Real-time Onboard Visual-Inertial State Estimation and Self-Calibration of MAVs in Unknown Environments

Real-time Onboard Visual-Inertial State Estimation and Self-Calibration of MAVs in Unknown Environments

Stephan Weiss, Markus W. Achtelik, Simon Lynen, Margarita Chli, Roland Siegwart

***Abstract*— The combination of visual and inertial sensors has proved to be very popular in robot navigation and, in particular, Micro Aerial Vehicle (MAV) navigation due the flexibility in weight, power consumption and low cost it offers. At the same time, coping with the big latency between inertial and visual measurements and processing images in real-time impose great research challenges. Most modern MAV navigation systems avoid to explicitly tackle this by employing a ground station for off-board processing.**

In this paper, we propose a navigation algorithm for MAVs equipped with a single camera and an Inertial Measurement Unit (IMU) which is able to run *onboard* and in *real-time*. The main focus here is on the proposed speed-estimation module which converts the camera into a metric body-speed sensor using IMU data within an EKF framework. We show how this module can be used for full self-calibration of the sensor suite in real-time. The module is then used both during initialization and as a fall-back solution at tracking failures of a keyframe-based VSLAM module. The latter is based on an existing high-performance algorithm, extended such that it achieves scalable 6DoF pose estimation at constant complexity. Fast onboard speed control is ensured by sole reliance on the optical flow of at least two features in two consecutive camera frames and the corresponding IMU readings. Our nonlinear observability analysis and our real experiments demonstrate that this approach can be used to control a MAV in speed, while we also show results of operation at 40Hz on an onboard Atom computer 1.6 GHz.

I. Introduction

A. Visual-Inertial based airborne navigation

The combination of visual and inertial sensors for effective control and navigation for Micro Aerial Vehicles (MAVs) has been shown to be a viable and increasingly popular approach. However, many implementations still rely on artificial features [1] or heavier and costly sensors like laser scanners [2]. Moreover, in order to manage the rich information of the visual data online, current visual-inertial systems ([3], [4], [5]) employ off-board processing units, limiting the robustness and flexibility of such navigation solutions. Even in outdoor scenarios where GPS might be available, it is unrealistic to assume regular unobstructed GPS reception or a permanent communication link to a ground station.

A carefully selected sensor suite can be used to increase the autonomy of a MAV and hence improve robustness of navigation. As a rule of thumb, every 10 grams require 1 W of motor power in hover mode for a small helicopter. Our camera-IMU (Inertial Measurement Unit) setup weights about 20 grams and provides the capability of real-time onboard control of the MAV without the need for artificial landmarks. At this point, it is worth mentioning that on an airborne vehicle, estimating a goal vector for control is not sufficient. Unlike ground vehicles, airborne vehicles cannot simply hold all actuators still to achieve zero velocity. Hence, for robust MAV control, a timely estimate of its actual state is a requirement – then, based on this estimate, goal vectors can be applied. As a result, this work will focus on the prompt and consistent availability of the MAV state estimate, allowing goal vectors to be generated by simply moving setpoints or holding a setpoint for hover mode. We do not focus on the control itself in this paper but refer to our previous work [6].

B. Inter-sensor calibration

As in any multi-sensor system, the camera-IMU calibration is crucial to the robustness of our estimation processes. While we assume the intrinsic camera parameters to be known and fixed, the inter-sensor calibration parameters describing the 6DoF pose between the camera and the IMU are unknown. There exist various methods in the literature to calibrate these unknowns [7], [8]. However, they usually address off-line calibration exhibiting complexity of at least $O(M^2)$ for M number of features observed by the camera. Here, we aim at a *power-on-and-go* system which calibrates itself while flying, thus computationally complex methods are unsuitable. Instead, our non-linear observability analysis in [9] reveals that we can decouple our vision algorithm by treating the arbitrarily scaled 3D camera speed or 6Dof pose as measurements. Since these measurements have constant size (3 or 6 dimensions, respectively) our state estimator which is also responsible for the inter-sensor calibration has constant complexity.

C. Overall navigation framework

The proposed navigation framework consists of two complementary visual measurement modules: a 3D speed estimator and a 6DoF pose estimator. Speed estimation is used to initialize the pose estimator by ensuring an appropriately wide baseline to start building a map of the unknown environment in a keyframe-based VSLAM (Visual Simultaneous Localization And Mapping) scheme. While we have recently seen some very successful monocular VSLAM systems [10], [11], it is PTAM [12] that has been the most popular across

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 231855 (sFly). Stephan Weiss, Markus Achtelik and Simon Lynen are currently PhD students and Margarita Chli is a senior researcher at the ETH Zurich (email: {stephan.weiss, markus.achtelik, simon.lynen, margarita.chli}@mavt.ethz.ch). Roland Siegwart is full professor at the ETH Zurich and head of the Autonomous Systems Lab (email: r.siegwart@ieee.org).

the Robotics literature due to the free availability of the authors' implementation and the nature of the algorithm: its keyframe-based representation of past experience allows great freedom in adjusting the desired accuracy-complexity ratio. As a result, our pose estimator is based on PTAM, which we improved with respect to robustness on self-similar structure and computational complexity.

While the pose estimator relies on a feature map (thus prone to lose it and fail), our inertial-optical flow based speed estimator is map and feature-history independent. This makes the speed estimator much more robust against failures. Hence, whenever the SLAM map is lost, the speed-estimator is used as a fall-back solution. Since only speed is being calculated during a (re-)initialization phase, the position estimate is prone to drift. In order to avoid this drift, we exploit the dependency of optical flow with the ratio of speed and feature distance to recover the metric scene depth. Locking on at least 3 features prevents the MAV from position drift.

Processing the visual information is the most computationally demanding part of the speed estimation. In theory, to estimate the camera speed, it is enough to use the optical flow of two features in two consecutive camera frames and the corresponding IMU readings – i.e. we do not need to store a history of any features/measurements such as keyframes. Employing optical flow in visual-inertial tracking is increasingly popular but existing systems either demonstrate results in simulation [3], [13] or transmit images to a ground station for off-board processing [4], [5].

Here, we demonstrate real, successful MAV navigation at 40 Hz while all processing is done onboard (on an Atom computer 1.6 GHz). Moreover, we present a novel inertial-optical flow based metric speed estimation algorithm which not only provides a metric state estimate but is also capable of determining the visual scale factor and allows full, online calibration of the MAV's sensor suite (including IMU biases and inter-sensor states).

II. Camera as a Metric Speed Sensor

This section describes our inertial-optical flow framework for metric speed estimation of a self-calibrating camera-IMU setup. Firstly, we detail the pure vision part which is based on the continuous 8-point algorithm [14] augmented with IMU readings. This addition drastically reduced dimensionality. In a next step, we focus on the semi-tightly coupling of the vision part with an Extended Kalman Filter (EKF) framework. We apply a non-linear observability analysis to prove the observability of the visual scale as well as the inter-sensor calibration between camera and IMU.

A. Our Reduced, Continuous 8-Point Algorithm

1) Recovering the Velocity Direction: The (discrete) epipolar constraint is $\vec{x'}^T E\vec{x} = 0$ where $\vec{x}$ is the feature direction vector and the essential matrix $E(T, R)$ consists of the camera rotation R and translation T. In continuous space, we can calculate the translational and angular velocities v and ω. For each 3D point's coordinates $\mathbf{X}(t)$ the following holds:

$$\dot{\mathbf{X}}(t) = \lfloor \vec{\omega}(t) \rfloor \mathbf{X}(t) + \vec{v}(t) \tag{1}$$

Introducing the arbitrary scale factor λ yields $\lambda(t) * \vec{x}(t) = \mathbf{X}(t)$. We substitute $\dot{\vec{x}}$ by $\vec{u}$ as the optical flow vector and obtain the following continuous epipolar constraint:

$$\vec{u}^T \lfloor \vec{v}(t) \rfloor \vec{x} + \vec{x}^T \lfloor \vec{\omega}(t) \rfloor \lfloor \vec{v}(t) \rfloor \vec{x} = 0 \tag{2}$$

with the skew symmetric notation of a vector cross product $\lfloor \vec{a} \rfloor \vec{b} = \vec{a} \times \vec{b}$. As mentioned in [14], in contrast to the discrete version, solving for the continuous essential matrix yields a unique solution for $\vec{v}$ and $\vec{\omega}$ as the twisted-pair ambiguity is avoided. Moreover, the continuous approach handles well zero-baseline situations avoiding the singularities of the discrete case.

The system in (2) requires 8 features with their corresponding optical flow vectors. From the discrete version, we know that the problem actually has 5 dimensions only (3 for each of rotation and translation and -1 for the unknown scale). Here, we can also incorporate the knowledge of angular velocities from an attached IMU, bearing in mind that the IMU needs to be time-synchronized with the camera (i.e. temporal calibration, which is ensured in our setup) and also, the spatial calibration between IMU and camera needs to be known – the latter is addressed further on in this section. This eliminates 3 more dimensions such that only 2 dimensions remain (i.e. the direction of the velocity). Measuring the angular velocities with the IMU allows to unrotate the optical flow and allows to set ω in (2) to zero. Then the 2 dimensions of the new problem are immediately visible given that the velocity can be arbitrarily scaled. The new problem can now be formulated as:

$$\begin{aligned} \vec{u}^T \lfloor \vec{v}(t) \rfloor \vec{x} &= 0 \text{ , or equivalently:} \\ (\lfloor \vec{u}(t) \rfloor \vec{x})^T \vec{v} &= 0 \end{aligned} \tag{3}$$

Using the properties of the triple product. Equation (3) suggests that the camera speed $\vec{v}$ is orthogonal to the cross product of the optical flow $\vec{u}$ and the feature direction vector $\vec{x}$. In other words, $\vec{v}$ lies in the plane spanned by $\vec{u}$ and $\vec{x}$. Geometrically, the intersection of two such planes uniquely defines the direction of $\vec{v}$. This is already sufficient since $\vec{v}$ can be arbitrarily scaled, thus we need at least two vectors $\vec{u}$ and $\vec{x}$ to define $\vec{v}$ up to scale. Fig. 1 depicts this schematically.

With the above, we have an arbitrarily scaled visual speed by only observing at least 2 features and their current optical flow. Since any two features in a non-degenerate configuration yield this information, we do not need to store any feature history but instead solely depend on the last two camera frames.

2) Recovering a Unifying But Arbitrary Scale Factor: Above, we recovered the camera speed up to an arbitrary scale. Without loss of generality we can set $|\vec{v}| = 1$. We can then rewrite (1) as

$$\dot{\mathbf{X}}(t) = \lfloor \vec{\omega}(t) \rfloor \mathbf{X}(t) + \eta \vec{v}(t) \ , \tag{4}$$

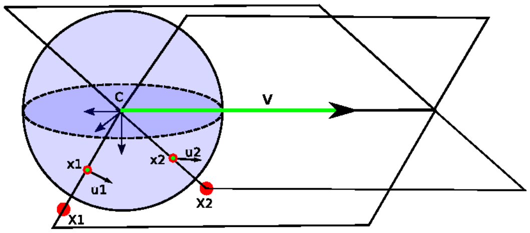

Fig. 1. Graphical setup for the continuous epipolar constraint given IMU measurements for the rotational velocities. The constraint simplifies to (3) such that only 2DoF (i.e. the direction of the velocity vector $\vec{v}$) remain. The equation suggests that this direction is in the plane spanned by a feature direction vector $\vec{x}$ and its optical flow $\vec{u}$. In other words, the triple product of these 3 vectors vanishes. Using a second feature X_2 and intersecting this second plane with the first, uniquely defines the speed direction vector $\vec{v}$.

with η being an arbitrary scale factor for the recovered unit norm velocity. We still unrotated the optical flow and have thus $\omega = 0$. Introducing the feature scales λ_i per feature i lets us rewrite (4) as

$$\dot{\lambda}_i(t)\vec{x}_i(t) + \lambda_i(t)\dot{\vec{x}}_i(t) = \eta\vec{v}(t) \ . \qquad (5)$$

We can stack all scale factors λ_i, their temporal derivatives $\dot{\lambda}_i$ and the common scale factor η for the velocity $\vec{v}$ into a single vector $\vec{\lambda} = [\lambda_1, \dots, \lambda_n, \dot{\lambda}_1, \dots, \dot{\lambda}_n, \eta]$. Since (5) is linear in $\vec{\lambda}$ we can write it as

$$M\vec{\lambda} = 0 \ , \qquad (6)$$

with M depending on the unit scaled velocity $\vec{v}$, the feature direction vectors x_i and their optical flow $\dot{x}_i$. Note that (6) unifies the scale factors to be consistent with each other in this particular camera reading. That is, λ_i of feature i is larger than λ_j of feature j if feature i is further away than feature j. We can pose this formally, as $\vec{\Lambda} = L\vec{\lambda}$ with $\vec{\Lambda}$ being the true metric scale factor and L the arbitrary but unifying scale factor by which our visual estimate is scaled.

In [14] the authors use the solution of (6) to find a globally and temporally consistent scale factor for subsequent camera frames (i.e. temporal scale propagation). In our framework, this task is handled by the underlying EKF framework considering also the IMU readings.

It is important to note that with the same feature direction vectors x_i and optical flow $\dot{x}_i$, (6) and (3) will yield the same η and $\vec{v}$. From the definition of optical flow, the same readings for x_i and $\dot{x}_i$ can only occur if the ratio between the (metric) velocity and distance to the scene is the same. This is the case when the camera moves slowly close to the scene or fast far away from the scene. Thus, the metric scale factor L changes since in reality we have change in metric speed but η and $\vec{v}$ remain unchanged. On the other hand, if the distance to the scene remains constant, η varies but L remains unchanged. Thus L may be considered as an average indicator for the scene depth which changes as slow as the scene depth changes. However, this average depends on the current scene structure which we do not use in this sense. We will make use of it and the definition of optical flow later in this section tackling the issue of the MAV's metric position w.r.t. a given set of features.

B. Semi-Tightly EKF-Based Coupling Of Camera and IMU

The unified, scaled camera speed $z_v = \eta\vec{v}$ is treated as a measurement to an EKF framework featuring a camera and an IMU. In the following, we describe very briefly the setup of the filter and then focus on its observability.

1) Filter State Definition and Propagation Equations: The filter setup concerning state definitions and propagation equations is very close to the one presented in our previous work [9] for a 6DoF pose measurement of a monocular SLAM algorithm.

We assume that the inertial measurements contain a certain bias b and white Gaussian noise n. Thus, for the real angular velocities ω and the real accelerations a we have

$$\omega = \omega_m - b_\omega - n_\omega \ , \qquad a = a_m - b_a - n_a, \qquad (7)$$

where m denotes the measured value. The dynamics of the non-static biases b are modeled as a random process:

$$\dot{b}_\omega = n_{b_\omega} \ , \qquad \dot{b}_a = n_{b_a}. \qquad (8)$$

The state of the filter is composed of the position p^i_w of the IMU in the world frame W, its velocity v^i_w, and its attitude quaternion q^i_w describing a rotation from the world frame W into the IMU frame I. We also add the gyro and acceleration biases b_ω and b_a as well as the visual scale factor L. The calibration states are the rotation from the IMU frame into the camera frame q^c_i, and the position of the camera center in the IMU frame p^c_i. This yields a 24-element state vector:

$$X = \{p^i_w \ v^i_w \ q^i_w \ b_\omega \ b_a \ L \ p^c_i \ q^c_i\} \ . \qquad (9)$$

The following differential equations govern the state:

$$\begin{aligned} \dot{p}^i_w &= v^i_w & (10)\\ \dot{v}^i_w &= C^T_{(q^i_w)}(a_m - b_a - n_a) - g & (11)\\ \dot{q}^i_w &= \frac{1}{2}\Omega(\omega_m - b_\omega - n_\omega)q^i_w & (12) \end{aligned}$$

$$\dot{b}_\omega = n_{b_\omega}, \quad \dot{b}_a = n_{b_a}, \quad \dot{L} = n_L, \quad \dot{p}^c_i = 0, \quad \dot{q}^c_i = 0, \quad (13)$$

where $C_{(q)}$ 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 to drift slowly and model this as a random walk with noise n_L. We design the filter in its error states and use the discretized propagations as described in [9].

2) Filter Update and Coupling with Optical Flow Measurements: In order to unrotate the optical flow vectors we integrate the IMU's gyroscope readings between two camera frames and calculate the relative rotation using a first-order quaternion integration. It is important to note the following:

- The initialization for the gyroscope biases as filter states is accurate. In contrast to the accelerometer biases which interfere with the MAV's initial attitude, the gyroscope biases can directly be measured at the initial phase where the MAV stands still.
- The change of the gyroscope biases over time is tracked by the filter as a filter state – i.e. the quaternion

integration with the gyroscope readings is done after statistically optimal bias compensation.

- Both IMU and camera devices are time synchronized.

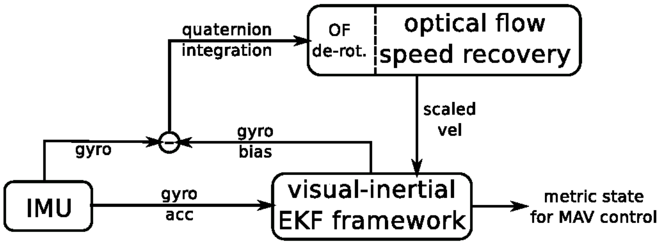
quaternion integration
OF de-rot.
optical flow speed recovery
scaled vel
gyro
gyro bias
IMU
gyro acc
visual-inertial EKF framework
metric state for MAV control

Fig. 2. System setup for our semi-tightly coupling of inertial-optical flow based speed recovery. The IMU is used as a prediction model in the EKF propagation phase, while the gyroscopes are used in a first-order quaternion integration to recover the relative rotation between two camera frames. The gyroscopic measurements are bias-compensated using the estimated bias term in the filter. The visual part unrotates the optical flow measurements using the relative rotation such that we can apply (3) to determine the arbitrarily scaled camera velocity to be used in the EKF update.

The whole framework, depicted in Fig. 2, is a mixture of sensor colligation (the IMU is used to unrotate the optical flow vectors) and statistical fusion (EKF framework). The camera velocity measurement can be described as:

$$z_v = (C_{(q_i^c)}C_{(q_w^i)}v_w^i + C_{(q_i^c)}(\lfloor\omega\rfloor p_i^c))L + n_v \ , \qquad (14)$$

where $C_{(q_w^i)}$ is the IMU's attitude and $C_{(q_i^c)}$ is the rotation between the IMU and camera. This can be linearized to $\hat{z}_v = HX$, so once the measurement matrix H is obtained, our estimate can be updated following the standard EKF procedure:

1) compute the residual $r = z_v - \hat{z}_v$
2) compute the innovation $S = HPH^T + R$
3) compute the Kalman gain $K = PH^T S^{-1}$
4) compute the correction $\hat{x} = Kr$
5) update the covariance matrix
$P_{k+1|k+1} = (\mathbf{I_d} - KH)P_{k+1|k}(\mathbf{I_d} - KH)^T + KRK^T$.

C. Observability Analysis

Summarizing the propagation equations in a control affine form:

$$\begin{bmatrix} \dot{p}_w^i \\ \dot{v}_w^i \\ \dot{q}_w^i \\ \dot{b}_\omega \\ \dot{b}_a \\ \dot{L} \\ \dot{p}_i^c \\ \dot{q}_i^c \end{bmatrix} = \underbrace{\begin{bmatrix} v_w^i \\ -C^T_{(q_w^i)}b_a - g \\ 0.5\Xi_{(q_w^i)}b_\omega \\ 0_{3\times1} \\ 0_{3\times1} \\ 0 \\ 0_{3\times1} \\ 0_{4\times1} \end{bmatrix}}_{f_0} + \underbrace{\begin{bmatrix} 0_{3\times3} \\ 0_{3\times3} \\ 0.5\Xi_{(q_w^i)} \\ 0_{3\times3} \\ 0_{3\times3} \\ 0_{1\times3} \\ 0_{3\times3} \\ 0_{4\times3} \end{bmatrix}}_{f_1} \omega_m + \underbrace{\begin{bmatrix} 0_{3\times3} \\ C^T_{(q_w^i)} \\ 0_{4\times3} \\ 0_{3\times3} \\ 0_{3\times3} \\ 0_{1\times3} \\ 0_{3\times3} \\ 0_{4\times3} \end{bmatrix}}_{f_2} a_m \qquad (15)$$

and the measurement equations

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} (C_{(q_i^c)}C_{(q_w^i)}v_w^i + C_{(q_i^c)}(\lfloor\omega\rfloor p_i^c))L \\ {q_w^i}^T q_w^i \\ {q_i^c}^T q_i^c \end{bmatrix} , \qquad (16)$$

with h_2 and h_3 being the constraints of unit norm quaternions of rotation.

We denote the gradient w.r.t. the state variables of the zero order Lie derivative of h_n as $\nabla L^0 h_n$, and its first order derivative w.r.t. f_m as $\nabla L^1_{f_m} h_n$. Following the suggestions in [9], [7], [8], we obtain the observability matrix $\mathcal{O}$:

$$\mathcal{O} = \begin{bmatrix} \nabla L^0 h_1 \\ \nabla L^0 h_2 \\ \nabla L^0 h_3 \\ \nabla L^1_{f_0} h_1 \\ \nabla L^1_{f_1} h_1 \\ \nabla L^1_{f_2} h_1 \\ \nabla L^2_{f_0 f_0} h_1 \\ \nabla L^2_{f_1 f_0} h_1 \\ \nabla L^2_{f_2 f_0} h_1 \end{bmatrix} .$$

A simple rank condition calculation reveals rank deficiency (rank 20 instead of 24). An analysis of the continuous symmetries as proposed in [15] reveals that the position states and the yaw state of the IMU attitude w.r.t. to the world reference frame are not observable. This is not surprising, since we only have camera velocities as measurement and thus no information about the absolute position and yaw orientation. The absolute roll and pitch angle of the IMU w.r.t. to the world frame is made observable by the IMU's gravity measurement. Moreover, the analysis shows, that not only the visual scale factor L is observable, but also all 6DoF of the inter-sensor calibration states between IMU and camera. Obviously, L and the distance between IMU and camera p_i^c have to be non-zero to be observable. Additionally, and less obvious, the system needs non-zero accelerations and non-zero angular velocities in at least 2 axes to render the mentioned states observable. This analysis is not the focus of this paper. The general approach for such an analysis is described in detail in [7].

The fact that only the position p_w^i and yaw part of the MAV's attitude q_w^i is unobservable suggests that the MAV is still able to self-calibrate its sensor suite and perform metric speed control. However, it will slowly drift in position and yaw direction. This is nevertheless sufficient for local stabilization of the MAV. We discuss in the next section how such a local short-term stabilization is sufficient to initialize a more elaborate monocular SLAM algorithm for full MAV control.

D. Towards inertial-optical flow based position control

Our focus in this paper is to have a robust visual algorithm not relying on the re-detection of certain features or any sort of feature history. However, we shortly highlight the possible extension of our inertial-optical flow speed recovery approach in order to eliminate position drift.

If we assume known camera calibration and project the camera readings to the unit sphere, we can apply the intercept theorem. Therefore, the optical flow $\dot{x}$ is defined as

$$\dot{x} = \frac{v}{D}\sin\alpha \ , \qquad (17)$$

where v is the camera velocity vector, D is the distance to the feature and α is the angle between v and the direction to the feature. Given the camera model, we can measure α (and $\dot{x}$). Hence we can add an extra measurement per observed feature including its optical flow and angle α to the EKF framework discussed above:

$$h_i = \frac{\dot{x}}{\sin\alpha} = \frac{v}{D} \ . \tag{18}$$

For one such measurement $h_4 = \frac{v}{D}$ we can define the MAV position w.r.t. a single feature as $D = \sqrt{x^2 + y^2 + z^2}$. The gradient of its Lie derivative w.r.t. to the state space yields $\nabla L^0 h_4$. For a general movement, one such measurement adds a 3×6 matrix block of rank 3 to the observability matrix $\mathcal{O}$. The entries of this matrix block are in particular non-zero at the indices of p_w^i since $D = D(p_w^i)$. This means, that (for general movement), the position of the MAV is observable and thus drift-free. In essence, we have a hyperplane that still renders the position unobservable. This hyperplane is a sphere around one feature or a circle between two features. Naturally, this is the region where D remains unchanged. In general, and as we know it from the Perspective-3-Point algorithm, we eliminate all such hyperplanes by observing 3 or more features. This means, we would need to extend our approach with a feature history of 3 or more features.

III. Efficient Visual Inertial 6DoF Pose Estimation

In the previous section we described an approach to speed-control an MAV. This approach solely depends on two consecutive camera frames (and the corresponding IMU readings), rendering it immune to failures due to map loss or feature history corruption as in visual SLAM or visual odometry systems. However, this approach is prone to position drift. In order to tackle this, we use the speed-based control as a back-up and (re-)initialization algorithm of more powerful, feature-history-based vision approaches. The latter are usually more expensive, hence an efficient solution is crucial for onboard MAV application. Here, we detail our improvements on the existing monocular keyframe-based VSLAM framework in [12], which enables onboard operation at 20 Hz on an onboard Atom computer 1.6 GHz. In Section IV, we show how to initialize this framework with the aid of our speed-based controller.

A. Real-Time Onboard Keyframe-based Monocular SLAM

As one of the most modern, high-performing systems, we choose to tailor PTAM [12] to the general needs of a computationally limited MAV platform. The framework has been ported to be compatible to the Robot Operating System[1] such that:

- the input image taken from an image node and a verification image including the features found, is published. This enables the user to handle PTAM on an embedded system without human-machine interfaces.
- the 6DoF pose is published as a pose with a covariance estimation calculated from PTAM's internal bundle adjustment.
- the visualization of camera keyframes, trajectory and features is ported to RVIZ such that visualization can be done on a ground station, if necessary.
- tuning parameters can be changed dynamically in a GUI for dynamic reconfiguration.

1) Keyframe Handling: In PTAM, the map is defined as a set of keyframes together with their observed features. In order to minimize the computational complexity, here we set a maximum number of keyframes retained in the map. If this number is exceeded, the keyframe furthest away from the current MAV pose gets deleted along with the features associated with it. If the maximum number of retained keyframes is infinite, then the algorithm is equivalent to the original PTAM, while if we set a maximum of 2 keyframes we obtain a visual odometry framework. Naturally, the larger the number of retained keyframes, the lower the estimation drift, but also the larger the computational complexity.

2) Improved Feature Handling for More Robust Maps: When flying outdoors, we experienced severe issues with self-similarity of the environment – e.g. the asphalt in urban areas or the grass in rural areas. Naturally, features extracted at higher pyramidal levels are more robust to scene ambiguity. Thus, while the finest-scale features are included for tracking, we omit them in map-handling – i.e. we only store features extracted in the highest 3 pyramidal levels. This improves tracking quality when moving away from a given feature (e.g. when taking-off with a MAV with a downward-looking camera), making it possible to navigate over both grass and asphalt.

Since this vision algorithm is keyframe-based, it has high measurement rates when tracking. However, at keyframe generation the frame-rate drops remarkably. Using only features detected at the highest pyramidal levels also reduces drastically the number of newly added features upon keyframe generation. This results to great speed-ups with keyframe-generation running at 13Hz (in contrast to the 7Hz of the original PTAM) and normal tracking rates of around 20 Hz on an onboard Atom computer 1.6 GHz.

3) Re-Initialization After Failure Mode: We use our speed-based controller described in Section II to initialize PTAM and stabilize the MAV on PTAM failures and during (re-)initialization sequences. 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) [16] in order to keep PTAM independent of IMU readings. For re-initializations, we store the median scene depth and pose of the closest keyframe and propagate this information to the new initialized map. This way we minimize large jumps in scale and pose at re-initializations.

4) Inverted Index Structure for Map-point Filtering: On each frame, PTAM projects the 3D points from the map into the current image according to the motion-model prior, which allows then point-correspondences to be established for tracking. Since no filtering on point visibility is preceding this step, it scales linearly with the number of points in the map. We implemented an inverted index structure based on

[1] www.ros.org

the grouping of map points inside keyframes which allows discarding large groups of map-points with low probability of being in the field-of-view. The search for visible points is performed by re-projecting a small set of distinct map-points from every keyframe which permits inference on their visibility from the current keyframe. The total number of points that need evaluation by reprojection is thereby significantly reduced leading to a scaling of the system in linear order of the visible keyframes rather than in linear order with the overall number of keyframes in the map.

IV. Results

The proposed approach for visual-inertial MAV speed-control is based on an EKF framework. While the observability analysis in Section II showed that the metric speed, visual scale and inter-sensor calibration parameters are observable, it is crucial to demonstrate its applicability in real scenarios in order to ensure that linearization effects are negligible. Here, we firstly present results in simulation highlighting the influence of non-observable states, followed by real data experiments obtained by handheld motion of the MAV with fully onboard computation. Finally, we assess the MAV's flight performance using the overall navigation framework proposed in this paper (speed and pose estimation).

A. Simulation Results

Table I lists the values we used for the system in simulation, during which we ensured that the motion has excitation in at least two axes in acceleration and angular velocity (as advocated in Section II). In practice, this is usually fulfilled due to the agile nature of a MAV. After convergence, the filter yields the average results listed in Table II for the inter-sensor calibration and the visual scale factor L.

TABLE I
Default simulation values

$p_i^c[m]$	$[0.1\ 0.5\ \text{-}0.04]^T$
$q_i^c[rad]$	$rpy[0.2\ \text{-}0.3\ 0.4]^T$
L	0.5
$b_a[\frac{m}{s^2}]$	$[\text{-}0.1\ \text{-}0.2\ 0.15]^T$
$b_w[\frac{rad}{s}]$	$[0.01\ 0.02\ \text{-}0.015]^T$

TABLE II
Inter-sensor calibration results on simulated data

	p_i^c[m]	q_i^c rpy[rad]	L	$b_a[m/s^2]$	$b_w[r/s]$
Average error	0.001	0.002	0.89%	0.037	$< \epsilon$
	0.002	0.003		0.033	$< \epsilon$
	0.007	0.005		0.017	$< \epsilon$

Note that the estimation of the acceleration biases is the least accurate whereas the estimation of the gyroscope biases is the most precise. This may coincide with the fact that the acceleration measurement is linked with the current attitude and motion including their uncertainties, whereas the gyroscope biases only link to the attitude change.

The RMS on the estimated pose is for v_w^i [0.044, 0.050, 0.034] $\mathrm{m/s}$ in x,y,z respectively and for q_w^i [0.024, 0.022] rad in roll and pitch respectively. Note that we do not list the RMS of the position p_w^i nor the yaw angle of the attitude q_w^i since the observability analysis showed that they are not observable.

The conducted experiment suggests that the linearization effects do not corrupt the EKF estimation. Fig. 3 depicts the state covariance matrix after convergence. The unobservability of the position (first 3 states) is clearly visible by a high uncertainty.

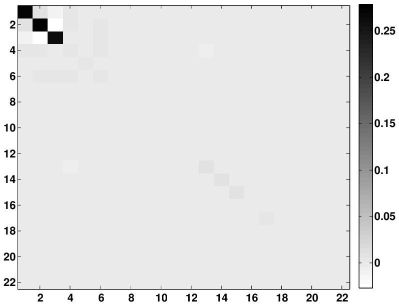

Fig. 3. State covariance matrix of the converged state. It is clearly visible that the position states p_w^i are not observable and hence they have a large uncertainty (first 3 states in the matrix). Note that the covariance matrix corresponds to the error state which represents all rotations in their minimal form (i.e. 3 elements). Thus the dimension is 22×22 only.

B. Real Experiments

1) Performance of the Inertial-Optical Flow Framework: For the experiments on a real MAV, we used a hexacopter platform provided by Ascending Technologies[2]. The platform is equipped with an IMU and a WVGA monochrome camera with global shutter. The camera frame-rate was set to 20Hz whereas the IMU provides measurements at 1kHz. As ground truth data, we use a Vicon system with mm and sub-degree accuracy. We take the temporal derivative of the Vicon position for ground truth velocity. We measured the ground truth inter-sensor calibration parameters to be $p_i^c =$ [0.015, -0.01, -0.03] m and $\mathrm{rpy}(q_i^c) =$ [0, π, 0]. Since the ground truth scale-factor is very difficult to determine, we omit direct analysis of this state. Instead, the true scale-factor is reflected in the metric ground truth velocity data. Thus, comparing the velocity estimate of the filter with ground truth implicitly yields a qualitative picture of the correct scale estimate.

For the first experiment, we moved the MAV handheld about $0.5\,\mathrm{m}$ above the ground in all Cartesian directions. The plot of the estimated speed vs. ground truth speed is plotted in Fig. 4.

Evidently, the velocity is well estimated despite some obvious outlier-updates from the visual speed measurement (e.g. after sec. 44 in x, sec. 56.5 in y, sec. 74 in z). These small outliers are sufficiently smoothed in the integrated position to not disturb the MAV position controller. Fig. 4 also shows that the scale is estimated correctly, since otherwise, the magnitude of the filter's speed-estimate would differ from the Vicon ground truth. In this experiment, we measured an RMS of [0.028, 0.035, 0.025] $\mathrm{m/s}$ in x,y,z, respectively.

[2] www.asctec.de

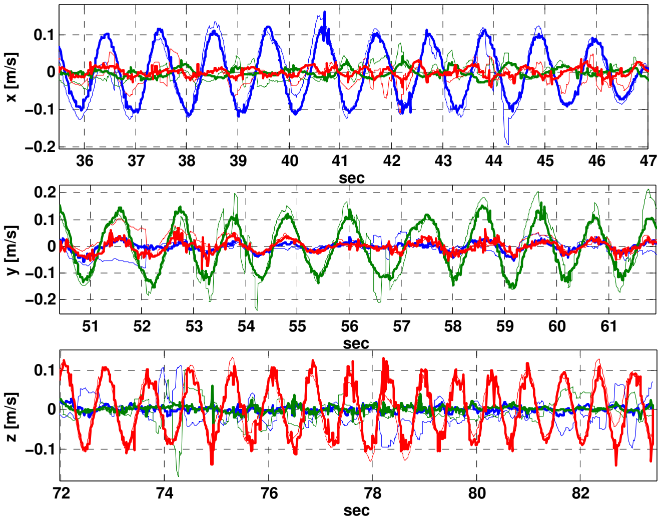

Fig. 4. Estimated MAV speed in x (blue), y (green), z (red) directions for a handheld MAV. The movements were made separately in x (top), y (middle) and z (bottom). Bold lines correspond to Vicon ground truth (noise arises from the position derivative of the Vicon data) and thin lines are the filter estimates. Notice that at points, the visual reading is corrupted, imposing a wrong update on the filter. Nevertheless, the estimates are robust with a RMS of [0.028, 0.035, 0.025] $\mathrm{m/s}$ in x, y and z, respectively.

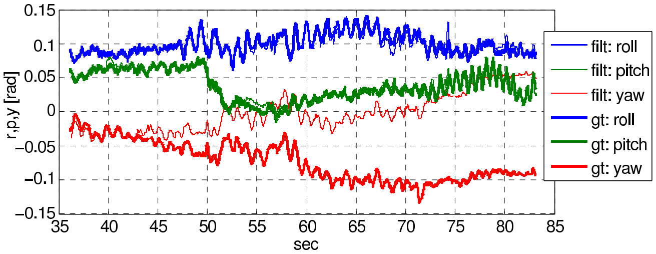

Fig. 5. Estimated MAV attitude. While roll and pitch are observable the yaw angle is not. This is clearly visible by its drift w.r.t. the ground truth. The RMS is [0.007, 0.014] rad in roll and pitch, respectively.

Fig. 5 illustrates that the roll and pitch angles are observable, while yaw is not which we derived theoretically in Section II. The sequence is the same as taken for the velocity plots in Fig. 4. The RMS to ground truth is [0.007, 0.014] rad in roll and pitch, respectively.

For the inter-sensor calibration we measured an RMS of [0.009, 0.011, 0.004] rad in roll, pitch, yaw of the attitude between camera and IMU. The results indicate, however, that our ground truth may not be precise enough to judge in detail this RMS value. We experienced the largest issue in estimating the translation between IMU and camera p_i^c. For this, we measured an RMS of [0.008, 0.005, 0.083] m in x,y,z, respectively. Given the small values for the ground truth distance, this RMS is large. We assume that the system would need more motion in order to converge better. Also, the larger the distance between the sensors, the more relevant is its influence on the measurements and thus the better can it be estimated. However, this issue needs further investigation.

In a new experiment, we let the MAV hover autonomously solely based on our inertial-optical flow approach. Fig. 6 shows the position plots. Note that the position performs a slow random walk since it is not observable, however, based on a good velocity estimate. This plot shows, that the MAV is indeed able to hover robustly long enough to perform a (re-)initialization of higher level vision algorithms such as a VSLAM framework.

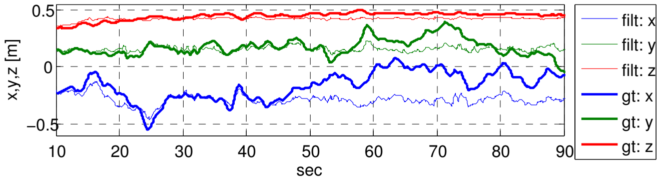

Fig. 6. Position plot of the autonomously hovering MAV. Because of the non-observability of the position state, it performs a random walk. Notice, however, that this walk is very slow due to the good estimate of the speed.

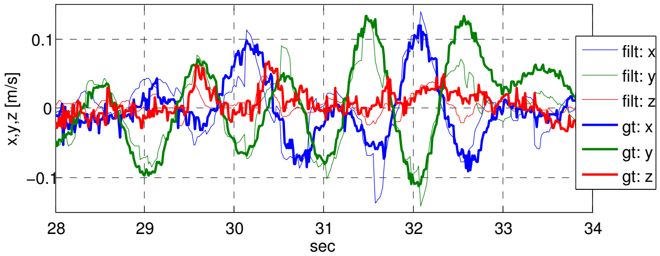

Fig. 7. Velocity of the autonomously flying MAV. The correct magnitude of the speed estimate lets us assume that the scale is estimated correctly. We notice also some erroneous vision updates (e.g. at sec. 32 in the y). The high RMS of [0.053, 0.041, 0.017] $\mathrm{m/s}$ may arise from bad feature matching and motion blur in the vision part due to vibrations during flight.

A sample of the velocity estimate of the filter versus the ground truth is plotted in Fig. 7. Again, the fitting in magnitude of the two velocities indicate a correct estimate of the scale factor. The RMS over the whole flight is [0.053, 0.041, 0.017] $\mathrm{m/s}$. We assume that the higher RMS arises from the vibrations while flying and thus possible motion blur and inaccurate feature matching.

For the attitude we measure an RMS of [0.023 0.041] rad for roll and pitch, respectively. After 80 sec of flight, the yaw drifts 0.5 rad. The inter-sensor attitude RMS is [0.045, 0.016, 0.027] rad whereas the inter-sensor distance RMS is [0.033, 0.016, 0.252] m. Again here, we justify the high RMS value for the estimate of p_i^c as due to insufficient movement and has to be investigated further in detail.

For all real experiments, we run the filter as well as the vision framework simultaneously on the onboard computer. The timings for the vision algorithm are listed in Fig. 8. The timings for the EKF framework are negligibly small since the prediction step is performed on the embedded ARM7 processor at 1 kHz and the most complex part of the update step on the onboard computer is an inversion of the 3×3 innovation matrix (since the visual velocity measurement has 3DoF only). Note that on average, our inertial-optical flow approach can run at just under 40 Hz on an onboard computer. We divided the algorithm in 3 parts: (a) Feature management ensures enough features equally spread in the image – in the case of bad feature readings, this algorithm takes longer to define suitable features. (b) Feature extraction and matching establishes correspondences in consecutive frames. (c) Visual velocity calculation using sparse SVD methods to solve (3) and (5). The big difference in timings in Fig. 8 are due to both our sparse matrix implementation and our feature prediction methods for fast feature matching.

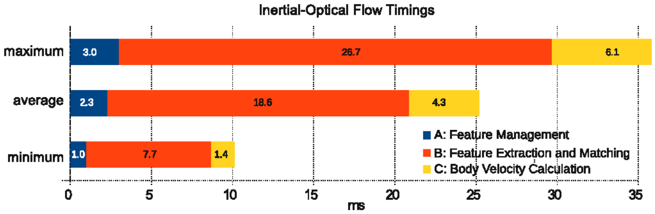

Fig. 8. Timings of our inertial-optical flow to recover the visual speed on an ATOM 1.6GHz. A corresponds to the feature management, B is the feature extraction and matching and C is the calculation of (3) and (5).

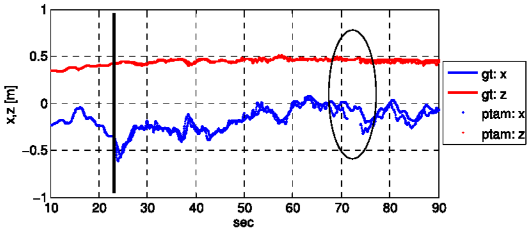

Fig. 9. (Re-)initialization of our 6DoF position estimation. We initialize it around sec. 23 while the MAV is controlled by our inertial-optical flow approach. At around sec. 71 we force a re-initialization by deleting the map. Note the small change in scale and pose – this would be undefined and arbitrarily large without using our scale and pose propagation improvement discussed in the previous section.

2) Initialization of the Improved Monocular SLAM System: We showed in Fig. 6 that our inertial-optical flow approach is capable of stabilizing the MAV. Thanks to a statistically optimal metric speed estimation, the position drift is very little (about 20 cm during the whole experiment of 80 sec)

As a final experiment, we initialize our 6DoF position-estimation SLAM module to show its capability of re-initializing in case of map-loss. Note that, in this case, we did not use the information of the position-estimation to control the MAV. This was solely done by the previously described inertial-optical flow approach. Fig. 9 shows that position estimation gets initialized at sec. 23 of the experiment. We scaled its output for better comparison with ground truth. At around sec. 71 we deliberately canceled the map, which triggered an automatic re-initialization using our improvement on scale and pose propagation. Note that after initialization the position and scale are slightly different, however, this is sufficiently accurate to position-control an MAV without having large pose jumps upon re-initialization sequences.

V. Conclusions

This paper proposes an inertial-optical flow approach capable of estimating the MAV state such that it can be robustly controlled in speed. Exploiting the full potential of the visual-inertial sensor fusion, we demonstrate both the metric 6DoF pose estimate and inter-sensor calibration with all processing onboard the MAV and in real-time. We prove theoretically the observability of the visual scale and thus the metric speed, the attitude in roll and pitch, and the full sensor calibration states (IMU biases, translation and rotation between camera-IMU). This analysis is backed up with simulated and real experiments on a real flying MAV, while our inertial-optical flow framework achieves a rate of 40 Hz on average on an onboard Atom computer 1.6 GHz.

We present a set of critical improvements on an existing monocular SLAM framework for drift-free position-control of onboard the MAV at 20 Hz. Moreover, a failure in the VSLAM framework can be bridged by (re-)initializing it online using our inertial-optical flow approach for short-term position hold, avoiding large pose jumps upon re-initialization.

VI. Acknowledgments

Stephan Weiss and Markus Achtelik would like to thank Prof. Daniela Rus and Prof. Nicholas Roy for hosting their stay in summer 2011 at CSAIL, MIT.

References

[1] J. H. Kim and S. Sukkarieh, "Airborne simultaneous localisation and map building," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003, pp. 406–411.

[2] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[3] G. Bleser and G. Hendeby, "Using optical flow for filling the gaps in visual-inertial tracking," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2010.

[4] B. Hérissé, T. Hamel, R. Mahony, and F.-X. Russotto, "A terrain-following control approach for a VTOL unmanned aerial vehicle using average optical flow," *Autonomous Robots*, vol. 29, pp. 381–399, 2010.

[5] F. Schill, R. Mahony, and P. Corke, "Estimating ego-motion in panoramic image sequences with inertial measurements," in *Robotics Research.* Springer Berlin / Heidelberg, 2011, vol. 70, pp. 87–101.

[6] M. W. Achtelik, M. C. Achtelik, S. Weiss, and R. Siegwart, "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)*, 2011.

[7] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *International Journal of Robotics Research (IJRR)*, vol. 30, no. 1, pp. 56–79, 2011.

[8] F. Mirzaei and S. Roumeliotis, "A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation," *IEEE Transactions on Robotics and Automation*, vol. 24, no. 5, pp. 1143 –1156, 2008.

[9] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[10] A. J. Davison, N. D. Molton, I. Reid, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 6, pp. 1052–1067, 2007.

[11] E. Eade, "Monocular simultaneous localisation and mapping," Ph.D. dissertation, University of Cambridge, 2008.

[12] G. Klein and D. W. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[13] J. J. L. Center and K. H. Knuth, "Bayesian visual odometry," *American Institude of Physics (AIP) Conference Proceedings*, vol. 1305, no. 1, pp. 75–82, 2011.

[14] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An invitation to 3D vision : from images to geometric models*, Springer, Ed. Springer, 2000.

[15] A. Martinelli, "State estimation based on the concept of continuous symmetry and observability analysis: The case of calibration," *Robotics, IEEE Transactions on*, vol. 27, no. 2, pp. 239 –255, april 2011.

[16] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 2, 26-may 1, 2004, pp. 1843 – 1848 Vol.2.