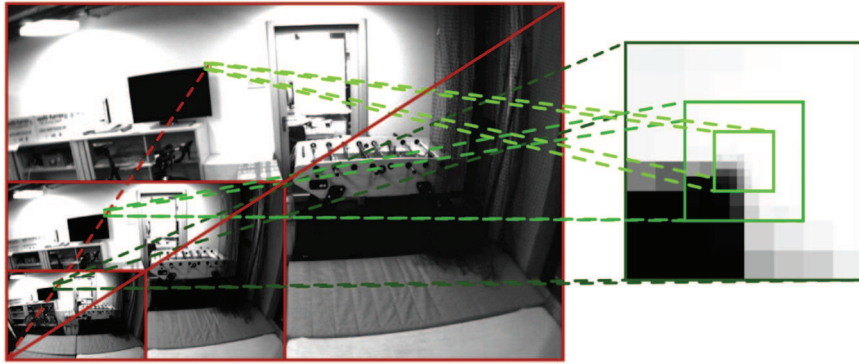

Fig. 2. Construction of multilevel patch from image pyramid. Here, each single patch is composed of 8×8 pixels and three pyramid levels are depicted. These settings may vary in the actual implementation.

avoided. Taking the information of every pixel gives much richer information about the environment, which not only helps in improving robustness in bad lighting conditions, but also inherently takes into account the texture of the tracked image patch. For instance, it enables the integration of edge-shaped features, whereby the gained information is along the perpendicular direction of the edge. In comparison, approaches based on reprojection errors typically attempt to minimize the distance between the predicted and detected feature location. This ignores the local texture around the landmark and, if no additional measures are taken, all landmarks are weighted equally.

4.2. Projection model and linear warping

Given the bearing vector μ of a landmark, the pixel coordinates in the camera frame can be retrieved by using the camera model π. Assuming a known intrinsic calibration, the pixel coordinates p can be directly expressed by $p = \pi(\mu)$. If the camera is moving, the feature moves through the

image and is seen from a different perspective. To account for a certain patch distortion effect, a linear warping matrix is tracked with each feature. This is done by concatenating all Jacobians when transforming a landmark location. For instance, if we detect a feature in some frame at pixel p_1, transform the corresponding bearing vector $\mu_1 = \pi^{-1}(p_1)$ with a process model $\mu_2 = f(\mu_1)$, and then re-project the bearing vector in a subsequent frame $p_2 = \pi(\mu_2)$, we obtain the following linear warping matrix

$$D = \frac{\partial \pi(\mu_2)}{\partial \mu_2} \frac{\partial f(\mu_1)}{\partial \mu_1} \frac{\partial \pi^{-1}(p_1)}{\partial p_1} \in \mathbb{R}^{2 \times 2} \qquad (34)$$

In essence, this maps the two patch axes (which were aligned with the image axes) from the point of patch extraction to the two distorted patch axes in the projection image. This approach tracks the distortion locally around the patch and ignores any larger-scale information, such as the geometric shape of a patch. To avoid large distortions and the accumulation of errors, the patches are re-extracted regularly and the warping matrix is reset to identity.

4.3. Photometric error and patch alignment

The photometric error between a given multilevel patch feature and a specific image is computed by extracting a warped patch at the estimated location and evaluating the pixel-wise intensity error. For a given multilevel patch feature (with coordinates p and multilevel patch $P = \{P_0, \ldots, P_L\}$) at a specific image level l and patch pixel p_j, the photometric error can be formalized as

$$e_{l,j}(p, P, I, D) = P_l(p_j) - a\, I_l(p\, s_l + D p_j) - b \quad (35)$$

where I_l is the image at the pyramid level l and $s_l = 0.5^l$ is a scaling factor to account for the down-sampling. The linear warping matrix D is used to map patch pixel coordinates to image coordinates. Furthermore, inter-frame illumination changes are taken into account by employing an affine intensity model composed of the scalars a and b (both are marginalized out). Figure 3 depicts the photometric error between a patch and its measurement in an image at a predicted location p.

If we minimize the squared error terms for a multilevel patch, we obtain a patch alignment algorithm that is very similar to the well-known Kanade–Lucas–Tomasi (KLT) feature tracker (Lucas and Kanade, 1981; Shi and Tomasi, 1994). A slight difference is given by the fact that we optimize over several image levels at once. The minimization can be solved by a Gauss–Newton method, which iteratively linearizes the optimization problem around an estimated patch location $\hat{p}$

$$b(\hat{p} + \delta p, P, I, D) = A(\hat{p}, I, D)\,\delta p + b(\hat{p}, P, I, D) \quad (36)$$

where $b(\hat{p}, P, I, D)$ represents the stacked error terms from equation (35) and $A(\hat{p}, I, D)$ represents the corresponding Jacobian. The corresponding normal equations are then given by

$$A(\hat{p}, I, D)^{\mathrm{T}} A(\hat{p}, I, D)\,\delta p = -A(\hat{p}, I, D)^{\mathrm{T}} b(\hat{p}, P, I, D) \quad (37)$$

which can be solved for the correction δp. This is analogous to one iteration step of the KLT feature tracker (but is not used as such in Rovio). In Section 5.4, we will demonstrate how equation (36) is leveraged into the innovation term of the employed iterated extended Kalman filter.

Note that, owing to the scaling factor s_l in equation (35), error terms for higher image levels will have a weaker corrective influence on the filter state or the patch alignment. However, they exhibit increased robustness with respect to image blur or bad initial alignment and thus strongly increase the robustness of the overall alignment method.

4.4. Detection and scoring

The detection of new landmarks is based on the FAST corner detector (Rosten and Drummond, 2006), which provides a large number of candidate feature locations. After removing candidates that are close to currently tracked features,

we compute a patch gradient-based score for selecting new features, which are added to the state. This basically represents an adaptation of the Shi–Tomasi score (Shi and Tomasi, 1994) by considering the combined Hessian on multiple image levels, instead of only a single level. The combined Hessian can be directly retrieved from equation (37)

$$H = A(\hat{p}, I, D)^{\mathrm{T}} A(\hat{p}, I, D) \quad (38)$$

where the minimal eigenvalue of H corresponds to the adapted Shi–Tomasi score.

The advantage is that a high score is correlated with the alignment accuracy of the corresponding multilevel patch feature. Instead of returning the minimal eigenvalue, the method can return other eigenvalue based scores, such as the 1- or 2-norm. This is useful in environments with scarce corner data, whereby edge-shaped features can also be considered. Finally, the detection process is also coupled with a bucketing technique to achieve a good distribution of the features within the camera frame.

5. Filter framework

5.1. Iterated extended Kalman filtering

The regular Kalman filter can be interpreted as the recursive optimal solution to the maximum likelihood estimation problem formulated over two subsequent time steps (Bell and Cathey, 1993). Analogously, the extended Kalman filter can be associated with a nonlinear maximum likelihood estimation and can be shown to yield the same result as the first iteration step of a corresponding Gauss–Newton optimization. However, in contrast with its linear counterpart, the extended Kalman filter cannot guarantee to retrieve the optimal solution, whereby linearization errors tend to become larger if the linearization point is further away from the real solution. A possibility of improving this aspect is to make use of an iterated extended Kalman filter, which is basically the recursive form of the Gauss–Newton optimization (Bell and Cathey, 1993).

A nonlinear discrete time system with state x, innovation term y, process noise $w \sim \mathcal{N}(0, W)$, and update noise $n \sim \mathcal{N}(0, R)$ can be written as

$$x_k = f(x_{k-1}, w_{k-1}) \quad (39)$$
$$y_k = h(x_k, n_k) \quad (40)$$

In equation (40), we made use of an implicit formulation of the measurement model, which directly yields the Kalman innovation y_k. This provides more flexibility in the design by allowing the direct integration of residuals. Given an a-posteriori estimate x_{k-1}^+ with covariance P_{k-1}^+, the prediction step of the iterated extended Kalman filter is analogous to the extended Kalman filter and yields the a-priori estimate at the next time step

$$x_k^- = f(x_{k-1}^+, 0) \quad (41)$$
$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^{\mathrm{T}} + G_{k-1} W_{k-1} G_{k-1}^{\mathrm{T}} \quad (42)$$
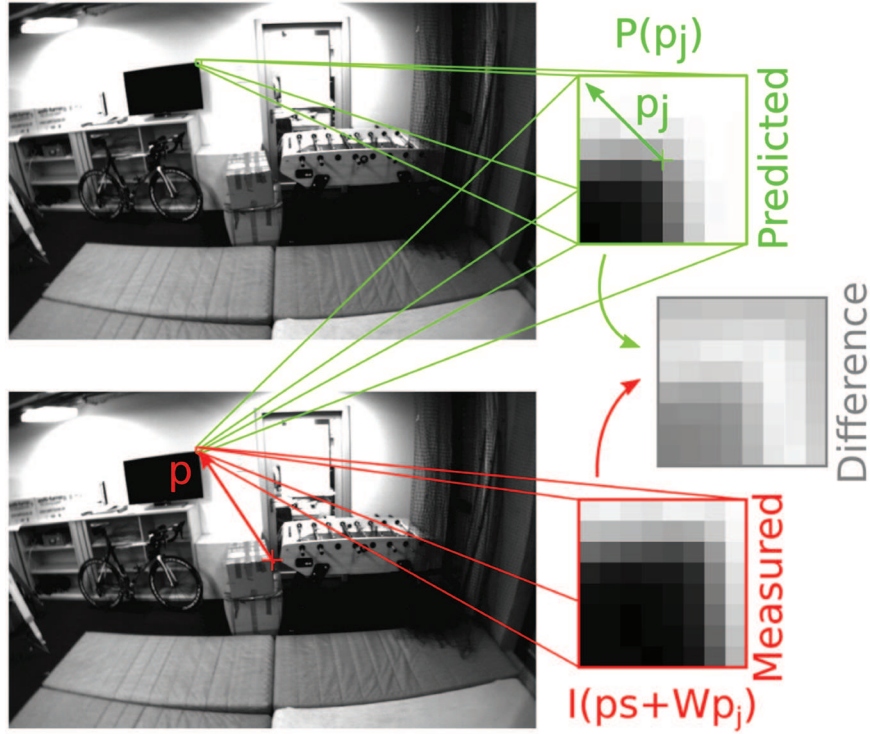
Fig. 3. (Signed) photometric error between a previously extracted patch (green) and its projection onto an image (measured, red) at a predicted location *p*. The bottom-left gray tone of the difference patch represents 0. Only a single image level is depicted. This photometric error is directly used as the innovation term in an iterated extended Kalman filter.   **[AQ: 2]**

with the Jacobians

$$F_{k-1} = \frac{\partial f}{\partial x_{k-1}}(x_{k-1}^+, 0) \tag{43}$$

$$G_{k-1} = \frac{\partial f}{\partial w_{k-1}}(x_{k-1}^+, 0) \tag{44}$$

Analogously to the extended Kalman filter, the update step of the iterated extended Kalman filter can be linked to an optimization problem considering the deviation from the prior x_k^- and the innovation term $h(x_k^+, 0)$

$$\min_{x_k^+} \|x_k^+ \boxminus x_k^-\|_{P_k^{-1}} + \|h(x_k^+, 0)\|_{(J_k R_k J_k^T)^{-1}} \tag{45}$$

However, in contrast with the extended Kalman filter, an iterative scheme is employed, where the problem is linearized around continuously refined linearization points $x_{k,j}^+$, starting with $x_{k,0}^+ = x_k^-$

$$\min_{\Delta x_{k,j}} \|x_{k,j}^+ \boxminus x_k^- + L_{k,j}^{-1} \Delta x_{k,j}\|_{P_k^{-1}}$$
$$+ \|h(x_{k,j}^+, 0) + H_{k,j} \Delta x_{k,j}\|_{(J_k R_k J_k^T)^{-1}} \tag{46}$$

where the Jacobians are updated every iteration step

$$H_{k,j} = \frac{\partial h}{\partial x_k}(x_{k,j}^+, 0) \tag{47}$$

$$J_{k,j} = \frac{\partial h}{\partial n_k}(x_{k,j}^+, 0) \tag{48}$$

$$L_{k,j} = \frac{\partial (x_k^- \boxplus \Delta x)}{\partial \Delta x}(x_{k,j}^+ \boxminus x_k^-) \tag{49}$$

The Jacobian $L_{k,j}$ of the boxplus operator is required to account for the special linearization of certain states, such as rotations or bearing vectors. Its inverse $L_{k,j}^{-1}$ is the corresponding Jacobian of the boxminus operation with respect to the left operand and is required to linearize the deviation of the prior in equation (46). Please note that, owing to the special notion of differentials on manifolds, the Jacobian $L_{k,j}$ is a square matrix (see equation (11)). Also, in the case of vector spaces, this Jacobian will be the identity matrix.

Setting the derivative of the cost function (equation (46)) with respect to the incremental update $\Delta x_{k,j}$ to zero and employing some matrix calculus yields the following recursive solution

$$S_{k,j} = H_{k,j} L_{k,j}^T P_k^- L_{k,j} H_{k,j}^T + J_{k,j} R_k J_{k,j}^T \tag{50}$$

$$K_{k,j} = L_{k,j}^T P_k^- L_{k,j} H_{k,j}^T S_{k,j}^{-1} \tag{51}$$

$$\Delta x_{k,j} = K_{k,j} \left(H_{k,j} L_{k,j} (x_{k,j}^+ \boxminus x_k^-) - h(x_{k,j}^+, 0) \right)$$

$$-\boldsymbol{L}_{k,j}(\boldsymbol{x}_{k,j}^+ \boxminus \boldsymbol{x}_k^-) \tag{52}$$

$$\boldsymbol{x}_{k,j+1}^+ = \boldsymbol{x}_{k,j}^+ \boxplus \Delta\boldsymbol{x}_{k,j} \tag{53}$$

whereby the iteration is terminated when the update $\Delta\boldsymbol{x}_{k,j}$ is below a certain threshold. Finally, the covariance matrix is only updated once the process has converged after n iterations

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_{k,n}\boldsymbol{H}_{k,n})\,\boldsymbol{L}_{k,n}^{\mathrm{T}}\boldsymbol{P}_k^-\boldsymbol{L}_{k,n} \tag{54}$$

Especially in setups with large initial uncertainties, the iterated extended Kalman filter can provide a significant gain in convergence and accuracy. Using a termination criterion based on the magnitude of the performed correction, the computational overhead can be limited to cases with large update corrections (e.g., the initial measurements of a new landmark). Once the state has properly converged, the number of iterations can be kept to a minimum and the computational costs remain similar to those of the regular extended Kalman filter.

5.2. Filter setup and state definition

Like other visual-inertial filtering frameworks (Jones and Soatto, 2011; Kelly and Sukhatme, 2011), inertial measurements are employed to propagate the filter state, while visual measurements are processed and integrated during the filter update step (see Figure 4). The proposed filter setup differs in that it makes use of a fully robocentric formulation of the filter state, which has previously been tested in vision-only approaches (Civera et al., 2009). The advantage of this formulation is that the position and yaw states, which are unobservable, can be fully decoupled from the rest of the filter states. This decreases the noise magnitude and improves the consistency of relevant states, such as velocity or inclination angles. Conversely, noise from the gyroscope affects all states that need to be rotated during the state propagation (see Section 5.3). However, since the gyroscope noise is often relatively small and because most states are observable, this does not represent a significant issue.

The state of the filter is composed of the following elements (including N visual landmarks)

$$\boldsymbol{x} := \left(\boldsymbol{r}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{b}_f, \boldsymbol{b}_\omega, \boldsymbol{c}, \boldsymbol{z}, \boldsymbol{\mu}_0, \dots, \boldsymbol{\mu}_N, \rho_0, \dots, \rho_N\right) \tag{55}$$

with:

- $\boldsymbol{r} := {}_{\mathcal{B}}\boldsymbol{r}_{\mathcal{IB}}$: robocentric position of IMU;
- $\boldsymbol{v} := {}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}}$: robocentric velocity of IMU;
- $\boldsymbol{q} := \boldsymbol{q}_{\mathcal{IB}}$: attitude of IMU (map from $\mathcal{B}$ to $\mathcal{I}$);
- $\boldsymbol{b}_f$: additive bias on accelerometer (expressed in $\mathcal{B}$);
- $\boldsymbol{b}_\omega$: additive bias on gyroscope (expressed in $\mathcal{B}$);
- $\boldsymbol{c} := {}_{\mathcal{B}}\boldsymbol{r}_{\mathcal{BC}}$: linear part of IMU-camera extrinsics;
- $\boldsymbol{z} := \boldsymbol{q}_{\mathcal{CB}}$: rotational part of IMU-camera extrinsics;
- $\boldsymbol{\mu}_i$: bearing vector to landmark i (expressed in $\mathcal{C}$);
- ρ_i: distance parameter of landmark i.

The generic parametrization for the distance d_i of a landmark i is given by the mapping $d_i = d(\rho_i)$ with derivative $d'(\rho_i)$. In the context of this work, we mainly tested the inverse distance parametrization, $d(\rho_i) = 1/\rho_i$. A brief comparison with the regular distance parametrization is provided in Section 7.3.

Rotations $(\boldsymbol{q}, \boldsymbol{z})$ and bearing vectors $(\boldsymbol{\mu}_i)$ are parametrized as detailed in Sections 3.2 and 3.3. This means that quantities such as differences, uncertainties, and errors are represented as elements of a vector space with minimal dimensions, i.e., 3D for rotations and 2D for bearing vectors. By using combined bearing vector and distance parametrization, landmarks can be initialized in an *undelayed* manner and can be integrated into the filter at detection time. The distance of a landmark is initialized with a fixed value or, if sufficiently converged, with an estimate of the current average scene distance. The corresponding covariance is set to a very large value. In comparison with other inverse-depth parametrizations, we do not overparametrize the 3D landmark location estimates, whereby each landmark corresponds to three columns in the covariance matrix of the state (two for the bearing vector and one for the distance parameter). This also avoids the need for re-parametrization (Solà et al., 2012).

A singularity-free parametrization of bearing vectors on the full unit sphere is essential here. It enables the proper representation of bearing vectors and their uncertainty estimates, even if outside the field of view of the camera. Furthermore, limiting the validity of the parametrization to a certain region would render online camera-IMU extrinsics calibration and multi-camera support more difficult.

5.3. State propagation

The state propagation is driven by the proper acceleration measurement, $\tilde{\boldsymbol{f}} = {}_{\mathcal{B}}\tilde{\boldsymbol{f}}_{\mathcal{B}}$, and the rotational rate measurement, $\tilde{\boldsymbol{\omega}} = {}_{\mathcal{B}}\tilde{\boldsymbol{\omega}}_{\mathcal{IB}}$. Both measurements are modeled as noise- and bias-affected, leading to the following bias-corrected but noise-affected estimates

$$\hat{\boldsymbol{f}} = \tilde{\boldsymbol{f}} - \boldsymbol{b}_f - \boldsymbol{w}_f \tag{56}$$

$$\hat{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}} - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega \tag{57}$$

Together with the estimated camera linear velocity and rotational rate

$$\hat{\boldsymbol{v}}_{\mathcal{C}} = \boldsymbol{z}(\boldsymbol{v} + \hat{\boldsymbol{\omega}}^\times \boldsymbol{c}) \tag{58}$$

$$\hat{\boldsymbol{\omega}}_{\mathcal{C}} = \boldsymbol{z}(\hat{\boldsymbol{\omega}}) \tag{59}$$

this yields the following set of continuous differential equations

$$\dot{\boldsymbol{r}} = -\hat{\boldsymbol{\omega}}^\times \boldsymbol{r} + \boldsymbol{v} + \boldsymbol{w}_r \tag{60}$$

$$\dot{\boldsymbol{v}} = -\hat{\boldsymbol{\omega}}^\times \boldsymbol{v} + \hat{\boldsymbol{f}} + \boldsymbol{q}^{-1}(\boldsymbol{g}) \tag{61}$$

$$\dot{\boldsymbol{q}} = -\boldsymbol{q}(\hat{\boldsymbol{\omega}}) \tag{62}$$

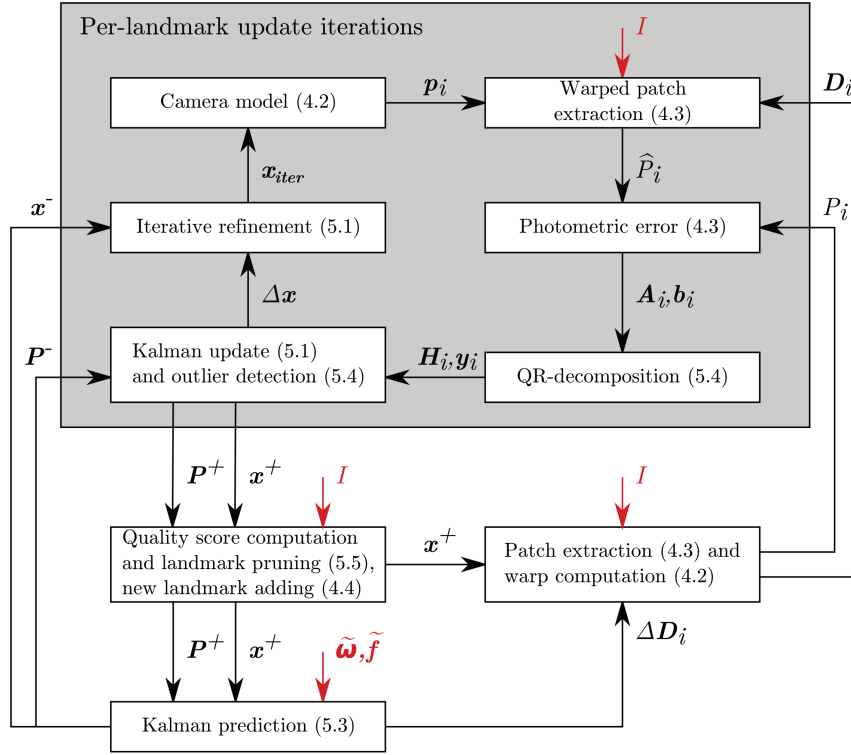$$\dot{\boldsymbol{b}}_f = \boldsymbol{w}_{bf} \tag{63}$$

Fig. 4. Flow graph of filter framework. The processing of the inertial ($\tilde{\omega}, \tilde{f}$) and visual ($I$) measurements (red terms) is tightly embedded in the iterated extended Kalman filter framework. The filter state x and covariance P are alternatively processed by an IMU-based prediction step and a vision-based update step. Most of the visual information is processed as part of the iterative update step (gray block). The order and dependencies of the image processing sub-steps are also illustrated and the numbers represent references to the corresponding sections.

$$\dot{b}_\omega = w_{b\omega} \tag{64}$$

$$\dot{c} = w_c \tag{65}$$

$$\dot{z} = w_z \tag{66}$$

$$\dot{\mu}_i = N(\mu_i)^{\mathrm{T}} \left(\hat{\omega}_{\mathcal{C}} + n(\mu_i)^{\times} \frac{\hat{v}_{\mathcal{C}}}{d(\rho_i)} \right) + w_{\mu,i} \tag{67}$$

$$\dot{\rho}_i = -n(\mu_i)^{\mathrm{T}} \hat{v}_{\mathcal{C}}/d'(\rho_i) + w_{\rho,i} \tag{68}$$

The term $N(\mu_i)^{\mathrm{T}}$ projects a 3D vector onto the 2D tangent space at the bearing vector μ_i (see Figure 1). Furthermore, g is the gravity vector expressed in the world coordinate frame, and the terms of the form w_* are white Gaussian noise processes. The corresponding covariance parameters can either be derived from the IMU specifications or can be tuned manually.

Note that the derivatives of bearing vectors and rotations lie within 2D and 3D vector spaces, respectively. This is required to achieve a minimal and consistent representation of the filter state and covariance. While most of these derivatives are relatively well known, the dynamics of the bearing vector and distance parameter are a novelty in this work. We give a sketch of the corresponding derivation in the appendix. It relies on the assumption that a 3D point landmark $\mathcal{F}$ with bearing vector μ and distance parameter

ρ remains stationary with respect to an inertial frame $\mathcal{I}$

$$_{\mathcal{I}}r_{\mathcal{I}\mathcal{F}} = {}_{\mathcal{I}}r_{\mathcal{I}\mathcal{C}} + q_{\mathcal{C}\mathcal{I}}^{-1}(\mu \, d(\rho)) \tag{69}$$

In equation (68), we can observe that the derivative of the distance parameter only depends on the velocity in direction of the bearing vector. Conversely, the derivative of the bearing vector, equation (67), is the sum of a velocity and rotational rate effect, whereby the magnitude of the velocity effect is proportional to the inverse distance of the landmark.

Using an appropriate Euler forward integration scheme, i.e., using the boxplus operator where appropriate, the time continuous equation can be transformed into a set of discrete prediction equations, which are used during the prediction of the filter state. For the attitude, the rotational IMU-camera extrinsics, and the bearing vectors, the discretization yields

$$\begin{aligned} q_{k+1} &= q_k \boxplus (-\Delta t \, q_k(\hat{\omega}_k)) \\ &= \exp(-\Delta t \, q_k(\hat{\omega}_k)) \otimes q_k \\ &= q_k \otimes \exp(-\Delta t \, \hat{\omega}_k) \otimes q_k^{-1} \otimes q_k \\ &= q_k \otimes \exp(\Delta t (b_{\omega,k} + w_{\omega,k} - \tilde{\omega}_k)) \end{aligned} \tag{70}$$

$$z_{k+1} = z_k \boxplus \Delta t w_{z,k}$$
$$= \exp(\Delta t w_{z,k}) \otimes z_k \qquad (71)$$

$$\boldsymbol{\mu}_{i,k+1} = \exp\left(\Delta t\left(\left(\boldsymbol{I} - \boldsymbol{n}(\boldsymbol{\mu}_{i,k})\,\boldsymbol{n}(\boldsymbol{\mu}_{i,k})^{\mathrm{T}}\right)\hat{\boldsymbol{\omega}}_{\mathcal{C}}\right.\right.$$
$$+ \boldsymbol{n}(\boldsymbol{\mu}_{i,k})^{\times}\frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho_{i,k})}$$
$$\left.\left.+ \boldsymbol{N}(\boldsymbol{\mu}_{i,k})\,\boldsymbol{w}_{\mu,i}\right)\right) \otimes \boldsymbol{\mu}_{i,k} \qquad (72)$$

The derivation of the bearing vector discretization is given in the appendix.

In typical visual-inertial sensor setups, the IMU measurements are often obtained at a higher rate than the images. As the proposed propagation step is driven by the IMU measurements, this can result in a high computational burden. A classical approach to mitigate this issue is to make use of IMU pre-integration techniques (Forster et al., 2017) in order to merge multiple IMU measurements into a single prediction step. However, since the duration between two consecutive images remains relatively small, we employ a simpler pre-integration approach, where the Jacobian is evaluated based on the mean of the IMU measurement. Thus, even if multiple IMU measurements are available between two consecutive images, equation (42) is evaluated only once. Compared with the regular solution, no notable performance loss could be observed.

5.4. Direct innovation term and update

In Section 4.3, we discussed how to construct a photometric error term based on the pixel-wise intensity difference between a previously extracted patch and its predicted location in a given image frame. Within an iterated extended Kalman filter, this can be directly used as an innovation term. However, for the multilevel patch format that we use, this would lead to a $6 \times 6 \times 2 = 72$ dimensional error term per patch, inducing very high computational cost. Fortunately, looking at equation (36), one can observe that the entire error term corresponding to a patch P_i and an image I is only dependent on the estimated pixel coordinates $\boldsymbol{p}_i = \pi(\boldsymbol{\mu}_i)$. Thus, the only direct filter state dependency of this error term is given by the bearing vector and an equivalent reduced 2D error term can be derived. This can be achieved by means of a QR-decomposition of the gradient matrix in equation (36)

$$A(\boldsymbol{p}_i, I, \boldsymbol{D}_i) = \boldsymbol{Q}(\boldsymbol{p}_i, I, \boldsymbol{D}_i)\,\boldsymbol{R}(\boldsymbol{p}_i, I, \boldsymbol{D}_i)$$
$$= \begin{bmatrix} \boldsymbol{Q}_1(\boldsymbol{p}_i, I, \boldsymbol{D}_i) & \boldsymbol{Q}_2(\boldsymbol{p}_i, I, \boldsymbol{D}_i) \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_1(\boldsymbol{p}_i, I, \boldsymbol{D}_i) \\ \boldsymbol{0} \end{bmatrix} \quad (73)$$

where the upper-triangular matrix $\boldsymbol{R}_1(\boldsymbol{p}_i, I, \boldsymbol{D}_i)$ has full row-rank 2 for regular features, row-rank 1 for line features, and goes toward $\boldsymbol{0}$ for uniform patches.

Considering this decomposition, the innovation term for the jth iteration step for a patch i yields

$$\boldsymbol{y}_{i,j} = \boldsymbol{Q}_1(\pi(\boldsymbol{\mu}_{i,j}^+), I, \boldsymbol{D}_i)^{\mathrm{T}}\,\boldsymbol{b}(\pi(\boldsymbol{\mu}_{i,j}^+), P_i, I, \boldsymbol{D}_i) \qquad (74)$$

This has a maximal dimension of two and loses dimensions for degenerate cases, such as line-shaped or uniform patches. Since this represents a left-multiplication with an orthonormal matrix, the noise characteristics are assumed to be of the same magnitude on every photometric error term. To account for potentially different noise properties of the intensity errors, a weighting based scheme could be introduced. The Jacobian for the innovation term is given by

$$\boldsymbol{H}_{i,j} = \boldsymbol{R}_1\left(\pi(\boldsymbol{\mu}_{i,j}^+), I, \boldsymbol{D}_i\right)\frac{\mathrm{d}\pi}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}_{i,j}^+) \qquad (75)$$

Within the iterated extended Kalman filter, the tracked landmarks are updated one after another, each undergoing a certain number of iterations. While the robocentric state formulation moved parts of the nonlinearities from the update into the propagation step, significant nonlinearities remain with the pixel intensity generation. The update iterations take care of aligning the patches in the current image and, simultaneously, spread the gained information throughout the filter state. Thus, all the landmark tracking functionality is intrinsically contained in the filter. In the case where a landmark's predicted image coordinates exhibit a large uncertainty (e.g., for newly initialized landmarks), a number of hypotheses are selected within the uncertainty bound. Figure 5 is a simplified sketch of the tracking concept. An advantage of this is that noncorner features can be properly tracked by considering the prior provided by the IMU-driven process model. In the case of line-shaped features, for instance, a corrective update only applies along the direction perpendicular to the line, while the other direction remains unaffected. In the degenerate case of uniformly textured patch features, the iteration finishes after one step without changes to the filter state (since no information is contained in the patch). Figure 6 shows the tracked landmarks in a frame. Each iteration for a landmark update is depicted by a yellow dot. For the newly added landmarks (the rightmost ones) especially, the initial uncertainty (yellow) ellipse and the number of iterations are increased.

To account for moving objects or other disturbances, a simple Mahalanobis-based outlier detection is implemented within the update step. It compares the obtained innovation residual with the predicted innovation covariance and rejects the measurement whenever the weighted norm exceeds a certain threshold. This method inherently takes into account the covariance of the state and measurements. It also considers the image gradients and thereby tends to reject gradient-less image patches more readily. In addition, a threshold on the total intensity error of a patch is introduced, where a patch measurement is rejected if the threshold is exceeded. Also, a landmark quality check is performed by sampling four nearby locations and evaluating the corresponding innovation residual. Tracking tends
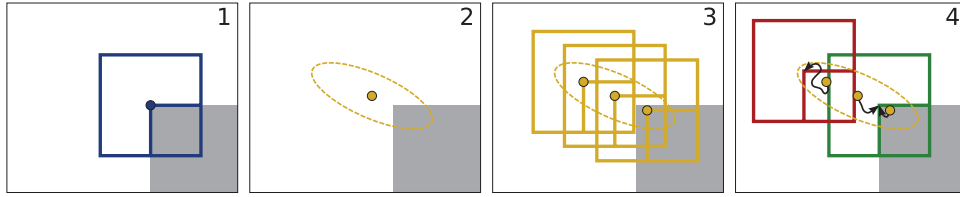
Fig. 5. Landmark tracking concept. Step 1: A patch feature (blue square) is extracted for the landmark (blue dot). Step 2: The estimated landmark image coordinates (yellow dot) and the corresponding covariance (yellow ellipse) are provided by the filter's IMU-driven process model. Step 3: Depending on the magnitude of the uncertainty, a number of candidates (yellow dots) are initialized. Step 4: For each candidate, an iterative update (black arrow) is performed, which integrates patch intensity errors together with the motion prior. Outlier detection and quality checks are performed to select the best valid tracking (green vs. red squares). Steps 3 and 4 are completely integrated in the iterative filter updates.
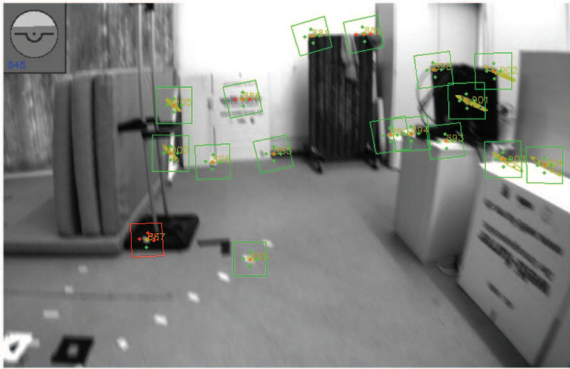

Fig. 6. Live screenshot of tracked landmarks. Projected patches (final iteration) are depicted by squares (green if successful, red if rejected). Predicted uncertainties of landmark locations are represented by yellow ellipses; each update iteration candidate is marked by a yellow dot. The final location is highlighted by a small red dot surrounded by four green or red dots. The surrounding locations are checked for higher innovation residuals (green). If more than two surrounding locations exhibit no increased innovation residuals (red), the match is rejected (e.g., the bottom-left landmark).

to be bad unless at least two locations exhibit a significantly higher residual than the matched landmark (see the bottom-left landmark in Figure 6).

The computation of the photometric error relies on an image patch from a previous frame. If parts of this image patch have influenced the filter state in the past, the resulting photometric error will exhibit a correlation with the current filter state. This correlation is not modeled in the current framework; doing so would significantly increase the computational burden (one possible approach would be to co-estimate the patch pixel intensities). This issue is also commonly encountered in dense approaches, where cross-correlations between localization and mapping are often neglected. In our case, however, the cross-correlation with the environment geometry is tracked and accounted for and the problem is limited to the texture of the environment. A refinement step on the patch intensities could reduce the pixel intensity noise and thereby reduce this effect. Investigations in this direction will be part of future work.

5.5. *Landmark management*

The iterated extended Kalman filter does not exhibit good scalability in terms of the size of the filter state. Consequently, only a limited number of landmarks can be tracked and they have to be selected and managed carefully in order to obtain good results. In Section 4.4, we outlined an intensity-based scoring which describes how informative the content of a patch can be. This is mainly used to decide what landmarks are added to the filter state. In addition to this, we maintain tracking and visibility information of a landmark, and a combined heuristic quality score is computed for each landmark being tracked. The quality score is composed of three sub-scores:

- The global quality: how often a landmark has been tracked since initialization;
- The local quality: how often a landmark has been tracked when expected to be in the field of view (limited to recent frames);
- The local visibility: how often the landmark was in the field of view (limited to recent frames).

If a landmark exhibits a high global quality, i.e., it has often been tracked since initialization, the pruning threshold on the two local sub-scores is kept more conservative. Using an adaptive thresholding, we can control the total number of landmarks that are currently in the frame, e.g., if we reach the maximal number of landmarks in the filter state and only a minor part is properly tracked, we make the landmark pruning stricter to make space for new landmarks.

6. Multi-camera setup

One issue with monocular visual-inertial setups is that they require sufficient motion in order to properly estimate the complete filter state. Also, a particular camera can be blind at times, because of fast lighting changes or very bad texture. Adding an additional camera can therefore improve the robustness of the overall system. In the case where a multi-camera setup has overlapping fields of view, several measurements of the same landmark are received at a given time. This provides information about the landmark's distance and the extrinsic calibration of the corresponding

camera frames. However, some excitation of the states is necessary to estimate the calibration parameters. Once the calibration estimates have converged, the distance of landmarks in overlapping fields of view becomes observable, even if the sensor remains stationary.

New landmarks are still detected in single camera frames only and the parametrization of the corresponding bearing vector and distance parameter is maintained with respect to the detection frame. In the case where the newly detected landmark can be seen in more than one camera frame, the initial distance estimate can be computed by triangulation.

For all subsequent time steps, the landmarks are projected in every camera frame. If the predicted pixel coordinates lie within a camera frame, an iterative update is performed (see Section 5.4). If the measurement camera frame $\mathcal{C}_m$ (where the landmark is observed) is not the same as the detection camera frame $\mathcal{C}_d$ (where the landmark is parametrized), the corresponding bearing vector must be transformed into the measurement frame first. This can be done by

$$_{\mathcal{C}_m}\boldsymbol{\mu}_i = \boldsymbol{z}_m(\boldsymbol{c}_d + \boldsymbol{z}_d^{-1}(_{\mathcal{C}_d}\boldsymbol{\mu}_i\, d(\rho_i)) - \boldsymbol{c}_m) \qquad (76)$$

where the terms of the form $\boldsymbol{c}_*$ and $\boldsymbol{z}_*$ represent the linear and rotational extrinsic IMU-camera calibrations of the corresponding cameras. Together with the parametrization of the landmark location ($_{\mathcal{C}_d}\boldsymbol{\mu}_i, \rho_i$), they are contained in the filter state. This represents the main difference to the monocular setup, where the innovation Jacobian in equation (75) must be right-multiplied by the Jacobian of equation (76).

7. Experimental results

The evaluation of the presented approach is divided into different parts: convergence evaluation (Section 7.1), parameter exploration (Section 7.2), evaluation of the photometric feedback (Section 7.3), comparison with other approaches (Section 7.4), and tests on a real micro air vehicle (Section 7.5). Whenever possible, publicly available EuRoC datasets are employed (Burri et al. (2016)). All experiments are performed with the same selection of parameters, except where explicitly mentioned. Our baseline implementation only employs the second and third image pyramid levels. While taking the first image level into account can increase accuracy, it is not really useful for highly dynamic and difficult cases. As a default, we use a patch size of 6×6, together with 25 filter landmarks.

Accuracy remains an important criterion for visual-inertial odometry and can be evaluated quantitatively. To a certain extent, it can also serve as a surrogate measure for the good functioning of an approach. To evaluate accuracy, we contemplate the root mean square estimation error per traveled distance (Geiger et al., 2012). For instance, if we want to evaluate the accuracy after 10 m of traveled distance and have a dataset that is 80 m long, we split the obtained estimation results into eight chunks of 10 m. The chunks

are then aligned with the corresponding bit of groundtruth data and the accumulated error after 10 m is evaluated. Boxplots are employed to depict the corresponding medians and quartiles. Assuming that the odometry output exhibits random walk drift with increasing traveled distance (which is often a good approximation, as long as the yaw error remains small), the observed errors should increase as the square root of the traveled distance. We select the spacing of the traveled distance samples quadratically, and should therefore observe a linear error increase in the plots. The following results can vary, depending on the setup and environment, and should not be over-interpreted.

7.1. Convergence evaluation

The proposed robocentric formulation allows undelayed initialization of landmarks but relies strongly on the proper convergence of the corresponding distance estimates (which are initialized with a very high uncertainty). While a decreasing uncertainty is desired, since it allows for more accurate tracking of the sensor pose, spurious and inconsistent convergence must be avoided. In the monocular case especially, the uncertainty should only be decreased if the sensor is moving and a sufficient baseline is acquired. To investigate the consistency of the distance estimates, a dataset was recorded for which a horizontal surface was the only visible object (the camera was directed toward the floor). The virtual groundtruth of the sensor height is inferred by averaging over the height of all converged landmarks (see Figure 7). This allows evaluation of the convergence of the landmark heights (which are strongly coupled to the distance estimates).

Figures 7 and 8 show the estimated height of the tracked landmarks over time for a monocular and a stereo setup, respectively. Since the landmarks are initialized at a fixed distance, which in this experiment tends to relate to points below the surface, a significant estimation error can be observed at initialization. Owing to the motion of the sensor, however, the height estimates quickly converge. In the top part of each figure, the estimated standard deviation (average of the estimated standard deviations) is compared against the measured standard deviation (empirical standard deviation of the actual height errors). In both cases, the estimated standard deviation encompasses the measured one. Since we averaged over many landmark tracks, this is not a strict check of consistency but still shows that the estimated covariance must lie within a reasonable range, on average. A better analysis is provided in both lower plots, which depict the normalized height error (height error divided by estimated standard deviation). In both experiments we can show that the normalized error remains below the 1% confidence threshold, i.e., there are no unreasonably large height estimate errors if compared with the corresponding uncertainty estimates.

The results of the monocular and the stereo setup are similar, owing to the significant amount of motion present in
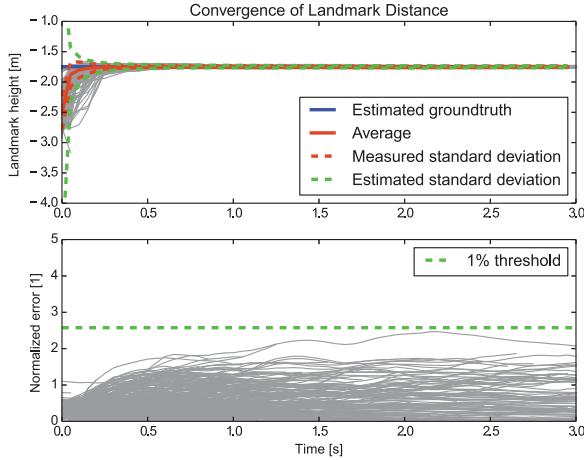
Fig. 7. Dataset observing a horizontal plane with a *monocular* setup. Estimated landmark heights (gray) together with their average (red) and empirical standard deviations (red dashes, 1-σ). The groundtruth (blue) is estimated by averaging the height estimates once converged. In the top plot, the average of the estimated standard deviation over multiple landmark heights is provided (green dashes, 1-σ). The bottom plot depicts the normalized error (height error divided by estimated standard deviation) together with the 1% confidence threshold.
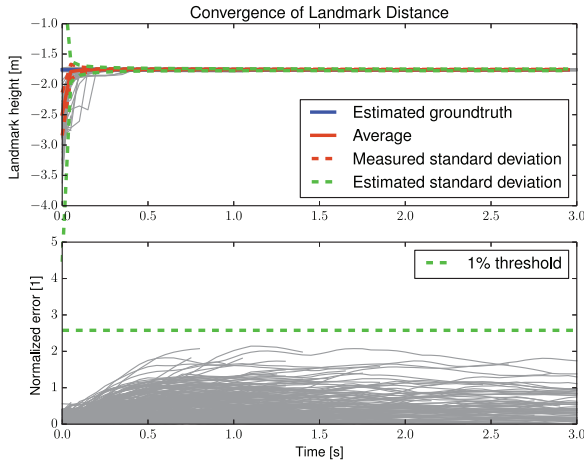

Fig. 8. Dataset observing a horizontal plane with a *stereo* setup. Stereo initialization is disabled. Estimated landmark heights (gray) together with their average (red) and empirical standard deviations (red dashes, 1-σ). The groundtruth (blue) is estimated by averaging the height estimates once converged. In the top plot, the average of the estimated standard deviation over multiple landmark heights is provided (green dashes, 1-σ). The bottom plot depicts the normalized error together with the 1% confidence threshold.

the recorded data. The final standard deviation of the height errors amounts to 0.0119 m for the monocular setup and 0.0073 m for the stereo setup.

Other parameters that must converge for a proper functioning of the filter are the online calibration parameters, which are composed of the IMU biases and the IMU-camera extrinsics. The latter should remain nearly constant
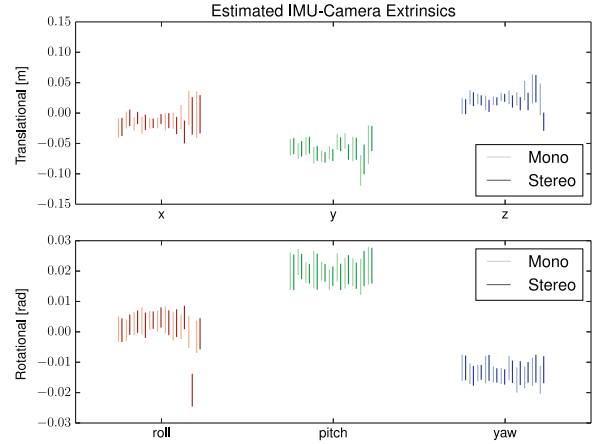

Fig. 9. Final camera-inertial measurement unit (IMU) extrinsics estimates for all 11 EuRoC datasets for monocular and stereo setups. The length of the lines corresponds to the 3-σ bounds of the estimates. The order of the lines corresponds to the datasets V1_01–03, V2_01–03, and MH_01–05.

for different datasets with the same sensor setup; we can thus evaluate the extrinsics on a number of datasets and compare the values to which they have converged. Figure 9 shows the final estimate of the rotational and translation parts of the extrinsics if running the proposed filter on all 11 EuRoC datasets. To make the task more difficult, the initial values were selected as zero translation and closest orthogonal rotation (corresponding to all zero angles in the figure). The resulting estimates, including uncertainties, seem to exhibit a large amount of accordance between the different datasets and between the monocular and stereo setups. In comparison with the first half of the datasets, the second half includes datasets with less motion which pose more difficulties for proper estimation of the extrinsics. Consequently, the estimated uncertainty (length of bar) remains larger as well. In general, the stereo setup only brings a marginal reduction of the uncertainties as long as sufficient motion is available. On the contrary, the stereo setup exhibits more difficulties for converging to the proper extrinsic calibration, as can be seen in the lower plot, where a single roll angle converges to a biased value. This is due to wrong stereo matches; a single wrong match can bias the estimation, especially if there is not sufficient motion to correct the wrong convergence.

For completeness, we also investigate the convergence of the IMU bias estimates. We only evaluate them on a single dataset (V1_03), since they exhibit intra-dataset variability. Since we have a stereo setup, we can perform two distinct monocular and one stereo evaluation with the same dataset. The results are depicted in Figures 10 and 11, where the estimate over time is plotted together with the 3-σ bounds. In particular, the gyroscope biases seem to converge very rapidly and exhibit only a very small variability. In addition, the accelerometer biases converge with sufficient excitation of the system. They typically converge faster, along
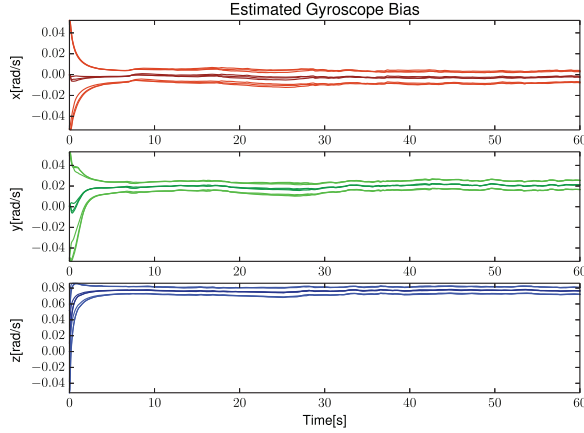
Fig. 10. Estimated gyroscope biases for dataset V1_03. Estimates (darker lines) together with 3-σ bounds (brighter lines). Results for two monocular (left and right cameras) and one stereo evaluation are depicted. The estimates converge very quickly and are less motion-dependent than the accelerometer biases.
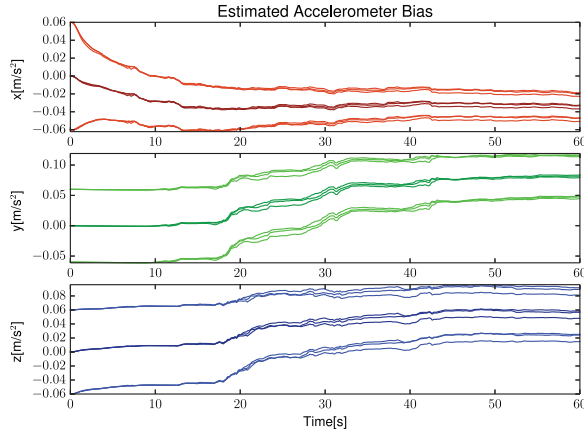

Fig. 11. Estimated accelerometer biases for dataset V1_03. Estimates (darker lines) together with 3-σ bounds (brighter lines). Results for two monocular (left and right cameras) and one stereo evaluation are depicted. The accelerometer bias converges more quickly along the gravity direction, which is mostly along the *x*-axis.

the gravity direction which is given by the *x*-axis at the beginning of the dataset.

7.2. Parameter exploration

Several important framework parameters are evaluated on the EuRoC dataset V1_03. Since the total number of landmarks in the filter state has a major influence on the computational cost of the framework, this is the first parameter that we explore. Figure 12 shows position error with respect to traveled distance for different numbers of landmarks. Surprisingly, increasing the number of landmarks does not improve the accuracy of the output once a certain amount of landmarks is reached (approximately 20). While for vision-only systems it has been shown that the
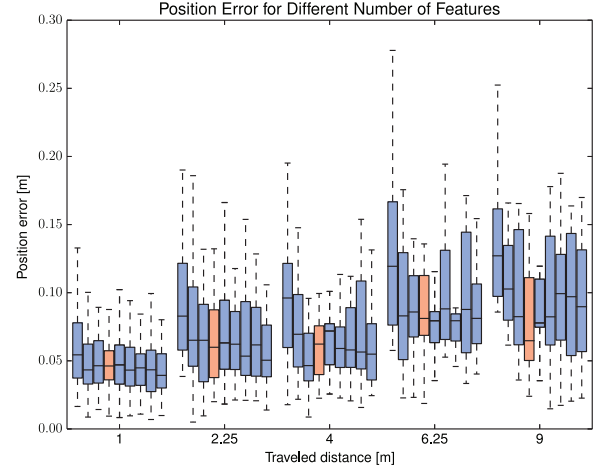

Fig. 12. Accumulated position error over traveled distance for different landmark counts. Numbers of landmarks from left to right are 10, 15, 20, 25 (red), 30, 35, 40, 45, 50. The patch size is fixed as 6. The medians and quartiles are depicted.

number of landmarks is a crucial parameter (Strasdat et al., 2010), it seems to be different for visual-inertial systems. In visual-inertial systems, the IMU provides a good prior on the motion of the systems and merely needs to be stabilized using recurrent stationary landmark observations. The number of required landmarks could depend on the quality of the employed IMU, where an IMU of lower quality would benefit more from higher landmark counts. Within this context, we also noticed a relatively strong influence of nonrejected outliers on the output's accuracy, although we selected the outlier rejection parameters to be rather strict. We noticed that properly tracking few high-quality landmarks often leads to better results than tracking many landmarks, with an increased risk of nonrejected outliers.

In our previous work (Bloesch et al., 2015), we fixed the patch size as 8×8. Here, we also investigate smaller patch, sizes since this reduces the computational load. Results for patch sizes down to 2×2 are depicted in Figure 13, which shows that we can reduce the patch size without significantly losing accuracy. Only the case with 2×2 patches exhibits notably increased errors. This could probably be tackled by increasing the number of pyramid levels, which is similar to having larger patches. All in all, we propose to employ a combination of patch size 6×6 together with 25 filter landmarks; results for these parameters are given in red in all boxplots. On a single core of an Intel®Core™ i7 at 2.4 MHz, the resulting framework uses 30–50% of the CPU load.

7.3. Photometric feedback evaluation

Photometric error feedback and its tight integration in the estimation process form a key component of the proposed framework. To assess the effect of the photometric feedback, we replace it with a traditional reprojection error based feedback combined with an explicit feature tracker.
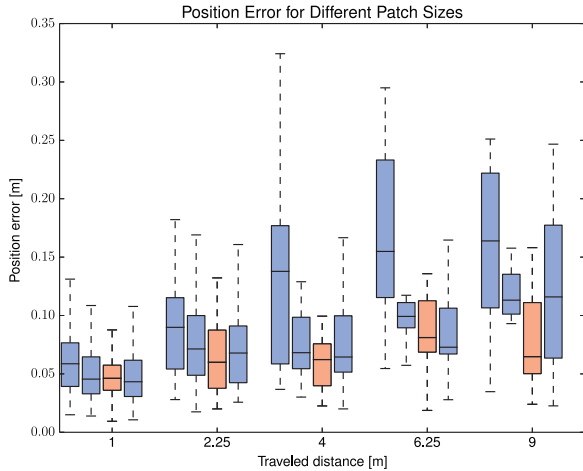
Fig. 13. Accumulated position error over traveled distance for different patch sizes. Patch sizes from left to right are 2, 4, 6 (red), and 8. The landmark count is 25. Medians and quartiles are depicted.

To this end, we implement the KLT tracker mentioned in Section 4.3 (including an initial guess from the IMU propagation and using patch warping) and try to keep all settings as similar as possible. For the reprojection error, a different measurement covariance is required, which is tuned to achieve best performance in terms of accumulated position error on the evaluation dataset (V1_03). The results are depicted in Figure 14. Two related experiments are included in this graph: a comparison with regular distance parametrization and a simple extended Kalman filter implementation (no update iterations).

An interesting observation is that the reprojection error based implementation does not lead to much larger tracking errors. This is because of the abundant corner features in the dataset, which are captured relatively well by the regular reprojection error. However, our framework does not rely on any external feature tracker; this can be seen as an advantage in terms of reduced complexity. Furthermore, in the extreme case, where no corner features are available (see Figure 15), the KLT tracker fails and the advantage of the inherent landmark tracking becomes more evident. Failing landmark tracking means that most sparse state-of-the-art visual-inertial odometry frameworks will be deprived of vision and strongly rely on an IMU-based prediction, together with an eventual relocalization. In contrast, Rovio's implicit landmark tracking is able to keep track of more subtle image regions, owing to the tight combination with inertial data, which can help to overcome visually degenerate sequences.

Finally, we can observe that the use of regular distance parametrization leads to significantly increased tracking errors. This is due to the less accurate stochastic model of the distance when compared with inverse distance parametrization and confirms previous results (Montiel et al., 2006). The extended Kalman filter implementation
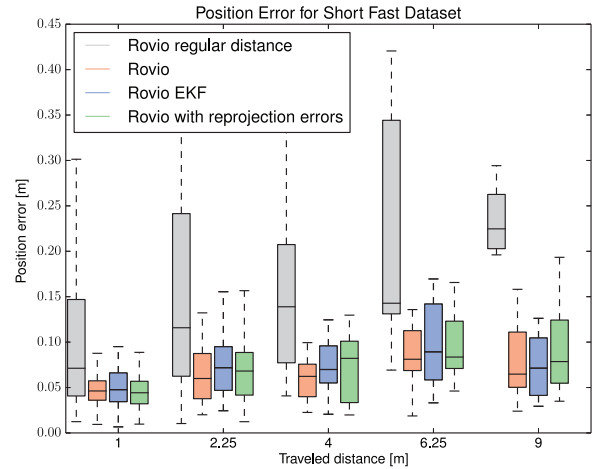

Fig. 14. Accumulated position error for different Rovio setups on the EuRoC dataset V1_03. The proposed Rovio setup (red) is compared with an implementation with regular distance parametrization (gray), an implementation without update iterations (blue), and an implementation that uses the reprojection error instead of the photometric error feedback (green). While overall the proposed setup exhibits the smallest tracking error, only the setup with the regular distance parametrization shows a significant accuracy loss.
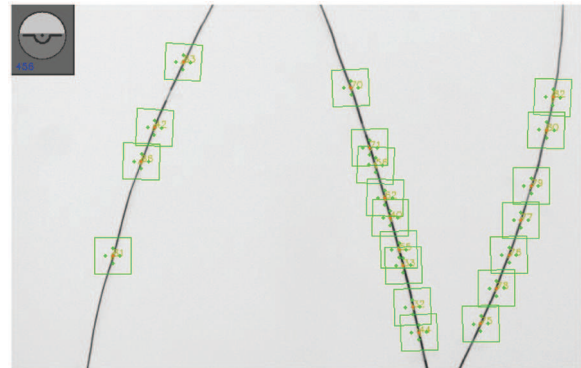EKF: extended Kalman filter.

Fig. 15. Tracking behavior without strong corner features. Snapshot taken at the end of a 20 s dataset with lines only. The inherent landmark tracking can handle such situations, owing to the additional prior it receives from the inertial-measurement-unit-driven state propagation.

exhibits only a slightly increased error metric. This indicates that a single update is often sufficient for inherent tracking. This may become more critical if the prediction of the landmark location is less accurate, such as when the initial landmark distance estimate is bad or in cases with high linear velocities.

7.4. Comparison with related work

Figure 16 compares both the monocular and stereo Rovio setups against the stereo Okvis framework on the EuRoC dataset V1_03. Okvis is the open-source release version
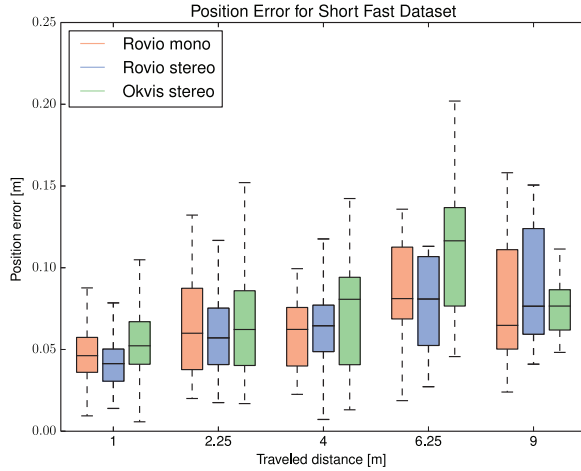
Fig. 16. Accumulated position error over traveled distance for a monocular and a stereo Rovio setup, as well as for Okvis. Red: standard monocular Rovio (6×6 patches, 25 landmarks). Blue: standard stereo Rovio (6×6 patches, 25 landmarks). Green: stereo Okvis with online parameter estimation. All frameworks exhibit similar tracking errors.

of the work of Leutenegger et al. (2015) and relies on a windowed bundle adjustment approach, which includes inertial measurements. Owing to the sufficient motion and good texture of the dataset, monocular and stereo setups exhibit very similar tracking errors. For this dataset, the performances of Rovio and Okvis are comparable and show that our approach can compete with state-of-the-art visual-inertial frameworks.

The accuracy of the presented approach was also evaluated on the 1.4 km long circular dataset employed by Forster et al. (2017), Leutenegger et al. (2015), and Usenko et al. (2016). Figures 17 and 18 show the position error and the yaw error over traveled distance for the standard monocular and stereo Rovio setups as well as for Okvis. The performance of Rovio is slightly inferior to Okvis over 360 m of traveled distance. Again, the performance is strongly dependent on the selected tuning parameters, which were kept constant for all experiments. Forster et al. (2017) and Usenko et al. (2016) both provide results that show 0.3 m position error after 360 m. Caution should be taken when interpreting these results, since only three nonoverlapping segments of length 360 m are contained in the 1.4 km long dataset (low statistical significance). Rovio seems to perform better for shorter distances and shows similar performance to all other visual-inertial frameworks, especially for a traveled distance of 10 m ($n = 140$), where it exhibits a median error of less than 0.1 m. One reason for the decreased long-term performance can be found in the increased yaw error, which has a strong impact on the position performance over longer distances.

We observed that this error could be further reduced by choosing stricter outlier rejection parameters and including the first pyramid level in the residual computation (below
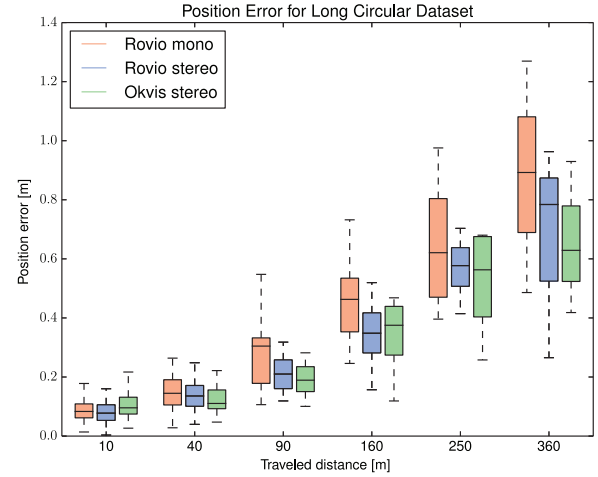

Fig. 17. Accumulated position error over traveled distance for the long circular dataset. Red: baseline monocular Rovio. Blue: baseline stereo Rovio. Green: stereo Okvis with online parameter estimation.

Fig. 18. Accumulated yaw error over traveled distance for the long circular dataset. Red: baseline monocular Rovio. Blue: baseline stereo Rovio. Green: stereo Okvis with online parameter estimation.
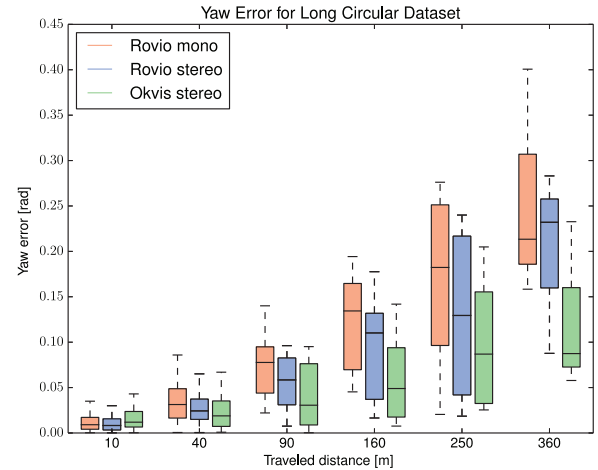
0.4 m for 360 m). In the end, however, large-scale accuracy should be provided by an enclosing back-end system performing loop closures and relocalization, rather than by over-tuning the front-end visual-inertial odometry at the cost of increased computational costs and lower overall robustness.

7.5. Robust micro air vehicle control

In this final evaluation section, we investigate the applicability of Rovio for feedback control on a micro air vehicle for fast aggressive flights under bad lighting conditions and motion blur. The system is initialized on the ground and remains stationary for 30 s. After take-off, it performs three fast circular loops before landing at the same location. The trajectory's position and attitude are depicted in
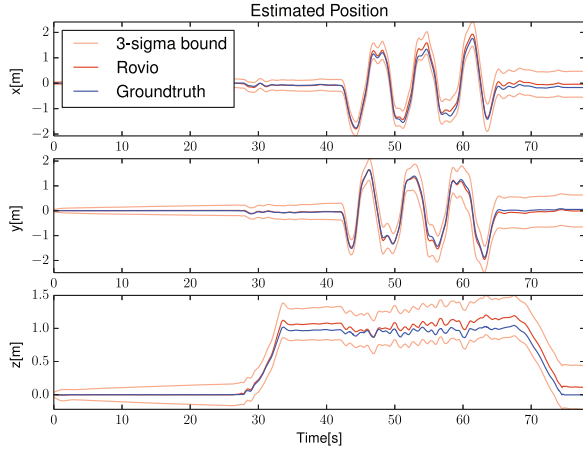
Fig. 19. Estimated micro air vehicle position for aggressive flight. Red: estimated position. Light-red: 3-σ bounds. Blue: Vicon groundtruth. The unobservable position accumulates uncertainty over time.
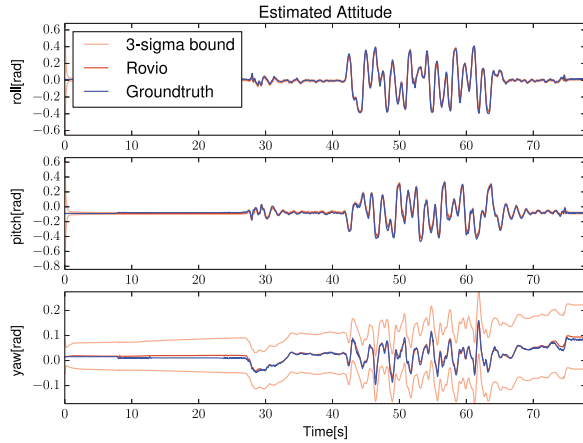

Fig. 20. Estimated micro air vehicle attitude for aggressive flight. Yaw–pitch–roll decomposition is employed to separate the unobservable yaw from the two inclination angles (only visualization). Red: estimated attitude. Light-red: 3-σ bounds. Blue: Vicon groundtruth.

Figures 19 and 20, respectively. The 3-σ bounds for the estimates are also plotted. The observable roll and pitch angles very quickly converge from their initially large uncertainties and accurately track the micro air vehicle's inclination angles after take-off. The global yaw angle and positions, however, accumulate uncertainty over time, confirming the inherent unobservability of those states. All in all, the experiment shows that Rovio can handle fast motions and difficult scenes while providing reliable state estimation for feedback control of an autonomous micro air vehicle.

Figure 21 shows the robocentric velocity of the micro air vehicle. Owing to the robocentric formulation of our filter, the observable states are entirely decoupled from the unobservable states. Hence, the uncertainties are bounded and the estimation error remains minimal, which is essential for
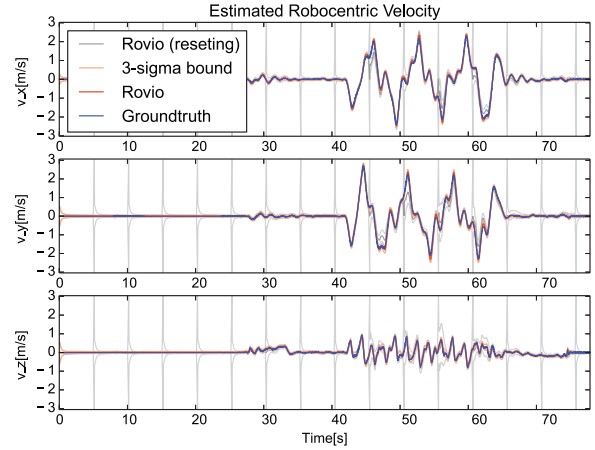

Fig. 21. Estimated micro air vehicle velocity for aggressive flight. Red: Estimated velocity. Light-red: 3-σ bounds. Blue: Vicon groundtruth. Gray: Estimated velocity when resetting Rovio every 5 s. Since the velocity is expressed in the robocentric inertial measurement unit coordinate frame, it is fully observable and the uncertainty remains bounded.

feedback control. Additionally, velocity estimates are provided where Rovio was reset every 5 s. The estimates very quickly converge to the true velocities for all resets. This highlights the very simple and robust initialization of our robocentric filtering approach, where ego-motion estimates are immediately available.

8. Conclusion

This paper presented a framework based on an iterated extended Kalman filter, which tightly fuses inertial measurements and image data. The originality and strength of the proposed approach lie in its fully robocentric formulation combined with direct feedback of photometric errors as the Kalman innovation term. This leads to a more robust implementation, since the observable states are not influenced by increasing global covariance. We introduce an iterative update scheme, which inherently takes care of landmark tracking. While simplifying the structure of the overall framework, data association is rendered more robust by the tight coupling with the IMU-driven process model. The employed minimal representations of rotations and bearing vectors improve the numerical consistency of the approach and reduce the computational effort. The extensive experimental evaluation shows that the presented approach can compete with state-of-the-art visual-inertial fusion techniques. Interestingly, our approach achieves comparable ego-motion estimation accuracy with a significantly lower landmark count. Robustness with respect to fast motions and bad lightning conditions, as well as the instantaneous initialization procedure, were demonstrated in a real autonomous micro air vehicle flight experiment. Additional features, such as optional GPS measurements, are included in the updated version of the publicly available

open-source software package (https://github.com/ethz-asl/rovio).

Funding

References

Bell B and Cathey F (1993) The iterated Kalman filter update as a Gauss–Newton method. *IEEE Transactions on Automatic Control* 38(2): 1991–1994.

Bloesch M, Gehring C, Fankhauser P, et al. (2013) State estimation for legged robots on unstable and slippery terrain. In: *IEEE/RSJ international conference on intelligent robots and systems*, Tokyo, Japan, 3–7 November 2013, pp. 6058–6064. Piscataway, NJ: IEEE.

Bloesch M, Omari S, Hutter M, et al. (2015) Robust visual inertial odometry using a direct EKF-based approach. In: *IEEE/RSJ international conference on intelligent robots and systems*, Hamburg, Germany, 28 September–2 October 2015, pp. 298–304. Piscataway, NJ: IEEE.

Bloesch M, Sommer H, Laidlow T, et al. (2016) A primer on the differential calculus of 3D orientations. *arXiv* 1606.05285

Burri M, Nikolic J, Gohl P, et al. (2016) The EuRoC micro aerial vehicle datasets. *International Journal of Robotics Research* 35(10): 1157–1163.

Castellanos JA, Neira J, and Tardos JD (2004) Limits to the consistency of EKF-based SLAM. *IFAC Proceedings Volumes* 37(8): 716–721.

Civera J, Grasa OG, Davison AJ, et al. (2009) 1-point RANSAC for EKF-based structure from motion. In: *IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, 10–15 October 2009, pp. 3498–3504. Piscataway, NJ: IEEE.

Davison AJ (2003) Real-time simultaneous localisation and mapping with a single camera. In: *IEEE international conference on computer vision*, Nice, France, 13–16 October 2003, vol. 2, pp. 1403–1410. Piscataway, NJ: IEEE.

Engel J, Schöps T, and Cremers D (2014) LSD-SLAM: Large-scale direct monocular SLAM. In: Fleet D, Pajdla T, Schiele B, et al. (eds.) *Computer Vision—ECCV 2014* (*Lecture Notes in Computer Science*, vol. 8690). Cham: Springer, pp. 834–849.

Forster C, Carlone L, Dellaert F, et al. (2017) On-manifold preintegration theory for fast and accurate visual–inertial navigation. *IEEE Transactions on Robotics* 33(1): 1–21.

Forster C, Pizzoli M, and Scaramuzza D (2014) SVO: Fast semi-direct monocular visual odometry. In: *IEEE international conference on robotics and automation*, Hong Kong, China, 31 May–7 June 2014. Piscataway, NJ: IEEE.

Geiger A, Lenz P, and Urtasun R (2012) Are we ready for autonomous driving? The KITTI vision benchmark suite, In: *IEEE conference on computer vision and pattern recognition*, Providence, RI, 16–21 June 2012. Piscataway, NJ: IEEE.

Hertzberg C, Wagner R, Frese U, et al. (2011) Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion* 14(1): 57–77.

Huang GP, Mourikis AI, and Roumeliotis SI (2008) Analysis and improvement of the consistency of extended Kalman filter based SLAM. In: *IEEE international conference on robotics and automation*, Pasadena, CA, 19–23 May 2008, pp. 473–479. Piscataway, NJ: IEEE.

Jin H, Favaro P, and Soatto S (2003) A semi-direct approach to structure from motion. *The Visual Computer* 19(6): 377–394.

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

Julier SJ and Uhlmann JK (2001) A counter example to the theory of simultaneous localization and map building. In: *IEEE international conference on robotics and automation*, Seoul, South Korea, 21–26 May 2001, pp. 4238–4243. Piscataway, NJ: IEEE.

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

Leutenegger S, Lynen S, Bosse M, et al. (2015) Keyframe-based visual–inertial odometry using nonlinear optimization. *International Journal of Robotics Research* 34(3): 314–334.

Li M and Mourikis AI (2013) High-precision, consistent EKF-based visual-inertial odometry. *International Journal of Robotics Research* 32(6): 690–711.

Lucas BD and Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: *Proceedings of the 7th international joint conference on artificial intelligence*, Vancouver, Canada, 24–28 August 1981, vol. 2, pp. 674–679. San Francisco, CA: Morgan Kaufmann.

Lynen S, Sattler T, Bosse M, et al. (2015) Get out of my lab: Large-scale, real-time visual-inertial localization. In: *Robotics: Science and systems* (eds. LE Kavraki, D Hsu, and J Buchli), Rome, Italy, 13–17 July 2015. Cambridge, MA: MIT Press.

Ma J, Bajracharya M, Susca S, et al. (2016) Real-time pose estimation of a dynamic quadruped in GPS-denied environments for 24-hour operation. *International Journal of Robotics Research* 35(6): 631–653.

Molton N, Davison A, and Reid I (2004) Locally planar patch features for real-time structure from motion. In: *Proceedings of the British machine vision conference* (eds. A Hoppe, S. Barman, and T Ellis), London, UK, 7–9 September 2004, pp. 90.1–90.10. Durham: British Machine Vision Association.

Montiel J, Civera J, and Davison A (2006) Unified inverse depth parametrization for monocular SLAM. In: *Proceedings of robotics: Science and systems* (eds. AS Sukhatme, S Schaal, W Burgard, et al.), Philadelphia, PA, 16–19 August 2006, pp. 16–19. Cambridge, MA: MIT Press.

Mourikis AI and Roumeliotis SI (2007) A multi-state constraint Kalman filter for vision-aided inertial navigation. In: *IEEE international conference on robotics and automation*, Rome, Italy, 10–14 April 2007, pp. 3565–3572. Piscataway, NJ: IEEE.

Rosten E and Drummond T (2006) Machine learning for high speed corner detection. In: Leonardis A, Bischof H, and Pinz A (eds.) *Computer Vision—ECCV 2006* (*Lecture Notes in Computer Science*, vol. 3951). Berlin: Springer, 430–443.

Shen S, Mulgaonkar Y, Michael N, et al. (2014) Initialization-free monocular visual-inertial estimation with application to autonomous MAVs. In: Hsieh M, Khatib O, and Kumar V (eds.) *Experimental Robotics* (*Springer Tracts in Advanced Robotics*, vol. 109). Cham: Springer, pp. 211–227.

Shi J and Tomasi C (1994) Good features to track. In: *IEEE Computer Society conference on computer vision and pattern recognition*, Seattle, WA, 21–23 June 1994, pp. 593–600. Piscataway, NJ: IEEE.

Silveira G, Malis E, and Rives P (2008) An efficient direct approach to visual SLAM. *IEEE Transactions on Robotics* 24(5): 969–979.

Solà J, Vidal-Calleja T, Civera J, et al. (2012) Impact of landmark parametrization on monocular EKF-SLAM with points and lines. *International Journal of Computer Vision* 97: 339–368.

Stelzer A, Hirschmuller H, and Gorner M (2012) Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain. *International Journal of Robotics Research* 31(4): 381–402.

Strasdat H, Montiel JMM, and Davison AJ (2010) Real-time monocular SLAM: Why filter? In: *IEEE international conference on robotics and automation*, Anchorage, AK, 3–7 May 2010, pp. 2657–2664. Piscataway, NJ: IEEE.

Tanskanen P, Naegeli T, Pollefeys M, et al. (2015) Semi-direct EKF-based monocular visual-inertial odometry. In: *IEEE/RSJ international conference on intelligent robots and systems*, Hamburg, Germany, 28 September–2 October 2015, pp. 6073–6078. Piscataway, NJ: IEEE.

Usenko V, Engel J, Stückler J, et al. (2016) Direct visual-inertial odometry with stereo cameras. In: *IEEE international conference on robotics and automation*, Stockholm, Sweden, 16–21 May 2016, pp. 1885–1892. Piscataway, NJ: IEEE.

Weiss S, Achtelik M, Lynen S, et al. (2013) Monocular vision for long-term micro aerial vehicle state estimation: A compendium. *Journal of Field Robotics* 30(5): 803–831.

Appendix

Bearing vector calculus

Assuming a stationary 3D point landmark $\mathcal{F}$ with bearing vector $\boldsymbol{\mu}$ and distance parameter ρ, the corresponding differential equations can be obtained by totally differentiating the kinematics

$$\frac{\mathrm{d}}{\mathrm{d}t}\left\{ {}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{IF}} = {}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{IC}} + \boldsymbol{q}_{C\mathcal{I}}^{-1}(\boldsymbol{\mu}\, d(\rho)) \right\} \tag{77}$$

For this we require the following partial differentials

$$\frac{\mathrm{d}}{\mathrm{d}t}({}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{IC}}) = {}_{\mathcal{I}}\boldsymbol{v}_{\mathcal{C}} \tag{78}$$

$$\frac{\partial}{\partial \boldsymbol{q}_{C\mathcal{I}}}(\boldsymbol{q}_{C\mathcal{I}}^{-1}(\boldsymbol{\mu}\, d(\rho))) = -\boldsymbol{q}_{C\mathcal{I}}^{-1}(\boldsymbol{\mu}\, d(\rho))^{\times}\, \boldsymbol{C}(\boldsymbol{q}_{C\mathcal{I}}^{-1}) \tag{79}$$

$$= -\boldsymbol{C}(\boldsymbol{q}_{C\mathcal{I}}^{-1})\,\boldsymbol{\mu}^{\times} d(\rho) \tag{80}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{q}_{C\mathcal{I}} = \boldsymbol{\omega}_{\mathcal{C}} \tag{81}$$

$$\frac{\partial}{\partial \boldsymbol{\mu}}(\boldsymbol{q}_{C\mathcal{I}}^{-1}(\boldsymbol{\mu}\, d(\rho))) = \boldsymbol{C}(\boldsymbol{q}_{C\mathcal{I}}^{-1})\,\boldsymbol{\mu}^{\times}\boldsymbol{N}(\boldsymbol{\mu})\, d(\rho) \tag{82}$$

$$\frac{\partial}{\partial \rho}(\boldsymbol{q}_{C\mathcal{I}}^{-1}(\boldsymbol{\mu}\, d(\rho))) = \boldsymbol{C}(\boldsymbol{q}_{C\mathcal{I}}^{-1})\,\boldsymbol{\mu}\, d'(\rho) \tag{83}$$

In equation (79), we use the chain rule together with equations (14) and (15); in equation (81), we directly employed equation (13); and equation (82) relies on equation (32). The total differential of equation (77) can be evaluated and simplified to (left-multiplication with $\boldsymbol{C}(\boldsymbol{q}_{C\mathcal{I}})$)

$$0 = {}_{\mathcal{I}}\boldsymbol{v}_{\mathcal{C}} - \boldsymbol{C}(\boldsymbol{q}_{C\mathcal{I}}^{-1})\,\boldsymbol{\mu}^{\times}\,\boldsymbol{\omega}_{\mathcal{C}} d(\rho) \tag{84}$$
$$+ \boldsymbol{C}(\boldsymbol{q}_{C\mathcal{I}}^{-1})\left(\boldsymbol{\mu}^{\times}\boldsymbol{N}(\boldsymbol{\mu})\,\dot{\boldsymbol{\mu}}\, d(\rho) + \boldsymbol{\mu}\, d'(\rho)\,\dot{\rho}\right)$$

$$0 = \boldsymbol{v}_{\mathcal{C}} - \boldsymbol{\mu}^{\times}\boldsymbol{\omega}_{\mathcal{C}}\, d(\rho) \tag{85}$$
$$+ \boldsymbol{\mu}^{\times}\boldsymbol{N}(\boldsymbol{\mu})\,\dot{\boldsymbol{\mu}}\, d(\rho) + \boldsymbol{\mu}\, d'(\rho)\,\dot{\rho}$$

From this, the dynamics for the bearing vector and distance parameter can be obtained by pre-multiplication with $1/d(\rho)\,\boldsymbol{N}(\boldsymbol{\mu})^{\mathrm{T}}\,\boldsymbol{\mu}^{\times}$ and $1/d'(\rho)\,\boldsymbol{\mu}^{\mathrm{T}}$, respectively

$$\dot{\boldsymbol{\mu}} = \boldsymbol{N}(\boldsymbol{\mu})^{\mathrm{T}}\left(\hat{\boldsymbol{\omega}}_{\mathcal{C}} + \boldsymbol{n}(\boldsymbol{\mu})^{\times}\,\frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho)}\right) + \boldsymbol{w}_{\mu} \tag{86}$$

$$\dot{\rho} = -\boldsymbol{n}(\boldsymbol{\mu})^{\mathrm{T}}\,\hat{\boldsymbol{v}}_{\mathcal{C}}/d'(\rho) + w_{\rho} \tag{87}$$

Here, we used the identities $\boldsymbol{N}(\boldsymbol{\mu})^{\mathrm{T}}\,\boldsymbol{\mu}^{\times}\boldsymbol{\mu}^{\times} = -\boldsymbol{N}(\boldsymbol{\mu})^{\mathrm{T}}$ and $\boldsymbol{N}(\boldsymbol{\mu})^{\mathrm{T}}\,\boldsymbol{N}(\boldsymbol{\mu}) = \boldsymbol{I}$. Also, some additive process noise has been added.

Applying the Euler forward discretization scheme to the continuous time differential equation of the bearing vectors (equation 67) yields

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k \boxplus \Delta t\left(\boldsymbol{N}(\boldsymbol{\mu}_k)^{\mathrm{T}}\left(\hat{\boldsymbol{\omega}}_{\mathcal{C}} + \boldsymbol{n}(\boldsymbol{\mu}_k)^{\times}\,\frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho_k)}\right) + \boldsymbol{w}_{\mu}\right)$$

$$\boldsymbol{\mu}_{k+1} = \exp\left(\Delta t \boldsymbol{N}(\boldsymbol{\mu}_k)\left(\boldsymbol{N}(\boldsymbol{\mu}_k)^{\mathrm{T}}\left(\hat{\boldsymbol{\omega}}_{\mathcal{C}}\right.\right.\right.$$
$$\left.\left.\left. + \boldsymbol{n}(\boldsymbol{\mu}_k)^{\times}\,\frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho_k)}\right) + \boldsymbol{w}_{\mu}\right)\right) \otimes \boldsymbol{\mu}_k$$

$$\boldsymbol{\mu}_{k+1} = \exp\left(\Delta t\left(\left(\boldsymbol{I} - \boldsymbol{n}(\boldsymbol{\mu}_k)\,\boldsymbol{n}(\boldsymbol{\mu}_k)^{\mathrm{T}}\right)\hat{\boldsymbol{\omega}}_{\mathcal{C}} + \boldsymbol{n}(\boldsymbol{\mu}_k)^{\times}\,\frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho_k)}\right.\right.$$
$$\left.\left. + \boldsymbol{N}(\boldsymbol{\mu}_k)\,\boldsymbol{w}_{\mu}\right)\right) \otimes \boldsymbol{\mu}_k \tag{88}$$

Here, we applied the definition of boxplus (equation 25) and used the identity $\boldsymbol{N}(\boldsymbol{\mu})\,\boldsymbol{N}(\boldsymbol{\mu})^{\mathrm{T}} = \boldsymbol{I} - \boldsymbol{n}(\boldsymbol{\mu})\,\boldsymbol{n}(\boldsymbol{\mu})^{\mathrm{T}}$. The three components influencing the bearing vector prediction can be observed here: the perpendicular part of the rotational rate, the linear velocity weighted by the inverse distance, and the additive noise.