Monocular Vision for Long-term Micro Aerial Vehicle State Estimation: A Compendium

Stephan Weiss, Markus W. Achtelik, and Simon Lynen
Autonomous Systems Lab, ETH Zurich
Michael C. Achtelik
*Ascending Technologies GmbH, Germany**
Laurent Kneip, Margarita Chli, and Roland Siegwart
Autonomous Systems Lab, ETH Zurich

Received 12 September 2012; accepted 13 May 2013

The recent technological advances in Micro Aerial Vehicles (MAVs) have triggered great interest in the robotics community, as their deployability in missions of surveillance and reconnaissance has now become a realistic prospect. The state of the art, however, still lacks solutions that can work for a long duration in large, unknown, and GPS-denied environments. Here, we present our visual pipeline and MAV state-estimation framework, which uses feeds from a monocular camera and an Inertial Measurement Unit (IMU) to achieve real-time and onboard autonomous flight in general and realistic scenarios. The challenge lies in dealing with the power and weight restrictions onboard a MAV while providing the robustness necessary in real and long-term missions. This article provides a concise summary of our work on achieving the first onboard vision-based power-on-and-go system for autonomous MAV flights. We discuss our insights on the lessons learned throughout the different stages of this research, from the conception of the idea to the thorough theoretical analysis of the proposed framework and, finally, the real-world implementation and deployment. Looking into the onboard estimation of monocular visual odometry, the sensor fusion strategy, the state estimation and self-calibration of the system, and finally some implementation issues, the reader is guided through the different modules comprising our framework. The validity and power of this framework are illustrated via a comprehensive set of experiments in a large outdoor mission, demonstrating successful operation over flights of more than 360 m trajectory and 70 m altitude change.[1]

1. INTRODUCTION

The unique combination of the agility and the small size of micro helicopters often renders them the only choice for deployment in areas inaccessible to humans or other robotic platforms. Enabling navigation through small openings (e.g., windows, pipes), but more importantly in disaster scenarios in which navigation is necessary in partially collapsed buildings with potentially trapped victims, micro helicopters can act as a far superior aid to rescue workers than ground vehicles or fixed-wing airplanes.

While Micro Aerial Vehicles (MAVs) have advantageous properties over other platforms, these come at the cost of a series of interlinked challenges in the design and algorithmic setup for autonomous navigation. First and foremost, the *limited payload* dictates a careful selection of all onboard equipment for a flight to become possible. This translates into low battery capacity and, thus, *low power consumption* for the processing board and the sensors. Conversely, a heavier platform would need more powerful propulsion to fly—indicatively, every 10 grams require roughly 1 W of lifting power in hover mode of a commercial multicopter system. It is evident, however, that even with a meticulous selection of components, the autonomy time of the MAV is bound by its battery life.

The weight and power restrictions not only imply *limited calculation power* and, thus, the need for efficient algorithms, but they also have a direct impact on the choice of sensors. The pursuit of light and low-power sensors, which are able to provide rich information at the same time, has

*The research leading to this article has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreements no. 231855 (www.sfly.org), no. 266470 (www.mycopter.eu), and no. 285417 (www.fp7-icarus.eu).
Stephan Weiss is technologist at NASA-JPL/CalTech (email: stephan.weiss@ieee.org). Markus W. Achtelik, Simon Lynen and Laurent Kneip are currently Ph.D. students at the ETH Zurich (email: "markus.achtelik, simon.lynen, laurent.kneip" @mavt.ethz.ch). Michael C. Achtelik is CEO of Ascending Technologies GmbH (email: michael.achtelik@asctec.de). Margarita Chli is a senior researcher at and deputy director of the Autonomous Systems Lab (ASL) at ETH Zurich (email: margarita.chli@mavt.ethz.ch). Roland Siegwart is full professor at the ETH Zurich and head of the ASL (email: r.siegwart@ieee.org).
[1]Video material can be found at www.youtube.com/watch?v=vHpw8zc7-JQ
Direct correspondence to:
Stephan Weiss, e-mail: stephan.weiss@ieee.org

Figure 1. The "Firefly" hexacopter by Ascending Technologies, with a mount of a 1.6 GHz single core ATOM computer and a MatrixVision "Bluefox" wide VGA camera.

been subject to research for some time in the MAV community. Inspired by several examples in nature and for the sake of general applicability, here we aim for a compact, *minimal sensor suite* providing visual and inertial cues. The key idea of this work is to demonstrate how this information can be used in a complementary manner to achieve an efficient and robust scene and motion estimation.

While ground robots by definition maintain a certain proximity to the scene under consideration, airborne vehicles have a very flexible range of navigation, rendering distance-measuring sensors or stereo-camera setups unsuitable.[2] As a result, this work studies flights using a single, monocular camera and an Inertial Measurement Unit (IMU). For a truly applicable framework, we eliminate any reliance on global information sensors (e.g., GPS), which are often unavailable or unreliable. In turn, this brings the interlinked algorithmic challenge of still maintaining a gravity-aligned MAV navigation frame without global information.

As an overall contribution, this work provides the MAV research community with a concise compendium on how to successfully approach the issue of monovision-based flights for MAVs in large GPS-denied environments and long-term missions. More precisely, as a first contribution of this work, we present a high-performing Visual Odometry (VO) framework, which runs with constant computational complexity onboard a MAV and exhibits robustness in natural and self-similar environments. The second contribution is the theoretical development of self-calibrating the sensors onboard the MAV and, subsequently, a sensor-fusion methodology. These contributions comprise a real-time *power-on-and-go system* maintaining a gravity-aligned navigation frame in long-term missions and in large environments, with all computation running entirely onboard the MAV. The MAV used in our analysis and experimentation is depicted in Figure 1, but it should be noted that the presented framework is general and, thus, not specifically tied to this platform—it can be

[2]Stereo-camera processing is limited by scene depth, e.g., flying far out of the scene (common with MAVs), stereo essentially reduces to mono.

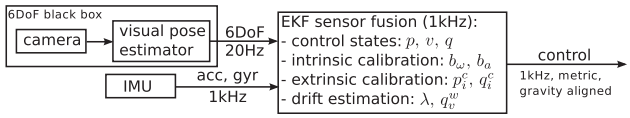


Figure 2. Schematic of our proposed framework. The only sensors used are an IMU (at 1 kHz measuring acceleration and angular velocity) and a camera (20 Hz on an ATOM 1.6 GHz processor). The camera is loosely coupled and can be seen as a black box sensor yielding an arbitrarily scaled six-degree-of-freedom pose. We describe a real-time method to compute this six-degree-of-freedom pose onboard the platform. The proposed sensor fusion not only provides the MAV control states (position p, velocity v, attitude q) but also calibrates the IMU intrinsic states (acceleration bias b_a, gyroscope bias b_ω), the extrinsic parameters between IMU and camera (translation p_i^c, rotation q_i^c), and the drifts in the visual estimate (scale factor λ, attitude drift q_v^w) yielding a gravity-aligned, metric MAV pose control at 1 kHz for a self-calibrating power-on-and-go system.

used onboard any vehicle with a camera and an IMU. As a final contribution, we conducted a wide range of experiments to test the behavior and correct functioning of our proposed approach. We demonstrate the practical applicability of the system in a real scenario, setting up a variety of challenges in order to reveal the power and limitations of the proposed system. In these experiments, we use a downward-looking camera setup since this seems to be most appropriate when flying at a mostly obstacle-free height outdoors. However, our approach is not limited to this particular configuration thanks to its self-calibrating aspect. In fact, it allows the camera (and IMU) to be placed completely arbitrarily on the vehicle. Figure 2 shows a rough overview of the different parts in our framework.

Bearing the generality of our framework in mind, in this work we refer to *long-term* as a (theoretically) infinite period and we refer to *large* as (theoretically) infinite in size. These two attributes require specific capabilities of the algorithms in our framework. First, *long-term* requires the state estimation algorithm to provide a gravity-aligned navigation frame and to self-calibrate the vehicle at all times despite potential drifts in the sensors or in the visual framework. Only this allows the MAV to stay airborne long-term. Second, *large* requires all algorithms to run at constant computational complexity. Our practical tests are limited to the battery lifetime on the MAV. Even during this relatively short time (about 8 min, 360 m trajectory), we will illustrate the importance of estimating a gravity-aligned navigation frame and of the constant computational complexity.

2. CURRENT STATE OF RESEARCH

The field of autonomous MAV navigation, despite being very young, has been receiving growing interest and advancing very rapidly over the past couple of years. The

level of autonomy currently possible, however, is far from what is necessary for realistic deployment, as existing works struggle greatly with long-term missions and operation in general environments, i.e., large in size, unknown, heavily cluttered, and potentially GPS-denied areas. It is only after we have addressed these issues successfully that higher level tasks such as autonomous exploration, swarm navigation, and large-scale trajectory planning can be tackled as part of real missions.

A key problem with airborne vehicles is the continuous stabilization and control in six degrees of freedom (DOF), i.e., attitude and position control. In our previous works [Weiss et al. (2011) and Blösch et al. (2010)], we discussed this issue of stabilization and control in detail using a linear quadratic Gaussian/loop transfer recovery (LQG/LTR) controller. In Achtelik et al. (2011), we used Simultaneous Localization and Mapping (SLAM) pose feeds in a Luenberger observer for MAV control. These works focus on control, presenting solutions for outdoor, vision-based MAV navigation and onboard computation. However, for the sake of simplicity, the latter does not take into account any visual drifts, which inevitably occur in any real system, nor does it have constant computational complexity. For these reasons, the approach does not fulfill our requirements for *long-term* missions in *large* environments. In Weiss et al. (2011) and Blösch et al. (2010), we proposed a method to align the navigation frame with gravity by using hover phases and the accelerometer readings. This approach is theoretically unclean since we did not take any biases of the IMU into account, nor can we guarantee hover phases in any situation. Both lead also in practice to a wrong alignment of the navigation frame with gravity. In this work, we specifically focus on accurate, scalable vision-based state estimation and full system self-calibration continuously during the mission, eliminating any unrealistic assumptions. Continuous self-calibration of the system is essential for handling drifts and changes in the sensor-suite (e.g., IMU bias drifts and extrinsic transformations between sensors) and the vision system (e.g., map and scale drifts) in order to keep the MAV airborne in long-term missions. Estimating the system's extrinsic parameters online during the mission additionally avoids the need for tedious premission calibration procedures. We assume the camera's intrinsic parameters to be fixed and thus they are calibrated only once. While our previous work in Weiss (2012) aims in a similar direction for multisensor fusion including GPS and magnetometer, the present work is a succinct reference work for pure visual-inertial systems in GPS-denied areas. We analyze in detail the specific visual pipeline used, the system state estimation, and the robustness of our framework both analytically and in real-world tests.

Successful results without reliance on GPS have recently been achieved using laser range finders in (Achtelik et al., 2009; Bachrach et al., 2009a, 2009b). In the latter two works, Bachrach et al. used a Hokuyo laser scanner and 2D SLAM for autonomous navigation in a maze, winning the international competition of MAVs (IMAV) in a challenge consisting of navigating through a maze. Unlike cameras, laser range finders have the advantage of providing useful cues in textureless environments. Although this is a very appealing aspect of range finders, their restricted range of perception, limited field of view (typically only within a plane), and most importantly their high weight and power demand render them unsuitable for use onboard our MAV setup. Our camera-IMU setup weighs about 20 g, and, as demonstrated in our experiments, this provides locally sufficient stability for real-time, onboard control of the MAV without the need for artificial landmarks.

One way to address the motion-estimation problem for MAV navigation is by installing a set of external cameras with a known location to track the MAV, covering the entire workspace. While this is a very time- and cost-demanding setup involving a lengthy installation process, and it is unrealistic in large outdoor areas, it can be used to successfully track the MAV with great precision, as done in Altug et al. (2002), Park et al. (2005), Klose et al. (2010). In recent years, the robotics literature has seen some impressive works using the motion capture system from Vicon[3]: a system of high-resolution external cameras able to track the 6 DOF pose of one or more vehicles with submillimeter accuracy (How et al., 2008; Lupashin et al., 2010; Michael, Mellinger, Lindsey, and Kumar, 2010; Valenti et al., 2006). Advanced airborne control systems have been demonstrated to perform aggressive maneuvers with multiple flips of the MAV (Lupashin et al., 2010), docking to an oblique wall (Mellinger, Michael, and Kumar, 2010) and cooperative grasping of objects (Mellinger, Shomin, Michael, and Kumar, 2010; Michael, Fink, and Kumar, 2010) using the Vicon system. Thus, they are limited to facilities with the appropriate installation.

2.1. Camera as a Motion Sensor

Egomotion estimation using an onboard camera can be approached by tracking fixed landmarks of known appearance and full three-dimensional (3D) position (e.g., artificial markers or user-specified points). Following the establishment of correspondences with the current view, the problem of position estimation essentially reduces to relocalization with respect to the known landmarks (Proctor and Johnson, 2004; Wu et al., 2013). Hamel et al. (2002) implemented a visual trajectory tracking method to control a MAV with an onboard camera observing n fixed points. A similar approach was developed by Cheviron et al. (2007), who used IMU cues in addition to visual data. Another possibility is to estimate the position of the MAV by tracking landmarks on a leading MAV, maintaining a fixed relative position and orientation. Chen and Dawson (2006) implemented

[3]http://www.vicon.com

this by tracking coplanar points on the leading vehicle and using homography to estimate the relative pose. Wenzel et al. (2010), on the other hand, used four light sources on a ground robot and homography estimation to perform autonomous take-off, tracking, and landing on the moving ground robot.

While the aforementioned approaches require some knowledge of the scene, user intervention, or modification of the workspace, motion estimation can be performed in a far more general manner; instead of relying on known features, distinctive natural landmarks can be extracted at run-time from the scene [e.g., SURF (Bay et al., 2008) and BRISK (Leutenegger et al., 2011)]. The 3D position of these landmarks is not known *a priori*, but the optical flow or the relative position to each other and to the camera can be estimated as done in visual SLAM and visual odometry. While this approach provides a more general solution, it poses more estimation challenges onboard an aerial vehicle, and as a result, far fewer works appear in the literature that use natural landmarks to control a helicopter. Apart of our previous work (Weiss et al., 2012b), known optical flow approaches either use an additional sensor (i.e., sonar) to retrieve metric information for control (Fraundorfer et al., 2012) or require delicate parameter tuning for each specific situation (Herisse et al., 2012). Common to all optical flow approaches is their velocity-based control and their requirement of additional algorithms for position-based navigation in large environments. Engel et al. (2012) proposed such an additional algorithm by using a pressure sensor and a camera to build a metric map in which the vehicle can navigate.

Schmid et al. (2012) used a stereo camera to perform odometry for MAV control. Stereo approaches require the scene to be within a certain distance for precise feature triangulation. A notable monocular work is that of Artieda et al. (2009), who implement visual SLAM using an extended Kalman filter (EKF) for Unmanned Aerial Vehicles (UAVs), but their visual SLAM approach is used only for mapping the scene and not for controlling the UAV. The opposite was done in Brockers et al. (2012), where the authors used PTAM in a premapped area for controlling the MAV on an autonomous landing-site detection and landing task. The work that is probably closest to the proposed framework is that of Ahrens et al. (2009). Based on the monocular EKF-SLAM approach of Davison et al. (2007), they built a localization and mapping framework that provides an almost drift-free pose estimate. With that, they implemented a position controller and obstacle avoidance. However, due to the simplification they used to speed up their feature-tracking algorithm, a non-negligible drift persists. Their choice of an EKF-based visual SLAM algorithm is particularly sensitive to outliers and requires high computation power for large environments (arising by either long exploration or local but dense workspaces). In fact, long-term flights in large environments are computationally infeasible with such an approach as the complexity of the estimation scales quadratically with the number of features in the map.

The study in Strasdat et al. (2010) illustrates the advantage of using multiple features rather than multiple frames for robust pose estimation, advocating the use of key-frame-based SLAM over filtering-based approaches in most scenarios. Following this evidence, the present work demonstrates how a self-calibrating navigation system can be decoupled from the computationally expensive EKF-based visual pose estimation, which, as a result, allows the employment of a key-frame-based approach instead.

Driven by the need for computationally feasible, onboard motion estimation for long-term missions, one of the contributions of this work is to demonstrate how we can render the camera onboard the MAV a real-time 6 DOF pose sensor. In contrast to existing (visual) motion estimation approaches, we do not require any *a priori* information on the environment, nor do we need external aids such as an external tracking system in order to obtain reliable information for MAV control. We not only aim at using the onboard camera as a bearing sensor, but also as a *real-time*[4] and *onboard motion* sensor. Furthermore, as reported in Ahrens et al. (2009), drift impedes long-term navigation in large environments, and as sensor calibration during long missions might change, they can give rise to further drifts. Thus, we aim at understanding in detail the different sources of drift and address them by continuous sensor calibration in-flight during the entire mission in order to build a true *power-on-and-go system*. While we have recently introduced our approach to this problem in Weiss et al. (2011), the framework presented here achieves constant computational complexity and enhanced robustness in highly challenging, self-similar (outdoor) environments with significantly increased computation speed.

2.2. Drifts and Sensor Self-Calibration

Fusing IMU data with a vision sensor is a well-studied topic in the literature. However, most works in the literature focus either on the calibration of these two sensors or on the pose estimate of the robot assuming known calibration parameters. As analyzed in Corke et al. (2007), the approaches of combining inertial and visual measurements for absolute scale structure and motion estimation can be categorized into *loosely coupled* and *tightly coupled*. The loosely coupled philosophy treats the inertial and visual units as two separate modules running at different rates and exchanging information, while the tightly coupled paradigm combines both sources of information into a single, optimal filter.

[4]The real-time capability of the camera as a *bearing* sensor is mainly given by its frame rate. For *motion estimation*, the images undergo computationally complex tasks, and the state of the art still lacks methods of providing real-time general camera motion estimates on computationally constrained platforms in large environments.

Among the loosely coupled approaches are the works of Corke (2004), Armesto et al. (2004, 2007), Gemeiner et al. (2007), Roumeliotis et al. (2002), Mourikis et al. (2009), and Weiss and Siegwart (2011), while among the tightly coupled ones are those of Strelow and Singh (2003), Chroust and Vincze (2004), Huster et al. (2002), Qian et al. (2002), Baldwin et al. (2009), Lupton and Sukkarieh (2008), Lupton and Sukkarieh (2009), Kelly and Sukhatme (2011), Jones (2009), and Jones and Soatto (2011).

Sterlow has presented real-time visual-inertial navigation using an iterated EKF in Strelow (2004), however the complexity of this approach grows at least quadratically with the number of features, and is thus unsuitable for large-scale navigation. More efficient methods are presented in Roumeliotis et al. (2002), Mourikis et al. (2009), and Mourikis and Roumeliotis (2007), which consider pairwise images for visual odometry and fuse the output afterward with inertial measurements in an EKF. In these works, the authors additionally use an altimeter for solving the unknown scale of the visual estimates, presenting an EKF approach that is linear in the number of features and quadratic in the number of included camera poses. All of these approaches, however, do not estimate the intersensor calibration, but rather assume it is known and fixed. Furthermore, they are tightly coupled approaches using directly the visual features as measurements, leading to high computational complexity of the algorithms. The proposed framework includes the inter sensor states in the estimation, which helps to eliminate sources of significant drift, while it follows the loosely coupled strategy, directly taking the estimated 6 DOF camera pose as measurement. This immediately renders the choice of the visual pipeline algorithm independent from the rest of the estimation, providing seamless interchangeability with any readily available 6 DOF camera pose estimation solution.

More closely related to the present work, is the work in Pinies et al. (2007), Jones (2009), and in particular Mirzaei and Roumeliotis (2008) and Kelly and Sukhatme (2011). Mirzaei and Roumeliotis (2008), presented a Kalman filter (KF) -based calibration solution to estimate rotation and translation between a camera and an inertial sensor mounted on a rigid body, retrieving the monocular scale factor from a feature pattern with known dimensions. This work is the solid foundation for both Kelly and Sukhatme (2011) and the work presented here. Kelly and Sukhatme (2011) extended the calibration method to operate when observing an *a priori* unknown static structure. This allows us to estimate the calibration between IMU and the camera without any additional equipment. In fact, since their work estimates the gravity vector in the vision frame,[5] it is a complete visual-inertial, self-calibrating EKF-based state-estimation system. Their approach allows us to continuously recover the angular offsets in the roll and pitch of the visual-inertial system with respect to an earth-fixed reference frame, even if the initial anchor landmarks are not visible anymore. The underlying visual EKF SLAM approach, however, impedes the use on MAVs in large-scale environments due to the rapidly increasing computational complexity, which quickly becomes infeasible for onboard processing.

3. FROM VISUAL SLAM TO ONBOARD VISUAL ODOMETRY

Our approach is an adaptation of the *key-frame*-based visual SLAM algorithm PTAM Klein and Murray (2007), which, as shown in Strasdat et al. (2010), is both computationally less expensive and more robust than filter-based visual SLAM implementations. When the camera is visiting a previously mapped area, only tracking of the current frame within the current map has to be performed, thus it is only the addition of new key-frames to the map (i.e., exploring new environment) that is computationally critical. The tracking and the mapping threads operate on the original image as well as on down-sampled versions of it, the so-called "pyramid levels." On each key-frame creation, features extracted on different levels are introduced to the map. A pyramid level prediction procedure in the tracking thread searches for the known features appearing in the map in the predicted pyramid level of the current image. If a feature is extracted at level 1 (the first down-sampled image) and added to the map, it will be searched for in level 0 (original image) if the camera moves away from it. The full SLAM framework of PTAM is not constant in computational complexity and is as such not a good choice for long-term navigation tasks. More recent work like that of Strasdat et al. (2011) presents a solution of constant complexity while still keeping the benefits of SLAM (i.e., loop closure). The authors report an algorithmic speed of 17 Hz on a 2.66 GHz dual core processor.

In the following, we detail the series of modifications we performed to the original PTAM based on a rigorous analysis and show their influence on performance. We achieve a solution that is robust in self-similar outdoor scenes and runs at 20 Hz on a 1.6 GHz single core processor (Figure 3). We ran several tests in a Vicon motion capture system that provides ground truth for the MAV position. Apart from the algorithmic speed, there are three distinct behaviors we are interested in when using the estimated pose as input for a MAV controller: the visual scale drift, the map or pose drift, and any localization failures. Common to all drifts (in scale, position, and attitude) is their spatial nature: while observing the same features, the visual pose estimate does not suffer from any drifts. For MAVs, this is

[5]We refer to the *vision frame* as the frame with respect to which the camera pose is estimated in the black-box vision framework. Kelly and Sukhatme (2011) refer to this frame as a *world frame*, which they clearly state is *not* a gravity-aligned frame. In this work, we refer to a *world frame* as a gravity-aligned frame unless otherwise stated.

Figure 3. Our improved monocular SLAM pipeline in action during a large-scale outdoor flight. The left image shows the trajectory flown so far (BingMaps). The green circle marks the current position. The middle image shows the camera image with features extracted on different pyramid levels. Note the leafless tree in the right part of the image. The right image shows the map built, using the last 15 key frames (the max. no. retained key frames at any instant). The bold tripod marks the current pose. The angular offset with respect to a gravity-aligned world frame is particularly visible. This shows the importance of estimating this visual drift for long-term vision-based MAV navigation (see Section 4).

the case in hover mode or when performing movements with only a bounded action radius, maintaining visibility of some anchor features.

3.1. Key-frame Handling

In PTAM, the map is defined as a set of key-frames together with their observed features. To render the computational complexity constant, we set a maximum number of key frames retained in the map at any instant. If this number is exceeded, the key-frame farthest away (in Euclidean distance) from the current MAV pose gets deleted along with the features associated with it. If the maximum number of retained key-frames is infinite, then the algorithm becomes equivalent to the original PTAM. Limiting the number of key-frames leads to a banded structure of the block-bundle-adjustment matrix. The required Cholesky decomposition usually has $O(n^3)$ complexity (with n being the dimension of the matrix being linear in points and key-frames), but for banded structures this can be as low as $O(n^2)$.

We define the drift of the visual framework to be a transform between the visual and world frames including a changing bias P_{bias} and the initial frame misalignment $\bar{P}_w^v$. The current misalignment between these two frames (i.e., the drift) is then $P_w^v = \bar{P}_w^v P_{\text{bias}}$, where $P_i^j = [R\ T]$ is a transform in rotation R and translation T from frame i to frame j. We analyze the drift behavior of P_w^v for different numbers of retained key-frames. Considering Sibley et al. (2009), we expect an asymptotic improvement in accuracy approaching the original PTAM implementation the more key-frames we keep in the estimation. Compared to Sibley et al. (2009), we avoid the selection of the map region (linear complexity with respect to the map size) by deleting past key-frames altogether. This leads to constant computational complexity.

Figure 4 shows the test trajectory flown in 3D. In the test runs, we flew at a height of about 1.2 m and created a key-frame roughly every 14 cm of traversed trajectory. The

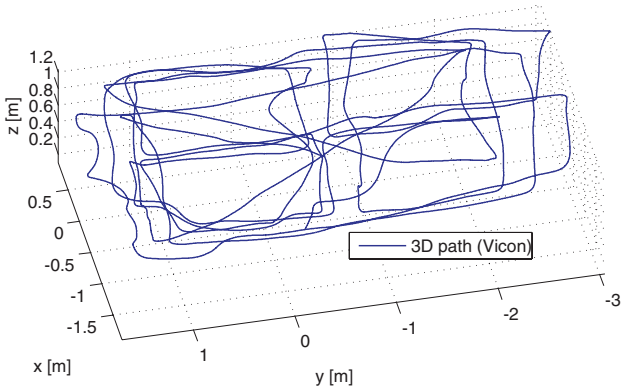


Figure 4. 3D trajectory of the path flown. Eleven colinear key-frames in the xy-plane cover roughly a distance of 1.5 m at a height of about 1.2 m.

test area was flat, Vicon-monitored for ground truth, and well-textured, with the flown trajectory containing multiple loop-closures. The same data were processed using different maximum key-frame thresholds (3, 7, and 11), as well as using the original PTAM. The loops can be detected in the original PTAM, but they have been defined to be sufficiently large such that the runs with a limited number of key-frames do not profit from them. Note that, at a height of 1.2 m, 11 collinear key frames only cover a distance of about 1.5 m in this setup.

Figure 5 illustrates the attitude drift between the vision frame and the world frame. The attitude drift is calculated as the delta-rotations between data and ground truth, following alignment of the initial attitude to ground truth. Although all cases show similar attitude drift behavior, the ones retaining fewer key-frames have the tendency to drift slightly faster. The yaw drifts are the slowest. This may be due to the coupling of roll and pitch with the uncertain feature depth estimation and the coupling of yaw with the more certain xy feature position.

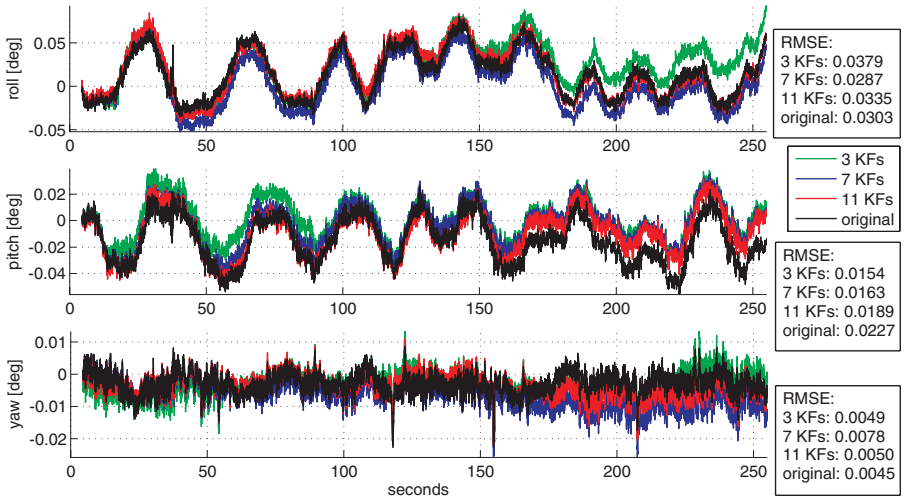


Figure 5. Attitude drifts for a different number of retained key-frames. Note that the plots are displayed in different scales. We observe that even the original code has large variations in roll and pitch drifts. This arises from an imperfect camera calibration. The fact that the imperfect camera calibration dominates the angular drifts shows the importance of having a self-calibrating system taking into account the visual drifts.

Even for the original framework, we observe large variations in the roll, smaller variations in the pitch, and almost no variations in the yaw drift. The variations in roll and pitch clearly dominate the drift behavior and arise from an imperfect camera calibration: the reconstruction of the (flat) area becomes a curved surface with its extremum in the point where the visual framework was initialized (starting point). The features continue to be perceived perpendicular to the camera z-axis (local consistency), but since the reconstruction is bent the global attitude of the camera is affected. Figure 6(a) depicts the situation schematically. This error due to imperfect intrinsic camera calibration is zero at the starting point and increases the farther away the camera moves. In addition to this systematic error, the system really drifts in attitude if we retain a limited number of key-frames. Hence, in Figure 5 the black curve (original PTAM, nondrifting due to loop closure capability) only shows the error arising from the imperfect camera calibration, and the remaining curves in Figure 5 show the superposition of this error together with the attitude drift. In the experiment, the moved distance is larger in the camera roll direction (y-axis in Figure 4), thus the variations in roll in Figure 5 are larger than in pitch. The pure calibration error is best visualized in Figure 6(b), where we compare the camera distance to the origin with the attitude error in the original PTAM. We normalized the roll and y-position and the pitch and x-position each with their maximum values, and we superimposed the graphs. The discussed correlation between the distance to the starting point (where the visual framework was initialized) and the error in roll and pitch is clearly visible. Since this effect is much more accentuated than the actual visual drift, it is the primary reason a state estimator needs to be able to track such changes to maintain a gravity-aligned navigation frame (see Section 4).

To isolate the pure position drift, as shown in Figure 7, we unrotate every MAV pose sample by its respective attitude drift and only then compare its position with ground truth. The figure shows the dependency of the drift on the number of retained key-frames: as expected, the *gain* in performance decreases with more key frames. The position drift in the original framework arises again from an imperfect camera calibration. While tests with limited key frames and visual odometry character show increasing drifts (best visible in z), the original code with SLAM character stays bounded around zero drift.

At each time-instant, we measure the median distance of features to the camera center. Since all features lie in the xy-plane of the ground truth frame, the ratio of this distance over the ground truth altitude yields a measure of the visual scale. As illustrated in Figure 8, the improvement in scale drift with more key-frames is clearly visible. Even for very low drifts, in long-term missions it is important for a state estimator to track these changes in scale in order to maintain metric commands to an underlying controller (see Section 4).

3.2. Improved Feature Handling for More Robust Maps

To achieve robust camera pose tracking and generate a consistent map, it is important to select high-quality features in each frame. Outdoors, self-similarity of the environment in high spatial frequency is an omnipresent issue, e.g., the asphalt in urban areas or the grass in rural areas. As the

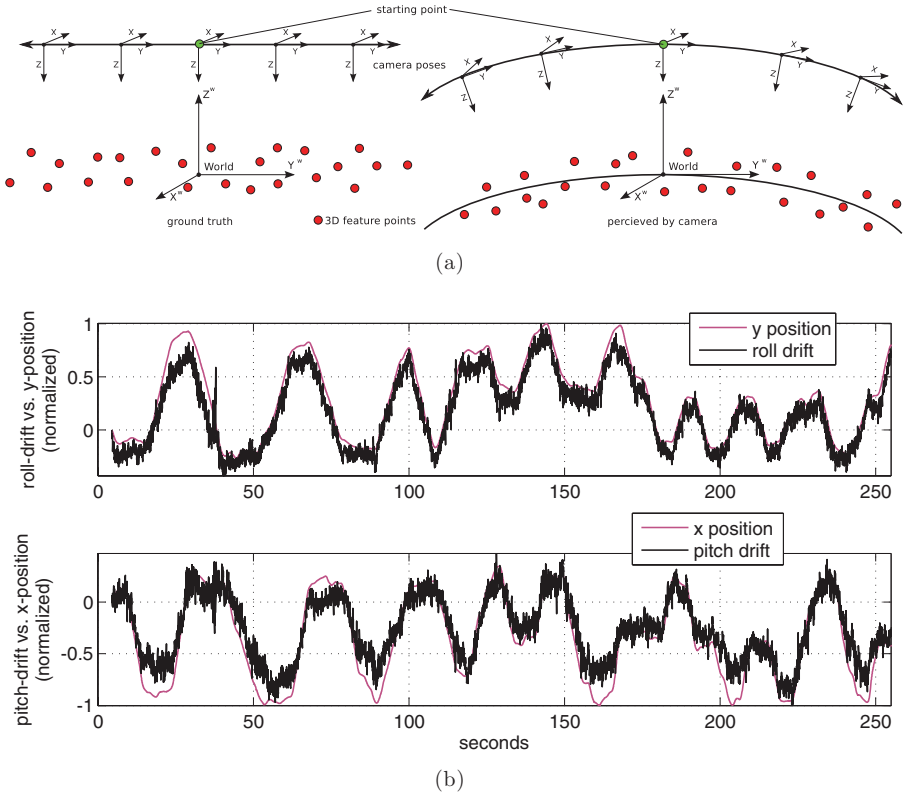


Figure 6. (a) Schematic presentation of the induced attitude error because of a wrong camera calibration. The flat area (left) is wrongly reconstructed as a bent surface (right). This induces a larger attitude error the farther the camera moves away from the starting (i.e., initialization) point. (b) Correlation between camera position (Vicon ground truth) and angular errors in roll and pitch (using the original visual framework) showing the effect of a wrong camera calibration on real data. The graphs are normalized with respect to their maximal value. No yaw motion was applied during this test.

camera resolution increases, more of these high-frequency self-similar structures get captured. Conversely, low-resolution cameras blur these structures in a similar way to a low-pass filter. A similar effect takes place in the down-sampled pyramid versions of a high-resolution image. Features in higher pyramid levels represent salient regions in lower spatial frequencies, whereas features in lower levels represent salient regions in high spatial frequencies, as depicted in Figure 9 schematically.

Low-frequency self-similarity is already handled well by the camera motion-model by limiting the search-radius for corresponding features [red circle in Figure 9(b)]. In contrast, the search-radius contains several matching candidates on high-frequency self-similar structure [Figure 9(a)]. Following this rationale, we only include features extracted in the higher pyramid levels for the map-building process. This directly avoids issues with high-frequency self-similar structures. To verify this intuition, we analyzed the feature quality per pyramid level, monitoring the percentage of outliers in the newly added features upon each key-frame creation. The recorded outlier percentages are visualized in Figure 10, while Figure 10(c) shows a typical camera image of our outdoor dataset, manifesting the issue of high-frequency self-similarity. The majority of outliers (96.04%) occur in the finest pyramid level [Figure 10(a)]. In fact, only about one of every three features in this level gets triangulated correctly. The inlier/outlier ratio is drastically improved in higher levels [Figure 10(b)]. This comes from the motion model accuracy versus self-similarity frequency discussed in Figure 9. The number of lower pyramid levels that should be discarded for the map-building process depends on the resolution of the original camera image and the frequency of the self-similarity in the mission area. By analyzing the inlier/outlier ratio as we showed in Figure 10(b), we can define the cutoff pyramid level in a sound and efficient manner. This could also be done adaptively during the mission.

It is important to discuss the features in the finest pyramid level. While proven unreliable during the map-building process, their importance in tracking is crucial; they have a direct effect on the tracking quality when moving away from a given feature. For wide-angle cameras, tracking at

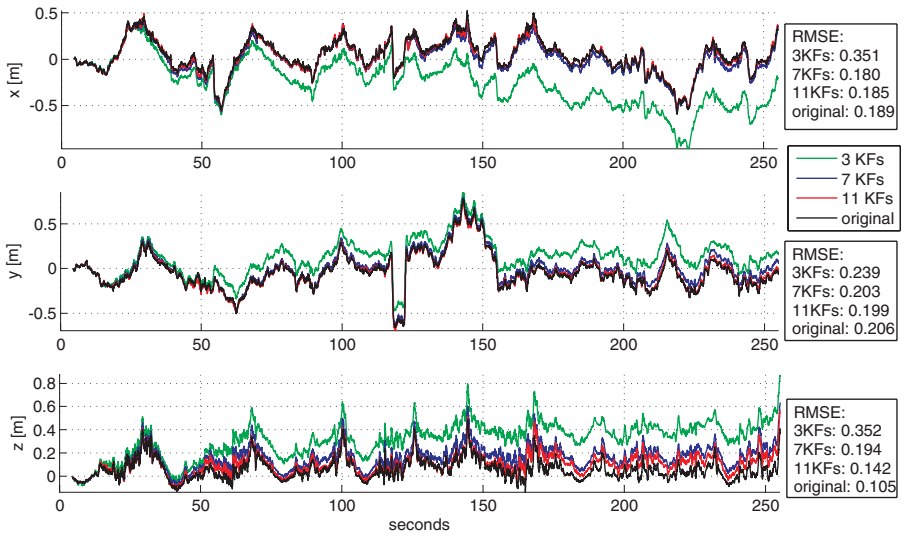


Figure 7. Position drifts for a different number of retained key-frames. Even the original framework shows some variations in the drift states due to an imperfect camera calibration, however these variations are bounded around zero due to loop closure techniques. Limiting the number of key-frames induces noticeable drifts. At around $t = 120$ s, the plots show a wrong data association.

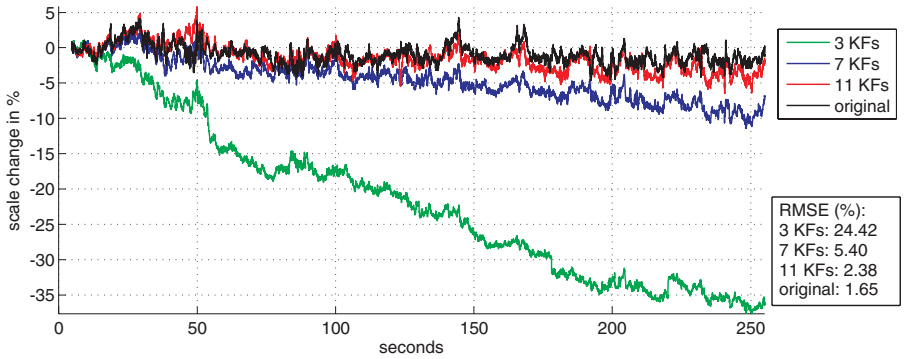
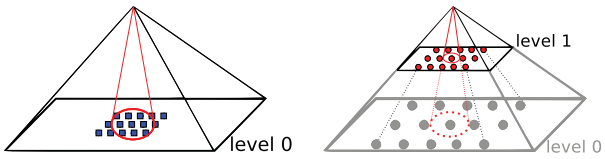


Figure 8. Scale drift versus number of retained key-frames.

the finest level provides the ability to track peripheral features. These two situations are shown schematically in Figure 11. As a result, here we take advantage of the decoupled mapping and tracking processes of the PTAM framework and we still use these features during tracking. Figure 11(c) depicts a real outdoor image captured with a wide-angle camera and demonstrates the case of Figure 11(b). The blue dots in the top left image area are previously added map points in higher pyramid levels tracked in the finest pyramid level of the current frame. The MAV explores the area from top left to bottom right. A robust estimator mitigates the influence of wrong data associations in tracking. A few wrong correspondences may lead to some local jitter in the pose estimate, but these wrong associations for tracking are not stored in the map avoiding any map inconsistencies.

To verify the gain in robustness, we evaluated two test runs over self-similar outdoor terrain. We compared our modified algorithm with respect to the original PTAM keeping all key-frames and with the original PTAM with a limited number of key-frames. While the first comparison also includes the issue of computational complexity, the latter comparison isolates the gain in robustness given by our improved feature selection method. Figure 12 shows the comparisons when manually flying over a cobblestone road along the world (negative) x-axis. The original PTAM loses track at about $t = 32$ s. Because it keeps all key-frames, it relocalizes again at about $t = 41$ s. This is more visible at $t = 58$ s until $t = 76$ s. Then, at $t = 76$ s the map is too corrupted and the system fails. Only retaining five key-frames in the original PTAM versions leads to complete failure at



(a) (b)

Figure 9. The relation of the images stored in the key-frame's pyramid and the image frequencies captured. In (a), features on high-frequency self-similar structures are extracted in the lowest pyramid level, since most detail is preserved in this level. The search radius on this level arising from the camera motion-model "cone" (red circle) covers several potential matches, rendering wrong data associations more likely. In (b) the down-sampling of the original image (e.g., to the first pyramid level) acts as a low-pass filter and eliminates high-frequency self-similar structures. Only features of low-frequency self-similar structure get extracted, easing any ambiguous data associations following the application of the motion model.

$t = 21$ s already. Or modified version runs robustly through the whole dataset using only five key-frames.

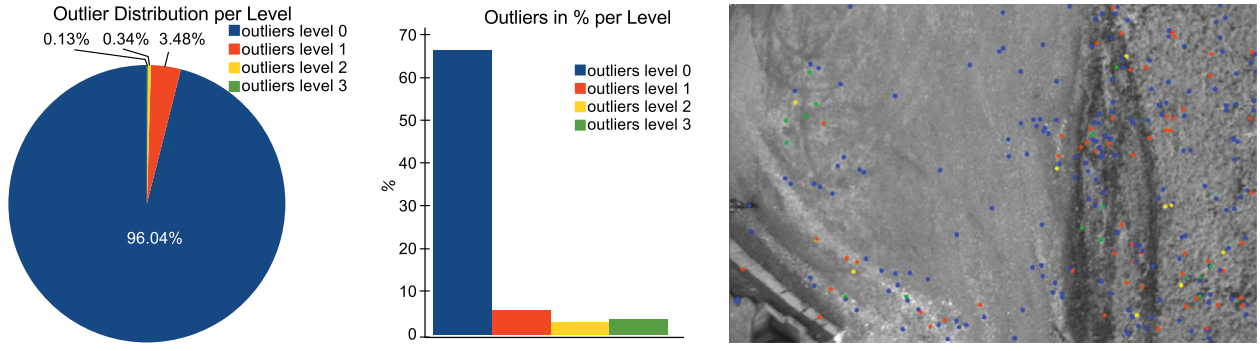
Figure 13 shows a similar behavior for a test above grass. The performance gain is even more obvious since the blades of grass that are visible in the finest pyramid level move constantly under the down-wash of the MAV and provide poor map features in the original PTAM version. Our improved version only uses these moving features for tracking, which may introduce some jitter in the current pose estimation, but they do not corrupt the map.

Figure 14(c) shows the timings for key-frame creation. We see that the most time-demanding parts are on a per-feature base: Shi-Tomasi score computation, FAST nonmax suppression, and the addition of new features to the map. Neglecting the lowest level (yielding the most features with the least reliable information) drastically reduces the number of features to be treated while creating a new key-frame. Thus, it directly reduces the computational cost and improves the map quality.

This results in great speed-ups with key-frame creation running at 13 Hz (in contrast to the 7 Hz of the original PTAM) and normal tracking rates of around 20 Hz on an onboard Atom single core computer 1.6 GHz. These timings are calculated using a maximum of five key-frames for the map representation in both the original and modified versions.

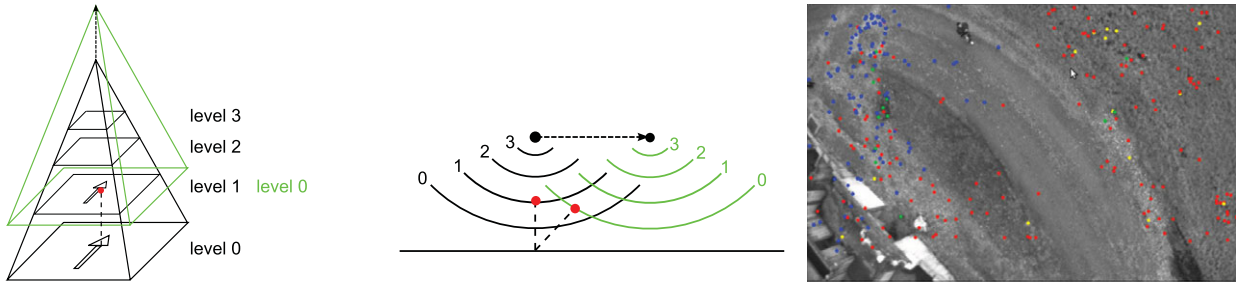
3.3. Reinitialization After Failure Mode

We need to recognize if the visual algorithm fails (see Section 4.2) and reinitialize it as fast as possible. For automatic initialization, we ensure that the baseline is sufficiently large by calculating the rotation-compensated median pixel disparity. For rotation compensation, we use efficient second-order minimization techniques (ESM) Malis (2004) in order to keep PTAM independent of IMU readings. This keeps our approach modular to other 6 DOF pose estimators. For reinitializations, we store the median scene depth and pose of the closest, last valid frame and propagate this information to the new initialized map. This way we minimize large jumps in scale and pose after reinitialization. In Figure 15, we actively initiated two reinitialization steps while hovering. The scale and position jumps after the first reinitialization at $t = 50$ s are barely noticeable. At the second reinitialization step at $t = 72$ s, we notice that a slight scale



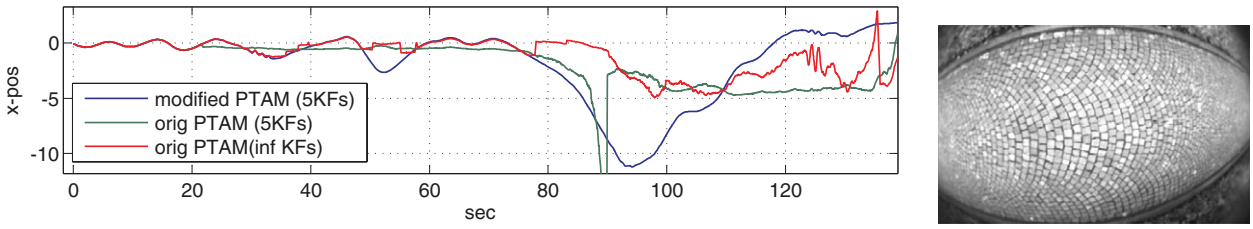
(a) (b) (c)

Figure 10. (a) In total, about 45% of the features triangulated during a new key-frame addition are flagged as outliers in the subsequent bundle adjustment step. 96.04% of these outliers are in the lowest pyramidal level. (b) The normalized outliers with respect to the number of triangulated features in each level, demonstrating that also in relative terms, the features in the lowest level are the least reliable ones. (c) A typical scene of an outdoor dataset [color coding corresponds to different levels and is consistent with that in (a) and (b)]. The high-frequency self-similarity is responsible for failed triangulations in the lowest pyramid level. Lower-frequency self-similarity is handled well by the camera motion-model and the resulting restricted search-areas. Thus, features in higher pyramid levels are more robust. This figure is best viewed in color.



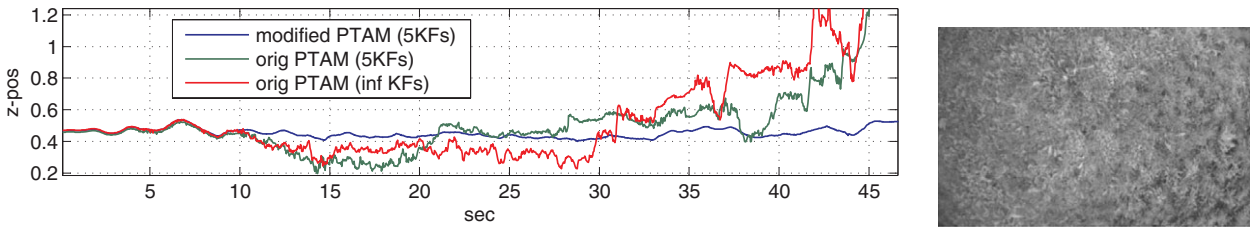
(a) (b) (c)

Figure 11. Two schematic situations in which tracking features in the lowest pyramid level are crucial. Black represents the key-frame in which one feature (red dot) was added to the map. Green corresponds to the current camera frame in which we attempt to find the previously mapped feature. (a) Moving away from a previously mapped feature in the first pyramid level results in tracking this feature in the lowest level in the current frame. Similarly, for wide-angle lenses as in (b), the peripheral features are more distant than centric features. Hence, they will be tracked on lower pyramid levels than centric ones. In (c) the map only contains features triangulated in higher pyramid levels. This image was captured with a 120° fish-eye lens. While features near the rim are naturally farther away, they get tracked correctly in the lowest level (blue dots).



(a) (b)

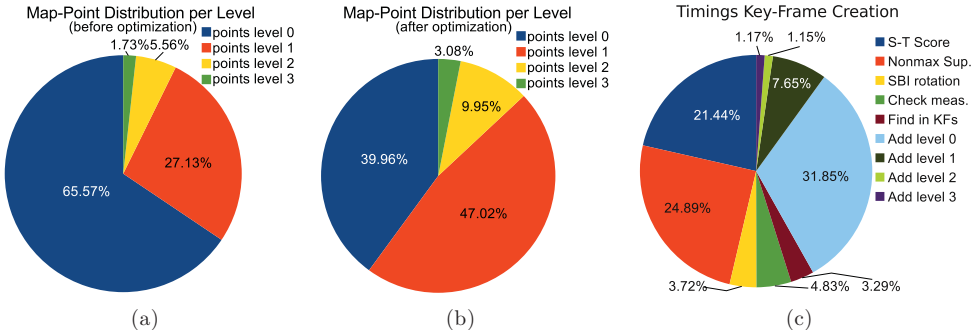
Figure 12. (a) Plot of the PTAM position estimate in x over a cobblestone road. We only plot the x-axis for better legibility and because the motion was along the (negative) x-axis. Thanks to the SLAM behavior of the original PTAM retaining all key-frames, it recovers its position once back at the original place ($t = 41$ and $t = 58$ s). At $t = 76$ s, the map is too corrupted and the algorithm fails. When we only retain the closest five key-frames, the original PTAM soon fails ($t = 21$ s), whereas our improved version runs through the entire dataset without issues. (b) Sample image of the cobblestone road in camera view.



(a) (b)

Figure 13. (a) Test over a grass area moving in x and y and maintaining roughly the same altitude in z in a manual flight. Because the grass moves under the down-wash of the MAV, features in the lowest pyramid level are unreliable and soon corrupt the map of the original PTAM. Our improved version runs through the entire dataset without issues. (b) Sample image of the grass area in camera view.

change occurred after successful reinitialization. This is represented in the drop of altitude with respect to ground truth and is due to the movement in the 2.5 s without any map. The visual estimate is scaled and shifted to match the Vicon ground truth. We only applied this at the beginning and note that our scale and pose propagation methods upon reinitialization work well. As in Weiss et al. (2012b), during the mapless phase, we controlled the MAV in velocity using an inertial-optical-flow-based body-velocity estimation and control.



(a) (b) (c)

Figure 14. (a) Feature distribution per level on key-frame creation before applying a bundle adjustment step. Roughly $\frac{2}{3}$ are features in the lowest pyramid level. (b) Feature distribution after the bundle adjustment step. We see that in the lowest pyramid level, not only are there the most features, but also most are rejected as outliers. (c) The time spent on the different tasks for creating a key-frame. All major tasks are on a per-feature base. This means we can directly save computation time when reducing the number of features Lynen (2011).

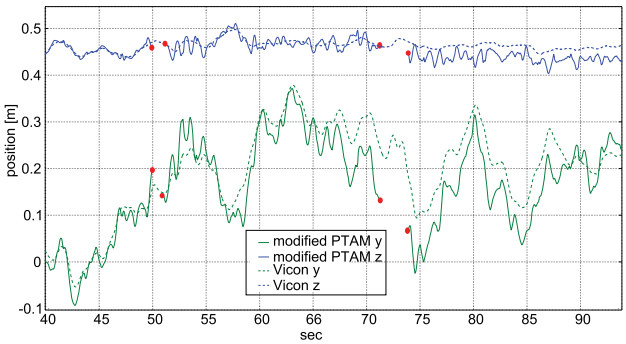


Figure 15. Two reinitialization steps at $t = 50$ and 72 s. The gap between the red dots marks the initialization phase. A slight scale shift is observed after the second reinitialization: the estimated z-value is lower than before and lower than the ground truth.

3.4. Inverted Index Structure for Map-point Filtering

A further improvement is an intelligent selection of map-points in order to increase tracking speed in large and sparse maps. On each frame, point-correspondences between the map points and points in the current image have to be established for tracking. In the original PTAM, this scales linearly with the number of points in the map Figure 17(a). We implemented an inverted index structure based on the grouping of map points inside key-frames, which enabled us to discard large groups of map-points with a low probability of being in the field-of-view Lynen (2011). The search for visible points is performed by reprojecting a small set of distinct map-points from every key-frame, which permits an inference on their visibility from the current key-frame. The total number of points that need evaluation by reprojection is thereby significantly reduced. This leads to a scaling of the system in linear order of the visible key-frames rather than in linear order with the overall number of points in the map.

Table I. Tracking timings for different map structures (Lynen, 2011).

	Dense Map	Sparse Map
Original	10.49 ms (std: 1.22 ms)	8.01 ms (std: 1.62 ms)
Inv. Index	11.05 ms (std: 3.33 ms)	1.32 ms (std: 0.42 ms)

Figure 17(b) shows the performance of our modification. Initially we moved the camera in loops. In this situation, many key-frames have to be considered in the current camera image and our modification shows a slight overhead with respect to the original code because of the computational overhead of indexing the features. In the second part of the data-set, we moved on a straight line. In such situations, we outperform the original framework clearly because of the selective feature reprojection. On large-scale maneuvers, the current camera frame often has to consider only a part of the map. Thus our approach yields a crucial performance gain in these situations. Figures 17(c) and 17(d) show the situation in which the overhead of our modification is dominant and where performance gain due to selective feature reprojection is dominant. Table I summarizes the results for the two different situations. Figure 16 shows graphically the improvement in speed for the reprojection step with growing map sizes. The maps are of the same type as Figure 17(d).

In this section, we presented an approach to render a single camera into a valid onboard and real-time motion

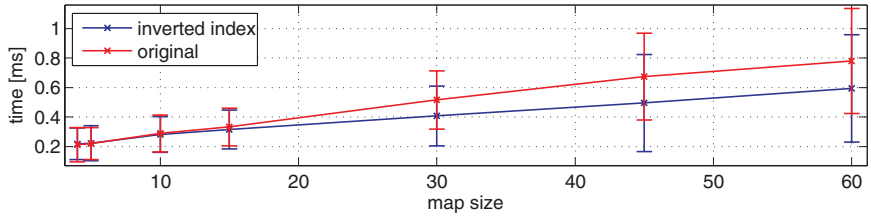


Figure 16. Timings for the reprojection step versus map size for the original PTAM and our improved version.

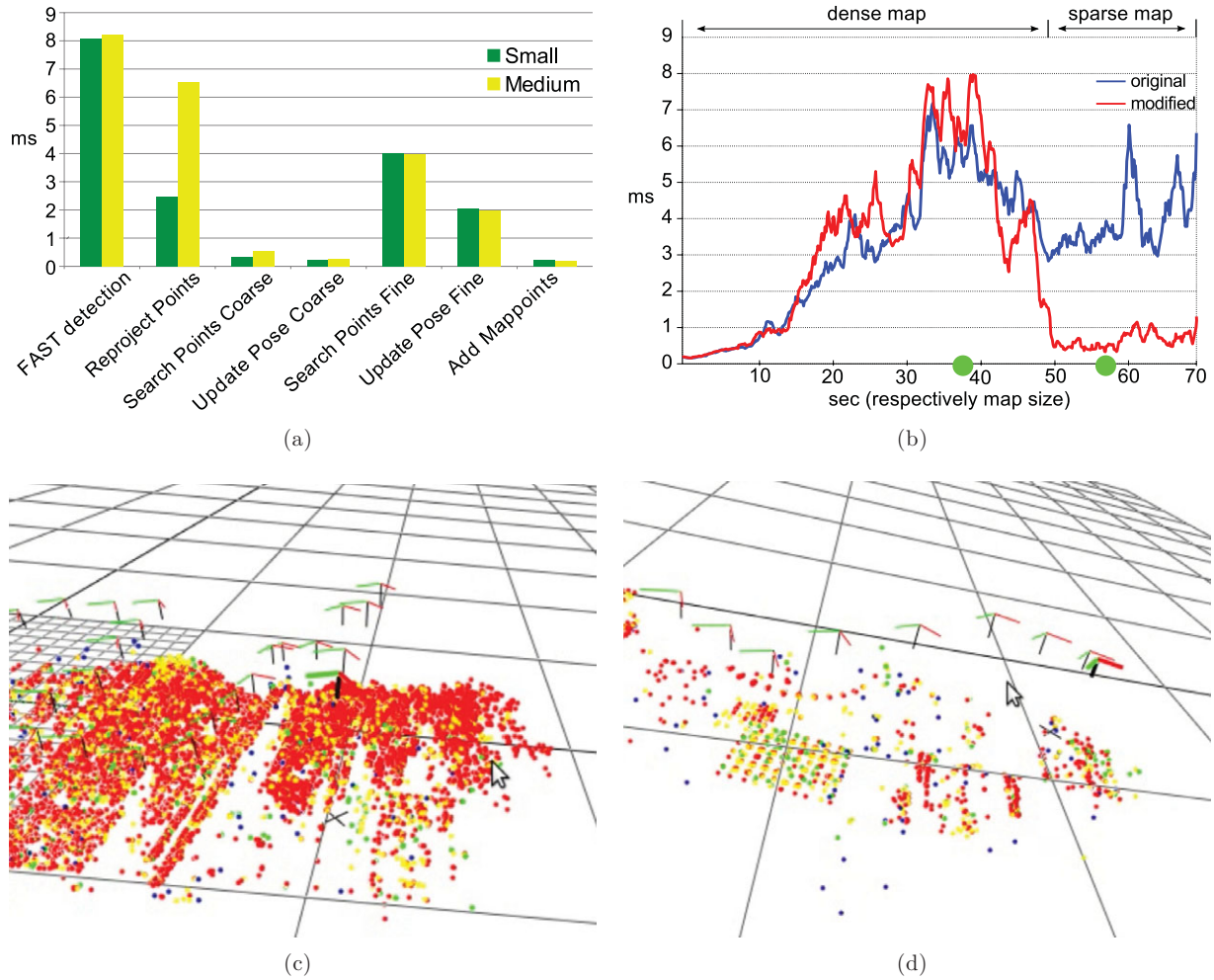


Figure 17. (a) Timings of the different steps performed on each camera frame for tracking. Only the reprojection of map points is dependent on the map size. (b) Timings for our modified code (red) and the original code (blue): Our code clearly outperforms the original code on sparse maps where only a few key-frames are marked as visible (typical for large environments and during exploration). On small and dense maps, the overhead slows our solution down. The green dots mark the screen-shots of (c) and (d). (c) depicts a dense map whereas (d) depicts a sparse map (Lynen, 2011).

sensor for the scaled camera pose. Whereas state-of-the-art uses offboard processing and/or artificial landmarks for vision-based micro helicopter pose estimation, we showed such an estimation onboard an atom 1.6 GHz computer, in real-time (20 Hz) and in a previously unknown outdoor environment.

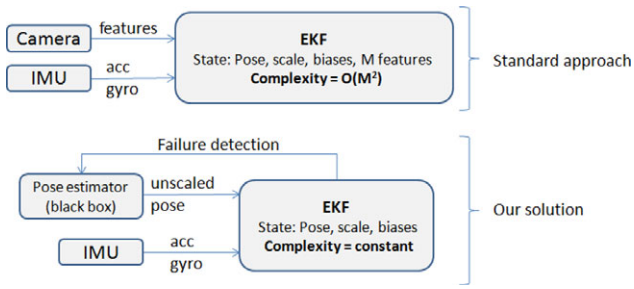


Figure 18. In contrast to tightly coupled solutions (top schematic), our filter runs at constant computational complexity. Moreover, treating the pose estimation part as a black box, we are independent of the underlying pose estimation method. The whole framework has thus the complexity of the chosen pose estimator. As we show in this work, we can still detect failures and drifts of the black box (Weiss and Siegwart, 2011).

4. MODULAR SENSOR FUSION: STATE ESTIMATION AND SENSOR SELF-CALIBRATION

In this section, we discuss a modular sensor-fusion approach in order to estimate the MAV's pose *and* (inter) sensor calibration of the full sensor suite using only a monocular camera and an IMU. The novelty of this contribution with respect to our previous work Weiss et al. (2011); Weiss and Siegwart (2011) is the addition of the visual pose drifts in the filter framework. The continuous estimation of these drifts ensures a gravity-aligned navigation frame that is—along with the constant computational complexity of the overall framework and the metric scale estimate—the most important requirement for long-term MAV navigation. As in our previous work, we use a loosely coupled approach that allows us to use a key-frame based pose estimator as a black box yielding an arbitrarily scaled 6 DOF pose. This allows us to be independent of computationally complex EKF-SLAM solutions and, equally importantly, we can exchange this black box with any other sensor. For example, in Weiss Achtelik, Chli, and Siegwart (2012), we exchanged the camera with a 6 DOF pose reading from a Vicon motion capture system or with a 3 DOF position reading while still using the same approach as described here. In Weiss (2012), we made use of this modularity to fuse multiple sensors on the system. Here, using the camera as a black box, we have to actively detect failure modes of the independent visual pipeline (Figure 18). In this work, we aim at a *combined* (pose estimation and calibration) approach while running *onboard* and in *real-time* rendering the approach suitable for *large* environments. The approach is based on an EKF sensor-fusion using the IMU readings for the propagation step and the visual measurement in the update step. The inertial sensor yields the rotational velocity and acceleration in three axes. The visual sensor provides the system with a 3D position in an undefined scale, and scale-free attitude estimations with respect to its own visual frame.

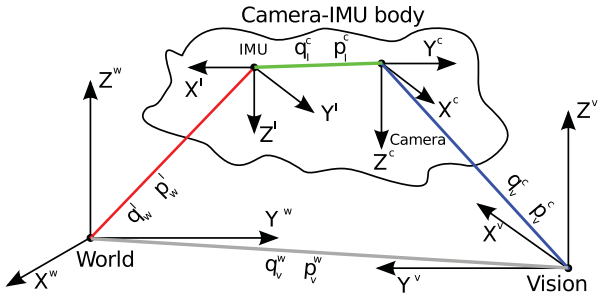


Figure 19. Coordinate frames in the setup. Between every frame there is a rotation q and translation p. In green, the IMU-camera transformation. Red values denote the transformation between the world and the IMU frame and are used for robot control. Blue values are the measurements of the black-boxed visual framework. In gray we denote the short-term stationary transformations between the vision and the world frame. Note that a change in q_v^w reflects the angular drift, whereas p_v^w reflects the translational drift of the visual frame with respect to the world frame (Weiss and Siegwart, 2011).

4.1. Extended Kalman Filter Framework

Figure 19 shows the situation of the camera-IMU setup with its corresponding coordinate frames: In green, the transformation from the IMU frame to the camera frame contains the calibration parameters of the system. They are constant up to structural deformations. In red, the transformation from the world frame to the IMU frame is our robot pose of interest. In blue, the transformation from the visual frame to the camera is measured with an unknown scale by the black-boxed visual framework. In gray, the transformation between the visual frame and the world frame is constant for short periods of time—their drift is of a spatial nature. We introduce this transformation as rotational (q_v^w) and translational (p_v^w) drifts of the black-boxed visual framework with respect to the world frame. The need for the introduction of these drift states and the world reference frame, respectively, arises from two properties of our proposed system. First, the (possibly arbitrarily scaled and spatially drifting) pose input can be generated by any suitable 6 DOF algorithm or sensor. Our system treats this input as one coming from a black box. Second, an independent reference frame allows a modular and versatile extension of the framework to additional sensors.

4.1.1. IMU Sensor Model

We assume that the inertial measurements contain a certain gyro bias b_ω and accelerometer bias b_a and additive, zero-mean white Gaussian noise for a gyro and accelerometer sensor, respectively, (n_ω and n_a). Thus, for the real angular velocities ω and the real accelerations a in the IMU frame, we have $\omega = \omega_m - b_\omega - n_\omega$ and $a = a_m - b_a - n_a$, where the subscript m denotes the measured value. The dynamics of

the nonstatic biases b are modeled as a random process $\dot{b_\omega} = n_{b_\omega}$ and $\dot{b_a} = n_{b_a}$. We use the manufacturer's data of the IMU to obtain a well-dimensioned noise model.

4.1.2. State Representation

The state of the filter is composed of the position p_w^i of the IMU in the world frame W, its velocity v_w^i, and its attitude quaternion q_w^i describing a rotation from the world frame W to the IMU frame I. We also add the gyro and acceleration biases b_ω and b_a, as well as the visual scale factor λ. The calibration states are the rotation from the IMU frame to the camera frame q_i^c, and the position of the camera center in the IMU frame p_i^c. Additionally, we include the drifts between the black-boxed visual frame V and the fixed world frame W. The rotational drifts are reflected in q_v^w and the translational ones in p_v^w. We do not include the gravity vector in the state space, since our world reference frame is aligned with it. Our entire state yields a 31-element state vector x:

$$x = \left\{ p_w^i \; v_w^i \; q_w^i \; b_\omega \; b_a \; \lambda \; p_i^c \; q_i^c \; p_v^w \; q_v^w \right\}. \tag{1}$$

The introduction of the visual drifts in position p_v^w and attitude q_v^w represents the main difference from our previous work (Weiss et al. 2011; Weiss and Siegwart, 2011), and they are crucial for a gravity-aligned navigation frame in long-term missions. The following differential equations govern the state:

$$\begin{aligned}
\dot{p}_w^i &= v_w^i, \\
\dot{v}_w^i &= C_{(q_w^i)}^T (a_m - b_a - n_a) - g, \\
\dot{q}_w^i &= \frac{1}{2} \Omega(\omega_m - b_\omega - n_\omega) q_w^i, \\
\dot{b}_\omega &= n_{b_\omega}, \quad \dot{b}_a = n_{b_a}, \quad \dot{\lambda} = 0, \quad \dot{p_i^c} = 0, \\
&\dot{q_i^c} = 0, \quad \dot{p_v^w} = 0, \quad \dot{q_v^w} = 0.
\end{aligned} \tag{2}$$

$C_{(q)}$ is the rotational matrix corresponding to the quaternion q, g is the gravity vector in the world frame, and $\Omega(\omega)$ is the quaternion-multiplication matrix of ω. We assume the scale and visual frame V to drift spatially and not temporally. Thus we apply a zero motion model. Such states need special attention in the implementation of the system because their uncertainty has to be actively adapted upon the specific spatial event (e.g., key-frame creation). For example, when using a key-frame-based visual framework to calculate the camera pose, the covariance of the visual scale λ and the visual frame drift states p_v^w, q_v^w should be augmented upon key-frame creation only.

Taking the expectations of the above derivatives and the beforehand discussed noise model for the filter state propagation, we obtain

$$\hat{\dot{p}}_w^i = \hat{v}_w^i, \tag{3}$$

$$\hat{\dot{v}}_w^i = C_{(\hat{q}_w^i)}^T (a_m - \hat{b}_a) - g, \tag{4}$$

$$\hat{\dot{q}}_w^i = \frac{1}{2} \Omega(\omega_m - \hat{b}_\omega) \hat{q}_w^i, \tag{5}$$

$$\begin{aligned}
&\hat{\dot{b}}_\omega = 0, \quad \hat{\dot{b}}_a = 0, \quad \hat{\dot{\lambda}} = 0 \; \hat{\dot{p}}_i^c = 0, \quad \hat{\dot{q}}_i^c = 0, \\
&\hat{\dot{p}}_v^w = 0, \quad \hat{\dot{q}}_v^w = 0.
\end{aligned} \tag{6}$$

4.1.3. Error State Representation

In the above-described state representation, we use quaternions as an attitude description. It is common that in such a case we represent the error and its covariance not in terms of an arithmetic difference but with the aid of an error quaternion. This increases numerical stability and handles the quaternion in its minimal representation Trawny and Roumeliotis (2005). Therefore, we define the 28-element error state vector

$$\tilde{x} = \left\{ \Delta p_w^{i^T} \; \Delta v_w^{i^T} \; \delta\theta_w^{i^T} \; \Delta b_\omega^T \; \Delta b_a^T \; \Delta\lambda \; \Delta p_i^{c^T} \; \delta\theta_i^{c^T} \; \Delta p_v^{w^T} \; \delta\theta_v^{w^T} \right\} \tag{7}$$

as the difference of an estimate $\hat{x}$ to its quantity x, i.e., $\tilde{x} = x - \hat{x}$. We apply this to all state variables except the error quaternions, which are defined as follows:

$$\delta q_w^i = q_w^i \otimes \hat{q}_w^{i^{-1}} \approx \left[\frac{1}{2} \delta\theta_w^{i^T} \; 1 \right]^T, \tag{8}$$

$$\delta q_i^c = q_i^c \otimes \hat{q}_i^{c^{-1}} \approx \left[\frac{1}{2} \delta\theta_i^{c^T} \; 1 \right]^T, \tag{9}$$

$$\delta q_v^w = q_v^w \otimes \hat{q}_v^{w^{-1}} \approx \left[\frac{1}{2} \delta\theta_v^{w^T} \; 1 \right]^T. \tag{10}$$

The differential equations for the continuous time error state are

$$\Delta \dot{p}_w^i = \Delta v_w^i, \tag{11}$$

$$\Delta \dot{v}_w^i = -C_{(\hat{q}_w^i)}^T \lfloor \hat{a}_\times \rfloor \, \delta\theta - C_{(\hat{q}_w^i)}^T \Delta b_a - C_{(\hat{q}_w^i)}^T n_a, \tag{12}$$

$$\delta\dot{\theta}_w^i = -\lfloor \hat{\omega}_\times \rfloor \, \delta\theta - \Delta b_\omega - n_\omega, \tag{13}$$

$$\begin{aligned}
&\Delta \dot{b}_\omega = n_{b_\omega}, \quad \Delta \dot{b}_a = n_{b_a}, \quad \Delta\dot{\lambda} = 0 \Delta \dot{p_i^c} = 0, \quad \delta\dot{q_i^c} = 0, \\
&\Delta \dot{p_v^w} = 0, \quad \delta\dot{q_v^w} = 0,
\end{aligned} \tag{14}$$

with $\hat{\omega} = \omega_m - \hat{b}_\omega$, $\hat{a} = a_m - \hat{b_a}$, and $\lfloor \hat{\omega}_\times \rfloor$ as the skew-symmetric matrix of $\hat{\omega}$. This can be linearized to the continuous-time error state equation

$$\dot{\tilde{x}} = F_c \tilde{x} + G_c n, \tag{15}$$

with n being the noise vector $n = [n_a^T, n_{b_a}^T, n_\omega^T, n_{b_\omega}^T]^T$. In the solution presented here, we are particularly interested in the speed of the algorithm. This is why we assume F_c and

G_c to be constant over the integration time step between two consecutive state propagations. For the discretization, we may therefore write

$$F_d = \exp(F_c \Delta t) = \mathbf{I_d} + F_c \Delta t + \frac{1}{2!} F_c^2 \Delta t^2 + \cdots . \tag{16}$$

Careful analysis of the matrix exponents reveals a repetitive and sparse structure. This allows us to express F_d exactly without approximation and to use sparse matrix operations later in the implementation of the filter. The matrix F_d has the following structure:

$$F_d = \begin{bmatrix} \mathbf{I_{d_3}} & \Delta t & A & B & -C^T_{(\hat{q}^i_w)} \frac{\Delta t^2}{2} & \mathbf{0_{3x13}} \\ \mathbf{0_3} & \mathbf{I_{d_3}} & C & D & -C^T_{(\hat{q}^i_w)} \Delta t & \mathbf{0_{3x13}} \\ \mathbf{0_3} & \mathbf{0_3} & E & F & \mathbf{0_3} & \mathbf{0_{3x13}} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{I_{d_3}} & \mathbf{0_3} & \mathbf{0_{3x13}} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{I_{d_3}} & \mathbf{0_{3x13}} \\ \mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{I_{d_{13}}} \end{bmatrix}.$$

We now use the small-angle approximation for which $|\omega| \to 0$, apply de l'Hopital rule, and obtain a compact solution for the six matrix blocks A, B, C, D, E, F (Trawny and Roumeliotis, 2005):

$$A = -C^T_{(\hat{q}^i_w)} \lfloor \hat{a}_\times \rfloor \left(\frac{\Delta t^2}{2} - \frac{\Delta t^3}{3!} \lfloor \omega_\times \rfloor + \frac{\Delta t^4}{4!} \lfloor \omega_\times \rfloor^2 \right),$$

$$B = -C^T_{(\hat{q}^i_w)} \lfloor \hat{a}_\times \rfloor \left(\frac{-\Delta t^3}{3!} + \frac{\Delta t^4}{4!} \lfloor \omega_\times \rfloor - \frac{\Delta t^5}{5!} \lfloor \omega_\times \rfloor^2 \right),$$

$$C = -C^T_{(\hat{q}^i_w)} \lfloor \hat{a}_\times \rfloor \left(\Delta t - \frac{\Delta t^2}{2!} \lfloor \omega_\times \rfloor + \frac{\Delta t^3}{3!} \lfloor \omega_\times \rfloor^2 \right),$$

$$D = -A,$$

$$E = \mathbf{I_d} - \Delta t \lfloor \omega_\times \rfloor + \frac{\Delta t^2}{2!} \lfloor \omega_\times \rfloor^2,$$

$$F = -\Delta t + \frac{\Delta t^2}{2!} \lfloor \omega_\times \rfloor - \frac{\Delta t^3}{3!} \lfloor \omega_\times \rfloor^2 .$$

G_c is then, from Eq. (15),

$$G_c = \begin{bmatrix} \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\ -C^T_{(\hat{q}^i_w)} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & -\mathbf{I_{d_3}} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{I_{d_3}} \\ \mathbf{0_3} & \mathbf{I_{d_3}} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{0_{13x3}} \end{bmatrix}.$$

We can now derive the discrete-time covariance matrix Q_d Maybeck (1979) as

$$Q_d = \int_{\Delta t} F_d(\tau) G_c Q_c G_c^T F_d(\tau)^T d\tau, \tag{17}$$

with Q_c being the continuous-time system noise covariance matrix $Q_c = \mathrm{diag}(\sigma^2_{n_a}, \sigma^2_{n_{b_a}}, \sigma^2_{n_\omega}, \sigma^2_{n_{b_\omega}})$. With the discretized error state propagation and error process noise covariance matrices, we can propagate the state according to the regular EKF procedure.

4.1.4. Measurement

While body accelerations and angular velocities are used for the state propagation, the mentioned (camera) pose measurement is used for the filter update. That is, we account for the temporal drift of the sensor used in the propagation phase. The pose measurement may drift spatially in its attitude and position. For a better understanding, we neglect the position drift for the moment and assume the measurement only drifts in all three angular directions. Mathematically, this means that p^w_v is zero and can be neglected in the filter equations. Intuitively, this means that our world frame is no longer fixed, but the position drifts with the vision frame to which the visual pose sensor is referred. We will discuss the nonobservability of p^w_v in Section 5.

4.1.5. Measurement: Covariances

For the filter update, we first estimate the covariance of the measurement noise. For example, the covariance of the camera pose estimation in an EKF-SLAM is straightforward. For calculating the covariance of a pose estimation using the five-point algorithm or similar, we refer to Beder and Steffen (2006). Pose estimations from nonlinear optimization steps in key-frame-based solutions are more complex: We suggest using the estimation from Beder and Steffen (2006) for the first two key-frames. For the following ones, we suggest the method of Eudes and Lhuillier (2009) defining the N closest key-frames as loose and the rest (at least the first two) as fixed. As an approximation, we assume the actual pose to have the same uncertainty as the distance weighted mean of the uncertainties of the nearest N key-frames not fixed in the bundle adjustment step. Note that N is a design parameter to ensure fast processing time.

We distinguish between the position n_p and rotational n_q measurement noise. This yields the six-vector $n_m = [n_p \; n_q]^T$ with its measurement-covariance matrix R. Note that this noise model is an approximation that is needed in the Kalman filter theory. In reality, the noise in position or rotation may be temporally correlated and is not of a Gaussian nature. Such approximations are one reason why the implemented linearized and discretized system has to be tested under real conditions to ensure its functioning (see Section 6).

4.1.6. Measurement: Model

For the camera position measurement p^c_v, we have the following measurement model:

$$z_p = p^c_v = C^T_{(q^w_v)} (p^i_w + C^T_{(q^i_w)} p^c_i) \lambda + n_p, \tag{18}$$

with $C_{(q_w^i)}$ as the IMU's attitude and $C_{(q_v^w)}$ the rotation from the visual frame to the world frame.

We define the position error as

$$\begin{aligned}\tilde{z}_p &= z_p - \hat{z}_p = C^T_{(q_v^w)}(p_w^i + C^T_{(q_w^i)}p_i^c)\lambda + n_p \\ &\quad - C^T_{(\hat{q}_v^w)}(\hat{p}_w^i + C^T_{(\hat{q}_w^i)}\hat{p}_i^c)\hat{\lambda},\end{aligned} \tag{19}$$

which can be linearized to $\tilde{z}_{p_l} = H_p\tilde{x}$, with

$$H_p = \begin{bmatrix} C^T_{(\hat{q}_v^w)}\hat{\lambda} \\ \mathbf{0_3} \\ -C^T_{(\hat{q}_v^w)}C^T_{(\hat{q}_w^i)}\lfloor \hat{p}_i^c \rfloor \hat{\lambda} \\ \mathbf{0_3} \\ \mathbf{0_3} \\ C^T_{(\hat{q}_v^w)}C^T_{(\hat{q}_w^i)}\hat{p}_i^c + C^T_{(\hat{q}_v^w)}\hat{p} \\ C^T_{(\hat{q}_v^w)}C^T_{(\hat{q}_w^i)}\hat{\lambda} \\ \mathbf{0_3} \\ -C^T_{(\hat{q}_v^w)}\left\lfloor (p_w^i + C^T_{(\hat{q}_w^i)}p_i^c)\lambda \right\rfloor \end{bmatrix}^T$$

using the definition of the error quaternion

$$q_w^i = \delta q_w^i \otimes \hat{q}_w^i, \tag{20}$$

$$C_{(q_w^i)} = C_{(\delta q_w^i)}C_{(\hat{q}_w^i)}, \tag{21}$$

$$C_{(\delta q_w^i)} \approx \mathbf{I_d} - \lfloor \delta\theta_{w_\times}^i \rfloor. \tag{22}$$

For the rotation measurement, we again apply the notion of an error quaternion. The vision algorithm yields the rotation from the vision frame into the camera frame q_v^c. We can model this as

$$z_q = q_v^c = q_i^c \otimes q_w^i \otimes q_v^w, \tag{23}$$

which yields the error measurement

$$\begin{aligned}\tilde{z}_q &= z_q - \hat{z}_q \\ &= q_i^c \otimes q_w^i \otimes q_v^w \otimes \left(\hat{q}_i^c \otimes \hat{q}_w^i \otimes \hat{q}_v^w\right)^{-1} \\ &= H_q^{wi}\delta q_w^i = H_q^{ic}\delta q_i^c = H_q^{vw}\delta q_v^w,\end{aligned} \tag{24}$$

where H_q^{wi}, H_q^{ic}, and H_q^{vw} are the matrices when the error measurement is linearized versus the filter error states δq_w^i, δq_i^c, and δq_v^w, respectively. Finally, the measurements can be stacked together as

$$\begin{bmatrix} \tilde{z}_p \\ \tilde{z}_q \end{bmatrix} = \begin{bmatrix} H_p \\ \mathbf{0_{3x6}}\tilde{H}_q^{wi}\mathbf{0_{3x10}}\tilde{H}_q^{ic}\tilde{H}_q^{vw} \end{bmatrix}\tilde{x}$$
$$\tilde{\mathbf{z}} = \mathbf{H}\tilde{\mathbf{x}} \tag{25}$$

using the approximation of $H_q^{xy} = \begin{pmatrix} 1 & 0_{1x3} \\ 0_{3x1} & \tilde{H}_q^{xy} \end{pmatrix}$. This is justified since the expectation of the error quaternions is unit quaternions.

4.1.7. Update

Once we obtain the measurement matrix $\mathbf{H}$, we can update our estimate according to the well-known Kalman filter procedure. The error quaternion is calculated by following Eq. (8) and ensuring its unit length. The error state covariance is updated as follows:

$$P_{k+1|k+1} = (\mathbf{I_d} - KH)P_{k+1|k}(\mathbf{I_d} - KH)^T + KRK^T. \tag{26}$$

This concludes the basic filter design and its implementation in an EKF framework. As we will see in the next section, this design contains unobservable states. We discuss this issue in detail and find the interconnectivity between the unobservable states in order to find adequate solutions.

4.2. False Pose Estimate Detection

Unlike tightly coupled approaches, with our solution we can treat the visual part (i.e., the visual pose estimation) as a black box. Of course, one desires to detect failures and drifts of the black box to ensure the ongoing functionality of the whole framework. For details, we refer to our previous work in Weiss and Siegwart (2011). There, we did not estimate the visual attitude drift q_v^w but assumed it to be fixed during the whole mission time. In this work, we estimate this drift in the above-described EKF framework. This is a key step to enable long-term navigation. We can still apply our method to failure detection since the state is spatially drifting. For the present compendium, we include a summary of the approach below.

We note that in Section 4.1 we considered the rotation between the inertial world frame and the frame of the visual framework q_v^w as constant during a filter prediction step. This holds since the state is spatial drifting (i.e., upon keyframe creation) and not temporal drifting. At each filter step k we can measure this rotation as

$$q_v^w(k) = \hat{q}_w^{i^{-1}}(k) \otimes \hat{q}_i^{c^{-1}}(k) \otimes q_v^c(k). \tag{27}$$

Since the drift is slow compared to the filter and measurement frequency, we can smooth a sequence of measurements of $q_v^w(k)$. We suggest a median filter as the vision part is better modeled with nonzero mean outlier jumps. The estimation of the rotation between inertial and visual frame $\hat{q}_v^w(k)$ using a window of size N is then

$$\hat{q}_v^w(k) = \text{med}(q_v^w(i)), \quad i = [k - N, \ldots, k]. \tag{28}$$

Due to the fact that the drift is slow and spatial, we can identify abrupt jumps in the measured orientation $q_v^w(k)$ with respect to the smoothed estimate $\hat{q}_v^w(k)$ as failures of the visual pose estimation (see also Section 3). A typical plot of the measurement of $q_v^w(k)$ is shown in Figure 20. The sequences where the visual part failed to estimate a correct pose are clearly visible. As soon as a measurement $q_v^w(k)$ lies outside the 3σ error bounds of the past M estimates $\hat{q}_v^w(k)$, it is considered to be a false pose estimate. We analyze the

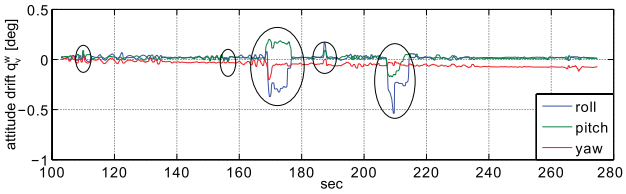


Figure 20. Roll, pitch, and yaw representation of the calculated rotation between the black-box visual frame and the world frame. Note the five encircled areas, which depict a failure of the vision algorithm. Even though we treat the vision algorithm as a black box, we can clearly identify failures. Note the apparent drift in yaw (Weiss and Siegwart, 2011).

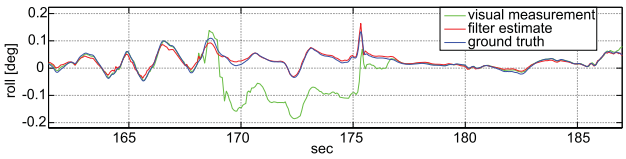


Figure 21. Roll-estimate evolution during a failure of the black-box visual framework. In failure mode, biases and scale are not updated. For the attitude, gyroscopes are short-term integrated since their noise level is sufficiently low. The position measurement-noise is increased to keep a bound on the fast drifting accelerometer integrations. This results in a slightly erroneous position estimate but a still very reliable attitude estimate (Weiss and Siegwart, 2011).

rotational drift only since it is observable in roll and pitch, as we will discuss in Section 5.

During such failure periods, neither $\hat{q}_v^w(k)$ nor the biases and scale are updated since the measured data are corrupted. Note that we do update the MAV attitude, the velocity, and the position to bound the error during long dropouts. Low-cost accelerometers suffer from large drifts and should not be integrated over longer periods of time without updates. In contrast, gyroscope integration is fairly robust over longer periods of time. Thus we increase the visual measurement noise for the position much less than for the attitude. The increasing of the measurement noise during failure sequences bounds the filter error yet trusts the simple integration to a large extent. As Figure 21 shows, the attitude is still estimated reliably.

In this section, we discussed our sensor-fusion approach in order to have a system that not only provides an optimal estimate of the vehicles 6 DOF pose and velocity at IMU rate, but also self-calibrates the (inter) sensor states and the visual drifts continuously. This provides the system with two key properties: First, it is a true power-on-and-go system without the need for tedious calibration procedures. Second, due to the continuous self-calibration, the system is truly capable of performing long-term navigation. We also showed that we can identify failures of the visual part of the system. Thus, we can fully appreciate the benefits of our improvement of fast visual map reinitialization discussed in Section 3.3.

5. OBSERVABILITY ANALYSIS

Intuitively, the above described state estimator will drift in position and yaw according to the position and yaw drift of the visual pose estimator. In this section we perform the observability analysis in order to analytically support this intuition. The analysis is done based on tools of differential geometry and we assume general motion – this is generally given on a real MAV. We apply the non-linear observability analysis proposed in Hermann and Krener (1977). We refer to the work in Mirzaei and Roumeliotis (2007), Kelly and Sukhatme (2011) and Martinelli (2011) for details about how to apply this method to a system similar to ours. We do not recommend an observability analysis in the spirit of Jones (2009) since we aim at an easily extensible method for additional sensors. For the properties and notation on quaternions we refer to Trawny and Roumeliotis (2005). For the notation of *Lie derivatives* we refer to Kelly and Sukhatme (2011). The authors give a short but very comprehensive overview about this mathematical tool for non-linear observability analysis.

5.1. Nonlinear Observability Analysis

Our goal is to show *locally weak observability* of the proposed system according to Hermann and Krener (1977). A short test shows that the above-described system is not globally observable: no acceleration, $p_i^c = 0$, or $\lambda = 0$ are only some examples that show the need for the system to be in a locally weakly observable area. For the analysis, we work on the state variables and not on the error state. The definition of the error state is an approximation in which second- and higher-order terms are discarded under the assumption of a small error state.[6] For the same reason, we do not perform a global observability analysis of the linearized system. In contrast, the observability analysis on the full nonlinear system prevents information loss. A discussion of unobservable areas (or modes) can be found in Martinelli (2011).

[6]The small-angle approximation is used in the derivations of Jacobians, where it allows us to "ignore" second- and higher-order terms. Similarly for additive error, second- and higher-order terms are discarded under the assumption that the error state is small. This results in a (linear) approximation to the nonlinear system, and as such, we are discarding some information that may be useful to prove the system's observability.

We still assume $p_v^w = 0$ and discuss it later in this section. The control affine form of the system (2) is

$$\dot{x} = \begin{bmatrix} \dot{p}_w^i \\ \dot{v}_w^i \\ \dot{q}_w^i \\ \dot{b}_\omega \\ \dot{b}_a \\ \dot{\lambda} \\ \dot{p}_i^c \\ \dot{q}_i^c \\ \dot{q}_v^w \end{bmatrix} = \underbrace{\begin{bmatrix} v_w^i \\ -C_{(q_w^i)}^T b_a - g \\ 0.5\Xi_{(q_w^i)} b_\omega \\ 0_{3\times1} \\ 0_{3\times1} \\ 0 \\ 0_{3\times1} \\ 0_{4\times1} \\ 0_{4\times1} \end{bmatrix}}_{f_0} + \underbrace{\begin{bmatrix} 0_{3\times3} \\ 0_{3\times3} \\ 0.5\Xi_{(q_w^i)} \\ 0_{3\times3} \\ 0_{3\times3} \\ 0_{1\times3} \\ 0_{3\times3} \\ 0_{4\times3} \\ 0_{4\times3} \end{bmatrix}}_{f_1} \omega_m + \underbrace{\begin{bmatrix} 0_{3\times3} \\ C_{(q_w^i)}^T \\ 0_{4\times3} \\ 0_{3\times3} \\ 0_{3\times3} \\ 0_{1\times3} \\ 0_{3\times3} \\ 0_{4\times3} \\ 0_{4\times3} \end{bmatrix}}_{f_2} a_m, \tag{29}$$

where $\Xi_{(q)}$ is the multiplication matrix for the quaternion of rotation q and we have $\dot{q} = \Omega(\omega)q = \Xi_{(q)}\omega$.

Note that Eq. (29) is similar to that stated in Kelly and Sukhatme (2011). The additional states q_v^w, however, will change the measurement equations and the observability analysis fundamentally. The measurements can be summarized as

$$h(x) = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{bmatrix} = \begin{bmatrix} C_{(q_v^w)}^T (p_w^i + C_{(q_w^i)}^T p_i^c)\lambda \\ q_i^c \otimes q_w^i \otimes q_v^w \\ {q_w^i}^T q_w^i \\ {q_i^c}^T q_i^c \\ {q_v^w}^T q_v^w \end{bmatrix}, \tag{30}$$

where h_1 expresses the visual position measurement p_v^c, h_2 expresses the visual attitude measurement q_v^c, and h_3 to h_5 are the unit norm constraints for each rotation quaternion in the system.

We denote the gradient with respect to the state variables of the zero-order Lie derivative of h_n as $\nabla L^0 h_n$, and we denote the gradient of its first-order derivative with respect to f_m as $\nabla L^1_{f_m} h_n$. Following the suggestions of Kelly in Kelly and Sukhatme (2011), we obtain the observability matrix $\mathcal{O}$, using the variables $a_{i,j}$ for better legibility. A matrix rank analysis shows that the system has rank 27 instead of column full rank 28 (apart from the three position drift states p_v^w).

5.2. Discussion of the Unobservable States

The knowledge that we are missing only one single rank motivates us to analyze the issue in more detail. Martinelli (2011) proposed a method to quantitatively describe the interdependencies of unobservable states. By analyzing the system on its continuous symmetries and corresponding indistinguishable regions, the method can be used to define an observable joint state containing the unobservable single states. Toward that end, we calculate the null space of $\mathcal{O}$. We find that the one-dimensional null space of $\mathcal{O}$ is in the subspace spanned by the states X_U,

$$X_U = {}_x p_w^i,\ {}_y p_w^i,\ {}_x v_w^i,\ {}_y v_w^i,\ {}_y q_w^i,\ q_v^w, \tag{31}$$

where ${}_x p_w^i$, ${}_y p_w^i$, ${}_x v_w^i$, ${}_y v_w^i$ are the x and y components of p_w^i and v_w^i, respectively, and ${}_y q_w^i$ is the yaw component of the MAV's attitude. We call the states X_U *jointly observable* since all N dimensions of the subspace spanned by X_U are observable except for one. According to Hermann and Krener (1977) and Martinelli (2011), we can define $N-1$ observable states $s_i(X_U)$, $i = 1, \ldots, N-1$, and one unobservable state $u(X_U)$ in this subspace. Or, in other words, one additional measurement containing information on any of the states in X_U renders all other states in X_U observable.

Furthermore, we see that the visual scale λ, all bias terms b_w, b_a, and the full intersensor calibration p_i^c, q_i^c are observable at all time. Most important to note is that the global roll and pitch of the MAV ${}_{rp}q_w^i$ are observable as well. Only its yaw component ${}_y q_w^i$ belongs to the jointly observable states X_U. In fact, X_U contains all states affected by the lack of having a global yaw reference. This means that

$$\mathcal{O} = \begin{bmatrix} \nabla L^0 h_1 \\ \nabla L^0 h_2 \\ \nabla L^0 h_3 \\ \nabla L^0 h_4 \\ \nabla L^0 h_5 \\ \nabla L^1_{f_0} h_1 \\ \nabla L^1_{f_0} h_2 \\ \nabla L^1_{f_1} h_1 \\ \nabla L^1_{f_1} h_2 \\ \nabla L^2_{f_0 f_0} h_1 \\ \nabla L_{f_0 f_2} h_1 \\ \nabla L_{f_0 f_0 f_2} h_1 \end{bmatrix} = \begin{array}{c} \begin{matrix} \overbrace{}^{p_w^i} & \overbrace{}^{v_w^i} & \overbrace{}^{q_w^i} & \overbrace{}^{b_\omega} & \overbrace{}^{b_a} & \overbrace{}^{\lambda} & \overbrace{}^{p_i^c} & \overbrace{}^{q_i^c} & \overbrace{}^{q_v^w} \end{matrix} \\ \begin{bmatrix} C_v^{w\mathrm{T}}\lambda & 0 & [a_{1,3}] & 0 & 0 & [a_{1,6}] & [a_{1,7}] & 0 & [a_{1,9}] \\ 0 & 0 & [a_{2,3}] & 0 & 0 & 0 & 0 & [a_{2,8}] & [a_{2,9}] \\ 0 & 0 & 2q_w^{i\mathrm{T}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2q_i^{c\mathrm{T}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2q_v^{w\mathrm{T}} \\ 0 & C_v^{w\mathrm{T}}\lambda & [a_{6,3}] & [a_{6,4}] & 0 & [a_{6,6}] & [a_{6,7}] & 0 & [a_{6,9}] \\ 0 & 0 & [a_{7,3}] & [a_{7,4}] & 0 & 0 & 0 & [a_{7,8}] & [a_{7,9}] \\ 0 & 0 & [a_{8,3}] & 0 & 0 & [a_{8,6}] & [a_{8,7}] & 0 & [a_{8,9}] \\ 0 & 0 & [a_{9,3}] & 0 & 0 & 0 & 0 & [a_{9,8}] & [a_{9,9}] \\ 0 & 0 & [a_{10,3}] & [a_{10,4}] & [a_{10,5}] & [a_{10,6}] & [a_{10,7}] & 0 & [a_{10,9}] \\ 0 & 0 & [a_{11,3}] & 0 & 0 & [a_{11,6}] & 0 & 0 & [a_{11,9}] \\ -0 & 0 & [a_{12,3}] & [a_{12,4}] & [a_{12,5}] & [a_{12,6}] & [a_{12,7}] & 0 & [a_{12,9}] \end{bmatrix} \end{array}$$

- measuring the roll or pitch of the IMU with respect to the world frame W does not add any new information. We already can estimate a gravity-aligned navigation frame at all times,
- in contrast, measuring the roll or pitch drift between the vision frame V and the world frame W yields a fully observable system. (This applies only if roll and pitch drifts are nonzero; otherwise, of course, the system still contains a yaw ambiguity),
- also, measuring either the yaw of the IMU with respect to the world frame W or measuring the *yaw drift* between the world frame W and the vision frame V renders the system observable. Using a visual compass to eliminate this latter drift would render the system fully observable. We will show in the next section that the global yaw drift is in practice extremely small in our improved vision pipeline from Section 3.

Since we know that the states in X_U only span one unobservable dimension, a single measurement containing any of the states in X_U renders the system fully observable.

So far, we considered the position drift p_v^w to be zero. Adding this 3D state without adding new measurements concerning the translational states simply results in three additional jointly observable variables with three additional unobservable directions. In addition to the unobservable mode spanning one dimension in the states mentioned in Eq. (31), we find three additional unobservable modes: each one spans one dimension in the subspaces $\{{}_x p_w^i,\ {}_x p_v^w\}$, $\{{}_y p_w^i,\ {}_y p_v^w\}$, and $\{{}_z p_w^i,\ {}_z p_v^w\}$, respectively. This means that, in this setup, the position and the position drift are only jointly observable in each axis and that the navigation frame will slowly drift in position with respect to a world fixed frame. Unlike the need for a gravity-aligned navigation frame, correct global position or yaw is not critical to keep the MAV airborne. We show in the next section that this position and yaw drift is extremely small in practice due to our improvements in the visual pipeline in Section 3.

6. RESULTS

In an attempt to demonstrate the applicability and robustness of the proposed vision-based framework for MAV navigation, we present here a set of experiments we have performed in large outdoor environments. Starting with motion estimation of a manually controlled helicopter, we discuss the performance of the estimation process. We then present the results of motion estimation during an autonomous flight in a disaster area, approaching the conditions of a real autonomous mission as much as possible. The duration of the experiments is bounded by the battery lifetime, which is short compared to *long-term* standards. Nevertheless, we show unprecedented results for vision-based MAV navigation, and our approach would (theoretically) hold for an infinite duration.

While the methodology presented in this article is universal across any platform with a camera and an IMU, for the experiments presented here we use the Firefly hexacopter (Achtelik, Doth, Gurdan, and Stump, 2012), depicted in Figure 1. This MAV is equipped with a downward-looking MatrixVision "Bluefox" wide VGA camera of 120° field of view and a Core2Duo 1.86 GHz processor board. In this architecture, our visual processing pipeline introduced in Section 3 uses 60% of one core at 30 Hz [compared to 100% at 20 Hz on an Atom 1.6 GHz computer Weiss et al., (2012b)]. The camera configuration is chosen to fit best the circumstances of flying outdoors at a mostly obstacle-free height. Note that the camera (and IMU) can be mounted arbitrarily on the system since our approach self-calibrates the extrinsic calibration between the sensors. The sensor-fusion and self-calibration framework presented in Section 4 has been implemented in a distributed manner, including compensation for time-delays on the onboard ARM7 (running the EKF-prediction step) and the onboard Core2Duo computer (running the EKF-update step). Our implementation allows precise and efficient state propagation and thus MAV pose control at 1 kHz. For a detailed description of this distributed implementation, we refer the motivated reader to our earlier work in (Weiss, Achtelik, Chli, and Siegwart, 2012). Throughout all the experiments, we used only the camera as a pose sensor as described in Section 3 and the IMU as sensors for state estimation and system self-calibration.

To mitigate issues with over- and undersaturated image areas, we fixed the camera shutter speed to a constant value before each experiment. Variable shutter speed would lead to changes in the overall image illumination during flight. This causes saturated areas to change their patch footprint (i.e., white areas around an extracted corner suddenly have distinguishable gray values), which is not desirable in our vision pipeline. With our improvements on PTAM and the fixed shutter speed, we managed to work with saturated areas in the scene, eliminating most illumination-related issues.

The attitude estimates provided by the MAV manufacturer, Ascending Technologies, are used here as ground truth. According to the manufacturer, this attitude is calculated using the magnetic field and the IMU information in a complementary filter. For position ground truth, we use GPS readings. For every data comparison, we align the estimated values to the ground truth data in position and absolute yaw at the beginning of the test run only. We note that GPS data have only a precision of 1–2 m; however, in this work it is the only data source available for reasonable comparisons. The ground truth values for the intersensor-calibration parameters are measured by hand.

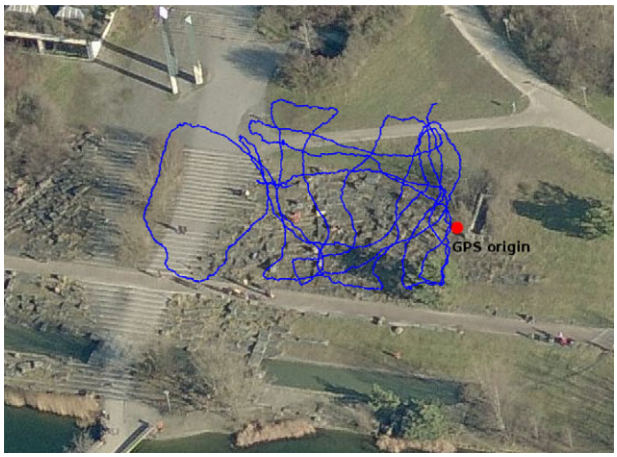


Figure 22. Irchel Park testing area. The area is a slightly inclined stairlike slope with a total inclination of about 15 deg. The blue path shows the about 300-m-long trajectory flown with the MAV (Bing Maps).

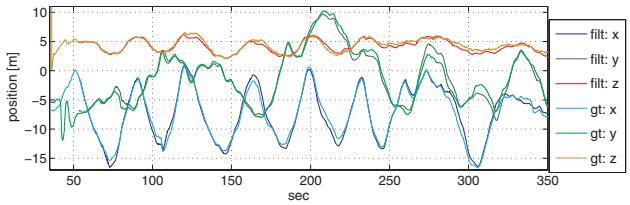


Figure 23. Irchel dataset: the position estimates of our filter framework ('filt') superimposed to GPS ground truth ('gt'). With very minor deviations from ground truth, the plot suggests that the scale is estimated correctly, while the drift of the visual estimation pipeline in position and yaw seems to be very small. We initialize our framework at around $t = 50$ s and retain a maximum of five key-frames.

6.1. Irchel Park: State Estimation and System Self-calibration During a Manual Flight

In this first mission, the MAV is manually instructed to take off and fly a trajectory of a total of about 300 m. The test field is a slightly inclined slope of about 40×80 m with plain grass and also rocky sections. The inclination is roughly measured to be 15 deg as shown in Figure 22, illustrating the GPS-recorded path of the MAV.

In this mission, we retain five key-frames in the visual estimation pipeline (Section 3.1). This results in a non-negligible drift of the visual frame with respect to the world frame and permits verification that the proposed self-calibrating framework can track such drifts correctly and provide a gravity-aligned navigation frame at all times.

Figures 23 and 24 show the ground truth data in position and attitude compared to the estimated values at each time instant. With the estimated values almost coinciding

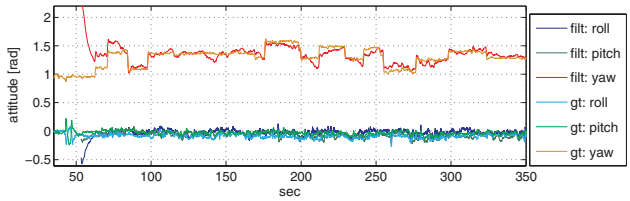


Figure 24. Irchel dataset: the attitude estimate of our filter framework ('filt') superimposed to ground truth from the MAV manufacturer ('gt'). The correct estimate of roll and pitch suggests a good tracking of the attitude drift between the visual and the world frame, while, the drift in the unobservable yaw seems to be very small. The self-calibration of the roll and pitch drift is crucial to keeping the MAV gravity aligned. We initialize our framework at around $t = 50$ s and retain a maximum of five key-frames.

with ground truth for position, one can observe that the visual scale is estimated correctly by our EKF-based framework. Similarly, the accurate match in roll and pitch between ground truth and estimated values shows good tracking of the visual attitude drifts. In Section 5, we discussed how the position and the global yaw are only jointly observable with their drift counterparts. Thus, for a rapidly drifting visual position estimate, we would observe an increasing difference between GPS readings and filter estimates in Figure 23. In the attitude plot in Figure 24, we would see a similar effect in the yaw angle. However, even when only retaining five key-frames in the map, our proposed visual pose estimator has a very low drift in these values, barely visible in these plots (compare Figure 5 to Figure 8 in Section 3.1: position and yaw drifts are barely visible, whereas scale, roll, and pitch drifts are more noticeable).

Even though the MAV position and attitude are accurately estimated, the distance between the IMU and the camera proves too small to estimate reliably for the given (low) excitations in angular velocities during this flight. Our estimates of this distance were generally about 5 cm off the true value. The error on this state, however, is bounded since large errors would result in large disturbances of other, more robust states. Conversely, the intersensor attitude states q_i^c are estimated accurately with an error of about 0.03 rad.

To show better the framework's capabilities of intersensor self-calibration, we use data from a hand-held experiment with adequate excitation but wrong state initializations for p_i^c and q_i^c. For good convergence behavior, the system needs either sufficient excitation or the true lever arm to be reasonably large. A detailed analysis of the required excitation levels per configuration would be beyond of the scope of this work. Figure 25 shows a typical convergence behavior for the distance p_i^c between camera and IMU. The excitation of this dataset has an RMS value of 0.5 rad/s in angular velocity and 1 m/s^2 in linear acceleration. The final error in x,y,z is [17,8,7] mm respectively.

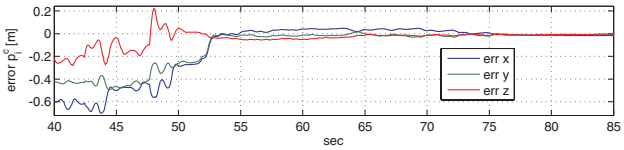


Figure 25. Typical convergence behavior of the intersensor translation p_i^c between camera and IMU. With sufficient excitation, the state converges from an initially wrong value to its correct value. The final error in x,y,z is [17,8,7] mm, respectively.

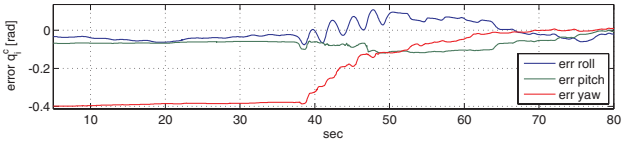


Figure 26. Typical convergence behavior of the intersensor rotation q_i^c between camera and IMU. Excitation in angular velocities (from $t = 38$ s on) visibly helps improve the convergence rate for this state. The final error is [−0.034, −0.034, 0.026] rad for roll, pitch, and yaw, respectively.

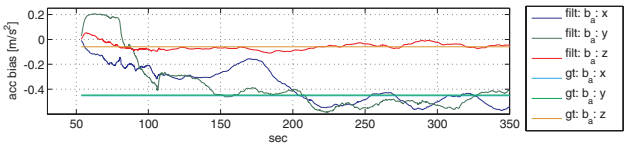


Figure 27. Irchel dataset: the acceleration bias estimate of our filter framework versus the mean value over the accelerations measured during the whole flight used as a ground truth. Note that the x and y biases were about the same during this test. The states slowly converge to the ground truth value.

Figure 26 shows typical convergence behavior for the attitude q_i^c between camera and IMU. Throughout the whole run, the excitation had an RMS value of 1 m/s^2 in linear acceleration. Until $t = 38$ s the excitation had a value of 0.2 rad/s, then a value of 0.9 rad/s. We see that even with low excitation the states converge but they do so extremely slowly. However, excitation in angular velocity is favorable for fast convergence. The final error in the estimate was [−0.034, −0.034, 0.026] rad for roll, pitch, and yaw, respectively. This coincides well with our findings in previous work (Weiss, Achtelik, Chli, and Siegwart, 2012).

The estimates of the gyroscope bias b_ω are very accurate; however, the bias of the accelerometer b_a, plotted in Figure 27, tends to be difficult to estimate because of its low quality of observability [refer to Weiss (2012), Section 3.4.7 for more details]. Even though this state is only slowly converging, it eventually converges to be in certain bounds around the true value as shown in Figure 27. Again, this "oscillating" behavior arises from the low quality of observability of this state.

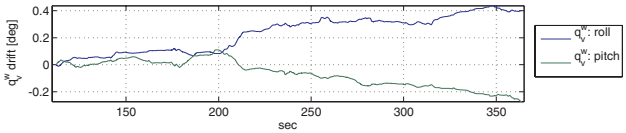


Figure 28. Evolution of the estimated visual drift in roll and pitch with respect to our world reference frame (i.e., navigation frame) in degrees. The very low drift is to be attained to the good feature distribution outdoors and the fact that we did not encounter any map-losses during the flight. The yaw is an unobservable state and is not plotted here.

Upon initialization, the visual framework created its visual frame such that the xy-plane was fitted to the slope. Our framework estimated this attitude offset in roll and pitch between the vision frame and the gravity-aligned world frame (i.e., navigation frame) to be 4.4° in roll and 15.8° in pitch, which are reasonable estimates for this inclined test area. To show the attitude drift of the vision frame with respect to the world frame during this experiment, we removed the initially estimated offset in the drift states q_v^w and only plotted the evolution (i.e., drift during the experiment) of these states in Figure 28. Note that being able to estimate this offset and drift online during the mission is a crucial part of our self-calibrating framework to keep the MAV gravity aligned (and thus airborne). We attribute the very low drift to the uniform distribution of features across the image (typical with a downward-looking camera in outdoor scenes) and the fact that we did not encounter any map-losses while flying. This shows that our improvements on the visual pose estimator over the original PTAM not only render this framework capable of real-time and onboard estimation for the MAV, but also render it sufficiently robust for outdoor scenarios.

6.2. Firefighters' Training Area: Autonomous Flight Mission

The final set of experiments is carried out during an autonomous flight mission in a disaster area shown in Figure 29.[7] We use our proposed framework for autonomous waypoint flight and MAV control. Here, we aim to test the applicability of the proposed navigation framework in a realistic mission under a variety of challenging factors, such as large height changes, strong wind, and autonomous navigation. Note that at the beginning of every experiment, the MAV is manually instructed to take off, followed by the automatic initialization phase of the visual navigation framework—this is identical to PTAM initialization, grabbing two frames from sufficiently different viewpoints to initialize the map grid. Here, the map grid is automatically

[7]Video material can be found at www.youtube.com/watch?v=vHpw8zc7-JQ.

Figure 29. Firefighters' training area for testing autonomous altitude and navigation missions. The area resembles a common disaster area after an earthquake or an explosion (Bing Maps).

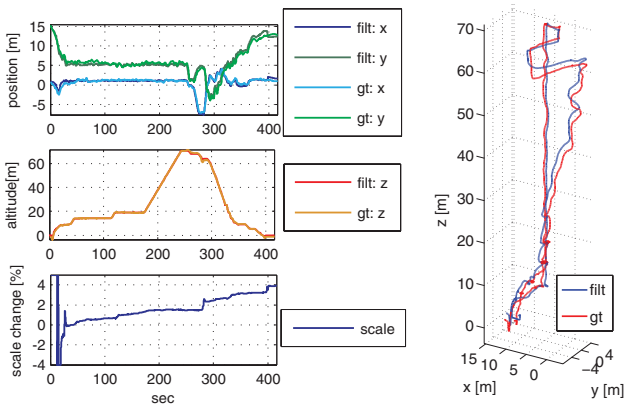


Figure 31. Altitude test: After initialization ($t = 30$ s) we switch to autonomous navigation mode (using only the onboard camera and the IMU as sensors) and ascend up to 70 m without intended lateral motion, as shown in (b). Then, we let the MAV descend while applying lateral motion to show the maneuverability during this drastic height change. The ends of the graphs show safe vision-based landing—still using our proposed framework to control the MAV but with a manually applied velocity vector in the negative z direction. The lower plot in (a) depicts the evolution of the visual scale λ.

initialized as soon as the algorithm detects a disparity of 20 pixels or more between the current camera frame and the frame where the initialization procedure started. That first frame is automatically taken as soon as the MAV detects a height of 4 m or more based on GPS readings. Autonomous takeoff could be performed in future work using our optical flow approach in Weiss et al. (2012b). Furthermore, with such a velocity-controlled MAV, autonomous PTAM initialization could be done as shown in Weiss et al. (2012b). We did not apply this in the present work.

In the first experiment, the MAV navigates autonomously in a vertical trajectory up to 70 m above ground followed by autonomous landing back to the ground. The camera view at different heights during this trajectory is shown in Figure 30. Figure 31 illustrates the filter estimates in both position and attitude, which evidently follow ground truth very closely throughout the experiment. Figure 30 shows the view from the camera onboard the MAV at different altitudes during the ascent at 70 m. This is a perfect example in which a stereo-camera (downward-looking) would quickly reduce to a monocular setup, advocating our choice of a minimal sensor suite of the single camera and the

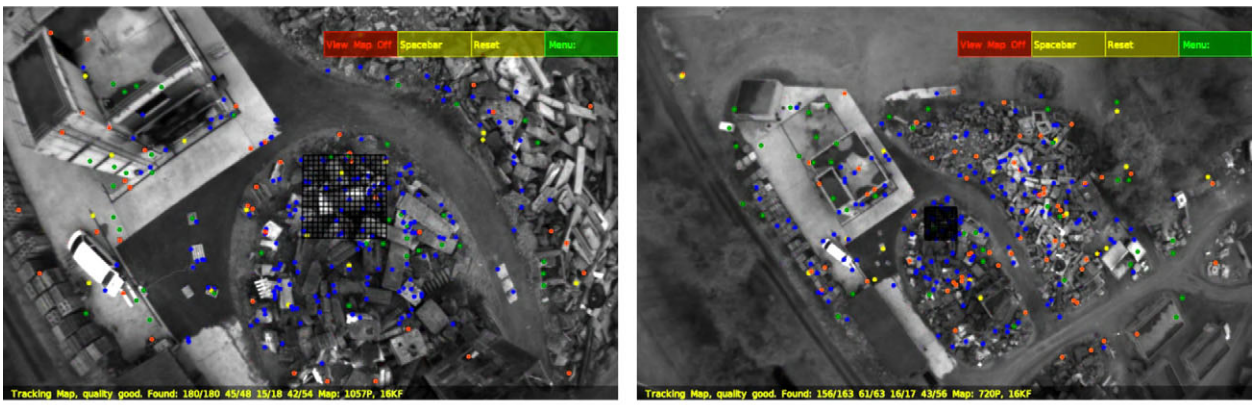
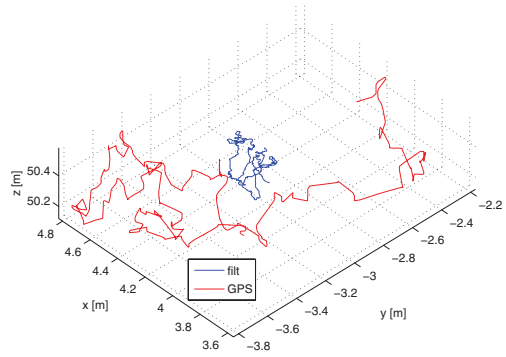
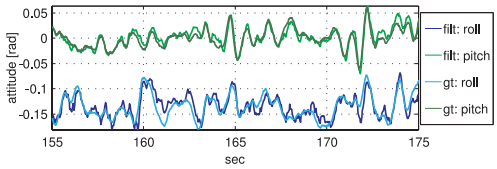


Figure 30. Camera-view with the detected features and map-grid overlaid, while navigating autonomously using the proposed visual-inertial framework purely vision. View (a) is captured at 35 m while ascending to 70 m above ground, as captured in view (b).



(a)



(b)

Figure 32. Hover performance at an altitude of 50 m above ground during 20 s. (a) Estimated trajectory. The position estimates are still reliable at this altitude. The GPS ground truth limitations are evident. In (b), the small roll offset in the MAV's attitude is evident. Our framework counteracts the influence of strong wind to maintain the MAV at the designated hovering spot.

IMU. With this extreme altitude change, the camera-view changes drastically and fast, demonstrating the power of our vision-based navigation framework.

To test the accuracy of our estimation, in Figure 32(a) we illustrate the hovering performance of the MAV at 50 m above ground. We note the limitations of using GPS as a ground truth signal. In Figure 32(b), we plot the attitude during the hovering. The offset in roll is caused by wind, while the MAV is acting against any thrusts in order to maintain its hovering position. To our knowledge, navigating within this height range (i.e., from 0 to 70 m), and achieving such accuracy despite wind disturbances, is unmatched in vision-based navigation for micro helicopters.

To test the navigation applicability of our vision-IMU framework, we performed a flight of a total of 360 m over the same testing area, as shown in Figure 33. The path was designated by a human operator sending waypoints to the MAV via a wifi connection, with the MAV reaching a maximum speed of 2 m/s during the flight. Note that the trajectory length for this experiment has only been bounded by the battery lifetime of the MAV. Figure 34 illustrates that the 3D position estimation from the filter follows GPS ground truth very closely, indicating not only a correct estimate of the scale drift, but also that the position and yaw drifts are

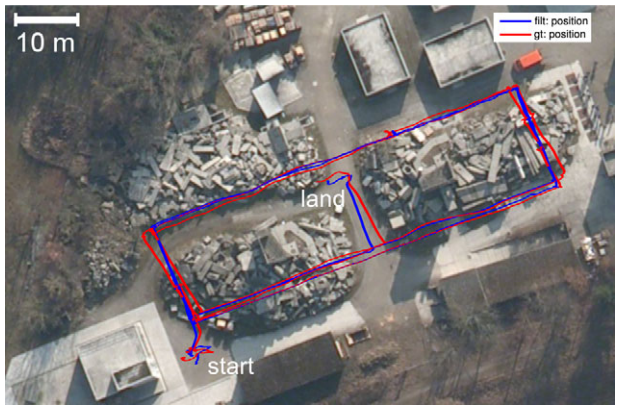


Figure 33. Long-trajectory flight test. After initialization, we switch to autonomous navigation mode, where only monocular and IMU feeds are used. Providing waypoints, the MAV completes a trajectory of 360 m before landing becomes necessary due to battery limitations. With the position estimate of our framework ('filt') following closely the GPS ground truth ('gt'), it is evident that the proposed methodology causes only a very low position and yaw drift despite the lengthy trajectory.

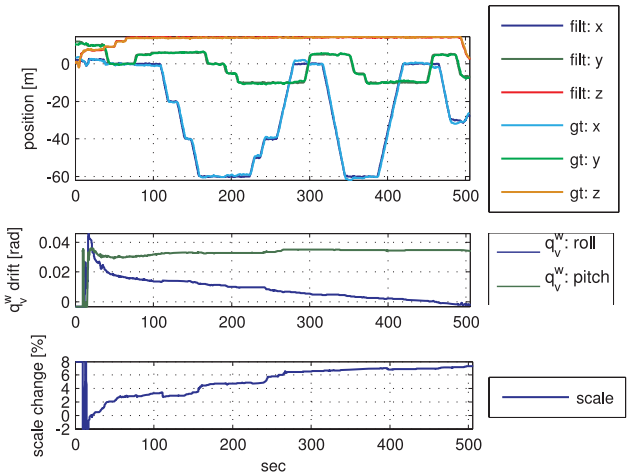


Figure 34. Top: Position estimate in x, y, and z. The saddle points along the x and y axes reflect hovering between the transmission of waypoints to the MAV. Middle: Visual attitude drift occurring throughout the flight. Global yaw being unobservable, we only show the evolution of the roll and pitch drifts. Precise estimation of these drifts prevents gravity-misalignment and subsequent MAV crash. Bottom: Evolution of the visual scale factor. Continuous estimation of this state ensures accurate navigation and control in a metric coordinate frame.

very small. In fact, at the end of this long trajectory, the position error of our visual navigation framework is only 1.47 m, corresponding to about 0.4% with respect to the overall trajectory length. Note here that since this is a visual odometry framework, no loop-closure is identified or

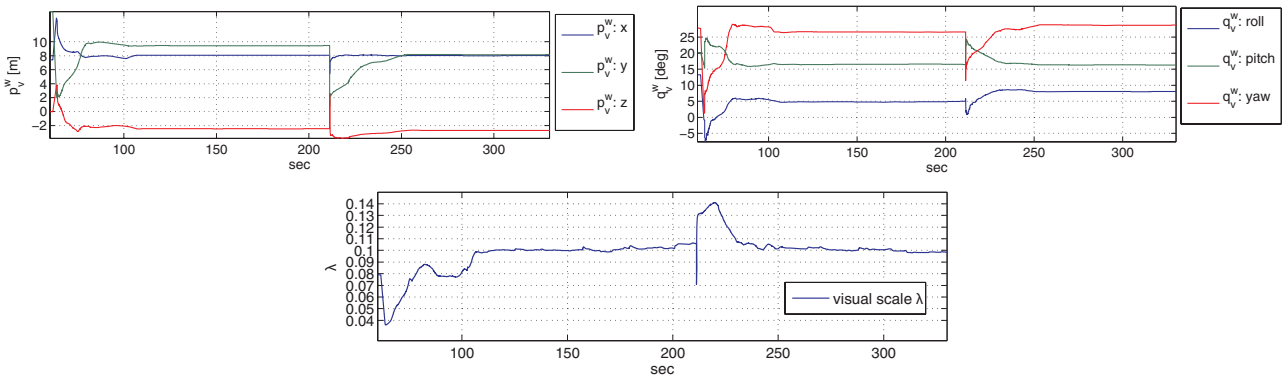


Figure 35. The influence of a map-loss to the visual drift states in position p_v^w, attitude q_v^w, and scale λ. At $t = 75$ s, the camera is introduced as a 6 DOF pose estimation in the system, followed by a brief initialization period. At $t = 211$ s, we force a map reset and reinitialization using our pose propagation improvement discussed in Section 3.3. The negligible change in the map's drift states p_v^w, q_v^w, and λ with respect to the world reference frame shows successful operation of our propagation methodology upon map reinitialization. Only the y component of p_v^w changes by about 1 m. This is due to MAV movement in the y direction between map-loss detection and reinitialization. Note that without this propagation, the new map would show random values for p_v^w, q_v^w, and λ after reinitialization.

enforced in the computation, since the key-frames kept in memory (15 in this case) are far fewer than what is necessary to detect a loop closure. These 15 key-frames cover approximately 10 m of the trajectory at the flight height of 13 m above ground in this flight, i.e., maintaining only a very local map of the MAV's workspace. The estimated visual attitude drift q_v^w in roll and pitch as well as the estimated visual scale drift λ are depicted in Figure 34 (lower plots). These values estimated by our framework render the navigation frame gravity-aligned and metric throughout the whole mission. To our knowledge, this is the longest path flown by a micro helicopter to date using vision-based navigation (no GPS or external tracker).

In Section 3.3, we discussed our approach to propagate the visual scale and pose during a map loss to the next, reinitialized map in order to minimize the impact of such a map loss to the state estimation and controller. Figure 35 shows a map loss (i.e., a forced map reset in this case) during a flight over the disaster area. Using our scale and pose propagation approach, all map properties p_v^w, q_v^w, and λ remain after a reinitialization. Only the y value of the visual position drift p_v^w has a difference of about 1 m. The reason for this difference is the distance traveled between the time instants of detecting the map-failure and reinitializing the new map. We only use GPS feeds to navigate the MAV during this short period between failure and reinitialization. In Weiss et al. (2012b), we showed that this would also be possible by using optical flow cues instead of GPS. GPS was also used after the mission for data processing to estimate the unobservable map properties (i.e., the map's position offset p_v^w and the map's global yaw offset ${}_{\text{yaw}}q_v^w$) to obtain the graphs in Figure 35, in order to prove the successful operation of our pose propagation approach.

7. DISCUSSION

The framework presented in this article, as is evident from the experimental testbed used, brings us a step closer to autonomous vision-based navigation for agile, airborne power-on-and-go systems. Following the detailed description of the methodology and validation process, we provide insight into the significance of the contributions of this work. Namely, we discuss the essence of transforming the camera into a real-time onboard motion sensor, the rationale behind our modular sensor-fusion system, and lastly the viability of our theoretical findings for real-world tasks. Note that all the software packages for the algorithms we discuss in this work are publicly available online under ROS.[8]

7.1. Camera as a Motion Sensor

Based on a high-performing state-of-the-art key-frame-based monocular SLAM framework, we have developed a visual odometry framework with unprecedented capabilities: running at constant computational complexity of 20 Hz onboard a MAV equipped with an Atom 1.6 GHz single-core platform, while exhibiting robustness in self-similar outdoor environments (e.g., grass, asphalt, etc.). Constant computational complexity and fast execution time are key for long-term missions in unknown environments and are a great challenge that existing systems struggle to address. The proposed modifications have been motivated

[8]Our software packages for autonomous vision-based MAV navigation can be found on www.ros.org/wiki/ethzasl_ptam for the vision pipeline discussed in Section 3, and www.ros.org/wiki/ethzasl_sensor_fusion for the sensor-fusion framework discussed in Section 4.

following a thorough analysis of the strengths and weaknesses of the original framework via extensive tests. As a result, the proposed methodology has transformed this system, which was originally designed to be used only in small indoor spaces, into an onboard, real-time visual odometry framework that is sufficiently robust for large-scale real-world outdoor tasks.

7.2. Visual-inertial Power-on-and-go Systems

The abstraction of the camera from a bearing sensor (as in visual EKF SLAM) to a 6 DOF pose sensor allows us to actively introduce a gravity-aligned navigation frame, enabling long-term MAV navigation. This reveals the so-called drift-states between the vision frame and the navigation frame, rendering our approach independent of the type of visual pose estimator—we can treat the visual processing pipeline as a 6 DOF pose estimation black-box module. Presenting an extensive analysis of the behavior of our visual pose estimation methodology, we can treat the drift-states explicitly, yielding precise knowledge about the system's drifts and unobservable modes.

The transformation of the camera into a general motion sensor is a necessary step for modular sensor-fusion. The main contribution of this work is the proposition of such a modular sensor-fusion method as well as its detailed theoretical analysis and validation. Our approach differs from the state-of-the-art in that we not only aim at a pose estimate for successful MAV control, but also at the (inter) sensor calibration of the sensor-suite at run-time. This renders the system truly self-calibrating and power-on-and-go. Furthermore, our approach is modular and runs in real-time and onboard the vehicle. The continuous estimation of drift to ensure gravity alignment renders this system inherently suitable for long-term autonomous navigation in large, real-world environments.

7.3. Practical Implementation

The study in Section 4 provides a solid foundation for the observability of a nonlinear, time-continuous multisensor system for long-term MAV navigation. However, real-world systems suffer not only from error sources like linearization and discretization, but also modeling and time-synchronization errors. As a result, any shortcomings of and unrealistic assumptions in the methodology only get revealed after extensive experiments of flights in large unknown environments and for long trajectories. This has been precisely the drive behind the thorough evaluation tests performed in Section 6 on the real MAV platform, showcasing long-term flights in large, unknown, outdoor environments [quantitative indoor tests were performed in our previous work Weiss, Achtelik, Chli, and Siegwart, 2012)]. We showed that vision-based flights without any feeds from GPS or external trackers is possible, as long as we are only concerned with the local consistency of our environment (i.e., position and global yaw drift can be neglected for a similar duration of missions bounded by the MAV's battery life). As a practical example, we showed a long-distance outdoor flight of about 360 m with a final position error of 0.4% and a successful flight of a vertical trajectory including a drastic altitude change (from 0 to 70 m height), including landing.

7.4. Future Work

This work focuses on the sensor fusion of a single camera with an IMU on an aerial vehicle. A key requirement for this system to be fully observable is excitation in angular velocities and linear accelerations. A study on the nature and the amplitude of excitation required on the real system (given specific sensors and their noise values) would help to understand and improve the overall system performance during a real-world mission. Such a study would also shed light on the usability of our approach on ground vehicles, where the assumption of general motion does not necessarily apply.

8. CONCLUSION

This article presents a concise yet comprehensive summary of our work leading to a complete, vision-based state-estimation framework for Micro Aerial Vehicles, achieving unprecedented levels of autonomy in the field of MAVs. Using feeds from a monocular camera and an IMU, this framework is demonstrated to perform real-time and onboard autonomous flights in general and realistic scenarios. We discuss the work permitting the transition from a thorough theoretical analysis of the framework all the way to the implementation and deployment in real missions, focusing on the main breakthroughs making this possible:

- Monocular Visual Odometry: our approach renders the camera a *real-time* onboard 6 DOF motion sensor able to perform constant complexity visual odometry, while exhibiting robustness to self-similar structure such as grass or asphalt, demonstrating its suitability for long-term flights.
- Modular Sensor Fusion and Self-Calibration: estimating the vehicle state *and* the (inter) sensor calibration at run-time permits an accurate state estimation and system self-calibration over long flights. Due to the modular architecture, the method is not bound to any specific visual motion estimator, but it can be used with any 6 DOF motion inputs while maintaining the ability to detect any failure of these motion inputs. Moreover, providing the functionality for self-calibration of the onboard sensor-suite, the system is rendered truly power-on-and-go, which is essential in maintaining robustness throughout the mission, while allowing seamless augmentation with additional sensors if necessary (even of different type).

Table II. Multimedia material.

Extension	Media Type	Description
1	Video	Altitude test up to a height of 50 m with wind disturbances.
2	Video	Demonstration of the overall system performance: Altitude test up to 70 m height with subsequent landing, fast maneuvers, active disturbances, difficult lighting conditions, and about 36-m-long flight Achtelik et al. (2012).

- Robustness and Deployability: through extensive tests using a real platform in real environments, we have demonstrated the power as well as the limitations of the proposed framework. Namely, we have demonstrated how to achieve autonomous flights of more than 360 m trajectory and 70 m altitude change followed by autonomous landing.

Pushing the theoretical and practical capabilities of the algorithms to the limits, this work is a step toward the deployability of autonomous MAVs in real missions. While robustness to a dynamic scene and more lengthy obstruction of sensor feeds (e.g., the presence of smoke in the field of view of the camera) are still research questions that need to be addressed, it is only with the use of a complete state-estimation framework as proposed here that higher-level tasks [e.g., obstacle avoidance and path planning Michael, Fink, and Kumar (2010); Mellinger and Kumar (2011)] can be tackled.

APPENDIX: INDEX TO MULTIMEDIA EXTENSIONS

The videos are available (Table II) as supporting information in the online version of this article.

REFERENCES

Achtelik, M., Bachrach, A., He, R., Prentice, S., & Roy, N. (2009). Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. In Proceedings of the SPIE (vol. 7332), Orlando, FL.

Achtelik, M., Lynen, S., Weiss, S., Kneip, L., Chli, M., & Siegwart, R. (2012). Visual-inertial SLAM for a small helicopter in large outdoor environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), video submission.

Achtelik, M. C., Doth, K.-M., Gurdan, D., & Stumpf, J. (2012). Design of a multi rotor MAV with regard to efficiency, dynamics and redundancy. In AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN.

Achtelik, M. W., Achtelik, M. C., Weiss, S., & Siegwart, R. (2011). Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments. In proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China.

Ahrens, S., Levine, D., Andrews, G., & How, J. (2009). Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments. In International Conference on Robotics and Automation, Kobe, Japan.

Altug, E., Ostrowski, J., & Mahony, R. (2002). Control of a quadrotor helicopter using visual feedback. In International Conference on Robotics and Automation (pp. 72–77), Washington, DC.

Armesto, L., Chroust, S., Vincze, M., & Tornero, J. (2004). Multi-rate fusion with vision and inertial sensors. In Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA.

Armesto, L., Tornero, J., & Vincze, M. (2007). Fast ego-motion estimation with multi-rate fusion of inertial and vision. International Journal on Robotics Research, 26(6), 577–589.

Artieda, J., Sebastian, J. M., Campoy, P., Correa, J. F., Mondragon, I. F., Martinez, C., & Olivares, M. (2009). Visual 3-D SLAM from UAVs. Journal of Intelligent and Robotic Systems. 55(4-5), 299–321.

Bachrach, A., He, R., & Roy, N. (2009a). Autonomous flight in unknown indoor environments. International Journal of Micro Air Vehicles. 1(2009), 217–228.

Bachrach, A., He, R., & Roy, N. (2009b). Autonomous flight in unstructured and unknown indoor environments. In European Conference on Micro Aerial Vehicles (EMAV), Delft, Netherlands.

Baldwin, G., Mahony, R., & Trumpf, J. (2009). A nonlinear observer for 6 DOF pose estimation from inertial and bearing measurements. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan.

Bay, H., Ess, A., Tuytelaars, T., & Gool, L. V. (2008). SURF: Speeded up robust features. Computer Vision and Image Understanding, 110(3), 346–359.

Beder, C., & Steffen, R. (2006). Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. Number 4174 in LNCS. Springer.

Blösch, M., Weiss, S., Scaramuzza, D., & Siegwart, R. (2010). Vision based MAV navigation in unknown and unstructured environments. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK.

Brockers, R., Susca, S., Zhu, D., & Matthies, L. (2012). Fully self-contained vision-aided navigation and landing of a micro air vehicle independent from external sensor inputs (pp. 83870Q–83870Q–10).

Chen, J., & Dawson, D. (2006). UAV tracking with a monocular camera. In Conference on Decision and Control, San Diego, CA.

Cheviron, T., Hamel, T., Mahony, R., & Baldwin, G. (2007). Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of UAV. In International Conference on Robotics and Automation (pp. 2010–2016), Roma, Italy.

Chroust, S., & Vincze, M. (2004). Fusion of vision and inertial data for motion and structure estimation. Journal of Robotic Systems, 21(2), 73–83.

Corke, P. (2004). An inertial and visual sensing system for a small autonomous helicopter. International Journal of Robotics Systems, 21(2), 43–51.

Corke, P., Lobo, J., & Dias, J. (2007). An introduction to inertial and visual sensing. International Journal of Robotics Research, 26(6), 519–535.

Davison, A. J., Molton, N. D., Reid, I., & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(6), 1052–1067.

Engel, J., Sturm, J., & Cremers, D. (2012). Camera-based navigation of a low-cost quadrocopter. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference (pp. 2815–2821), Vilamoura, Portugal.

Eudes, A., & Lhuillier, M. (2009). Error propagations for local bundle adjustment (vol. 0). IEEE Computer Society, Los Alamitos, CA.

Fraundorfer, F., Heng, L., Honegger, D., Lee, G., Meier, L., Tanskanen, P., & Pollefeys, M. (2012). Vision-based autonomous mapping and exploration using a quadrotor MAV. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference (pp. 4557–4564), Vilamoura, Portugal.

Gemeiner, P., Einramhof, P., & Vincze, M. (2007). Simultaneous motion and structure estimation by fusion of inertial and vision data. The International Journal of Robotics Research, 26(6), 591–605.

Hamel, T., Mahony, R., & Chriette, A. (2002). Visual servo trajectory tracking for a four rotor VTOL aerial vehicle. In International Conference on Robotics and Automation (pp. 2781–2786), Vilamoura, Portugal.

Herisse, B., Hamel, T., Mahony, R., & Russotto, F.-X. (2012). Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow. IEEE Transactions on Robotics, 28(1), 77–89.

Hermann, R., & Krener, A. (1977). Nonlinear controllability and observability. IEEE Transactions on Automatic Control, 22(5), 728–740.

How, J., Bethke, B., Frank, A., Dale, D., & Vian, J. (2008). Real-time indoor autonomous vehicle test environment. IEEE Control Systems Magazine.

Huster, A., Frew, E. W., & Rock, S. M. (2002). Relative position estimation for AUVs by fusing bearing and inertial rate sensor measurements. In Proceedings of the Oceans Conference (vol. 3, pp. 1857–1864), Biloxi: MTS/IEEE.

Jones, E. (2009). Large scale visual navigation and community map building. Ph.D. thesis, University of California at Los Angeles.

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

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

Klein, G., & Murray, D. W. (2007). Parallel tracking and mapping for small AR workspaces. International Symposium on Mixed and Augmented Reality (ISMAR), Nara, Japan.

Klose, S., M., J. W., G., A., Holzapfel, P. F., & Knoll, A. (2010). Markerless, vision-assisted flight control of a quadrocopter. In IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan.

Leutenegger, S., Chli, M., & Siegwart, R. (2011). BRISK: Binary Robust Invariant Scalable Keypoints. IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain.

Lupashin, S., Schoellig, A., Sherback, M., & D'Andrea, R. (2010). A simple learning strategy for high-speed quadrocopter multi-flips. In International Conference on Robotics and Automation, Ankorage, AK.

Lupton, T., & Sukkarieh, S. (2008). Removing scale biases and ambiguity from 6DoF monocular SLAM using inertial. In International Conference on Robotics and Automation, Pasadena, CA.

Lupton, T., & Sukkarieh, S. (2009). Efficient integration of inertial observations into visual SLAM without initialization. In IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO.

Lynen, S. (2011). Improving PTAM to allow operation in large scale environments. Master Thesis, Autonomous Systems Lab, ETH Zurich.

Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA.

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

Maybeck, P. S. (1979). Stochastic Models, Estimation and Control (vol. 1). New York: Academic Press.

Mellinger, D., & Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In Robotics and Automation (ICRA), 2011 IEEE International Conference (pp. 2520–2525), Shanghai, China.

Mellinger, D., Michael, N., & Kumar, V. (2010). Trajectory generation and control for precise aggressive maneuvers with quadrotors. In International Symposium on Experimental Robotics (ISER), New Delhi, India.

Mellinger, D., Shomin, M., Michael, N., & Kumar, V. (2010). Cooperative grasping and transport using multiple quadrotors. In International Symposium on Distributed Autonomous Robotic Systems (DARS), Lausanne, Switzerland.

Michael, N., Fink, J., & Kumar, V. (2010). Cooperative manipulation and transportation with aerial robots. Autonomous Robots.

Michael, N., Mellinger, D., Lindsey, Q., & Kumar, V. (2010). The grasp multiple micro UAV testbed. IEEE Robotics and Automation Magazine.

Mirzaei, F., & Roumeliotis, S. (2007). 1–a Kalman filter-based algorithm for IMU-camera calibration. In Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference (pp. 2427–2434), San Diego, CA.

Mirzaei, F., & Roumeliotis, S. (2008). A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. IEEE Transactions on Robotics, 24(5), 1143–1156.

Mourikis, A. I., & Roumeliotis, S. I. (2007). A multi-state constraint Kalman filter for vision-aided inertial navigation. In Proceedings of The IEEE International Conference on Robotics and Automation, Rome.

Mourikis, A. I., Trawny, N., Roumeliotis, S. I., Johnson, A. E., Ansar, A., & Matthies, L. (2009). Vision-aided inertial navigation for spacecraft entry, descent, and landing. IEEE Transactions on Robotics, 25(2), 264–280.

Park, S., Won, D., Kang, M., Kim, T., Lee, H., & Kwon, S. (2005). Ric (robust internal-loop compensator) based flight control of a quad-rotor type UAV. In International Conference on Intelligent Robots and Systems (pp. 3542–3547), Edmonton, CA.

Pinies, P., Lupton, T., Sukkarieh, S., & Tardós, J. D. (2007). Inertial aiding of inverse depth SLAM using a monocular camera. IEEE International Conference on Robotics and Automation (ICRA), Roma, Italy.

Proctor, A., & Johnson, E. (2004). Vision-only aircraft flight control methods and test results. In AIAA Guidance, Navigation, Control Conference, Providence, RI.

Qian, G., Chellappa, R., & Zheng, Q. (2002). Bayesian structure from motion using inertial information. In International Conference on Image Processing, Rochester, NY.

Roumeliotis, S. I., Johnson, A. E., & Montgomery, J. F. (2002). Augmenting inertial navigation with image-based motion estimation. In Proceedings of The IEEE International Conference on Robotics and Automation (pp. 4326–4333), Washington, D.C.

Schmid, K., Ruess, F., Suppa, M., & Burschka, D. (2012). State estimation for highly dynamic flying systems using key frame odometry with varying time delays. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference (pp. 2997–3004), Vilamoura, Portugal.

Sibley, G., Mei, C., Reid, I., & Newman, P. (2009). Adaptive relative bundle adjustment. In Proceedings of Robotics: Science and Systems (pp. 976–982).

Strasdat, H., Davison, A., Montiel, J., & Konolige, K. (2011). Double window optimisation for constant time visual slam. In Computer Vision (ICCV), 2011 IEEE International Conference (pp. 2352–2359), Barcelona, Spain.

Strasdat, H., Montiel, J. M. M., & Davison, A. J. (2010). Real-time monocular SLAM: Why filter? In IEEE International Conference on Robotics and Automation (ICRA), Ankorage, AK.

Strelow, D. (2004). Motion estimation from image and inertial measurements. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Strelow, D., & Singh, S. (2003). Online motion estimation from image and inertial measurements. In Workshop on Integration of Vision and Inertial Sensors (INERVIS), Coimbra, Portugal.

Trawny, N., & Roumeliotis, S. (2005). Indirect Kalman filter for 3d attitude estimation. Technical report, University of Minnesota, Department of Computing Science and Engineering.

Valenti, M., Bethke, B., Fiore, G., & How, J. P. (2006). Indoor multivehicle flight testbed for fault detection, isolation and recovery. In AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO.

Weiss, S. (2012). Vision based navigation for micro helicopters. Ph.D. thesis, ETH Zurich.

Weiss, S., Achtelik, M. W., Chli, M., & Siegwart, R. (2012). Versatile distributed pose estimation and sensor self-calibration for an autonomous MAV. In IEEE International Conference on Robotics and Automation (ICRA), Minneapolis, MN.

Weiss, S., Achtelik, M. W., Lynen, S., Chli, M., & Siegwart, R. (2012b). Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In IEEE International Conference on Robotics and Automation (ICRA), Minneapolis, MN.

Weiss, S., Scaramuzza, D., & Siegwart, R. (2011). Monocular-SLAM based navigation for autonomous micro helicopters in GPS-denied environments. Journal of Field Robotics, 28(6), 854–874.

Weiss, S., & Siegwart, R. (2011). Real-time metric state estimation for modular vision-inertial systems. In IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China.

Wenzel, K. E., Masselli, A., & Zell, A. (2010). Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle. In UAV'10 3rd International Symposium on Unmanned Aerial Vehicles, Dubai, UAE.

Wu, A., Johnson, E., Kaess, M., Dellaert, F., & Chowdhary, G. (2013). Autonomous flight in GPS-denied environments using monocular vision and inertial sensors. Journal of Aerospace Computing, Information and Communication, 10, 172–186.