A Robust and Modular Multi-Sensor Fusion Approach Applied to MAV Navigation

Simon Lynen[1], Markus W. Achtelik[1], Stephan Weiss[2], Margarita Chli[1] and Roland Siegwart[1]

***Abstract*— It has been long known that fusing information from multiple sensors for robot navigation results in increased robustness and accuracy. However, accurate calibration of the sensor ensemble prior to deployment in the field as well as coping with sensor outages, different measurement rates and delays, render multi-sensor fusion a challenge. As a result, most often, systems do not exploit all the sensor information available in exchange for simplicity. For example, on a mission requiring transition of the robot from indoors to outdoors, it is the norm to ignore the Global Positioning System (GPS) signals which become freely available once outdoors and instead, rely only on sensor feeds (e.g., vision and laser) continuously available throughout the mission. Naturally, this comes at the expense of robustness and accuracy in real deployment. This paper presents a generic framework, dubbed *Multi-Sensor-Fusion Extended Kalman Filter* (MSF-EKF), able to process delayed, relative and absolute measurements from a theoretically unlimited number of different sensors and sensor types, while allowing self-calibration of the sensor-suite online. The modularity of MSF-EKF allows seamless handling of additional/lost sensor signals during operation while employing a state buffering scheme augmented with Iterated EKF (IEKF) updates to allow for efficient re-linearization of the prediction to get near optimal linearization points for both absolute and relative state updates. We demonstrate our approach in outdoor navigation experiments using a Micro Aerial Vehicle (MAV) equipped with a GPS receiver as well as visual, inertial, and pressure sensors.**

I. Introduction

Precise and consistent localization is a core problem in many areas of mobile robotics, in both research and industrial applications. Driven by the need for effective solutions, the literature is currently host to an abundance of approaches to state estimation. Addressing different choices of on-board sensor suites the employed frameworks however are tailored tightly to the task at hand. The use of GPS feeds, for example, is a common and convenient approach to localization for platforms operating in open (GPS-accessible) spaces. Conversely, in GPS-denied environments, vision or laser based approaches are often employed instead. The transition, however, across domains with different sensor-signal availability and suitability, remains a challenging problem.

In this paper, we present an effective approach to tackle the problem of seamless sensor-feed integration within state estimation. We put the focus on rotor-based Micro Aerial Vehicles (MAVs), as they are most capable of acting in and traversing across different domains, while imposing delicate challenges due to their high agility and limitations on both payload and computational power. Building on our earlier work [16], [17], we propose a highly generic, open source c++ state estimation framework which comprises:

- Modular support for an unlimited number of sensors providing relative and absolute measurements.
- Estimation of calibration states between sensors and dynamic compensation of measurement delays.
- Re-linearization of constraints from both proprioceptive and exteroceptive information sources in filter form.
- Efficient tracking of cross covariance terms for relative updates allowing estimation rates of several kHz.

Following an analysis of the limitations of our earlier work, we also present a derivation to include relative poses from key-frame based Simultaneous Localization And Mapping (SLAM) system, which is essential when employing visual/laser odometric sensors. Finally, we demonstrate the MSF-EKF framework in real experiments, flying trajectories of more than 800 m with speeds of up to 4 m/s.[1]

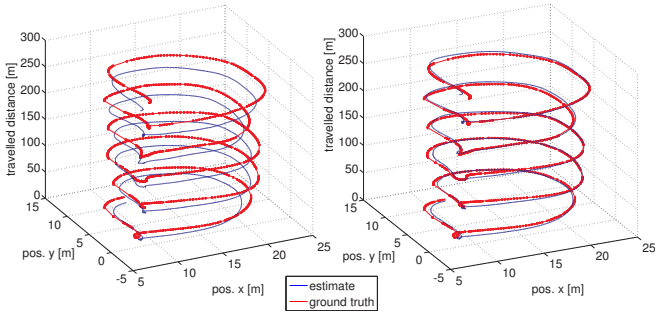

Fig. 1: The scale error of a visual SLAM system combined with our sensor-fusion framework commonly is in the area of 3-5% depending on the structure observed and the movements carried out. The left plot shows the deviations of the trajectory with the scale error not accounted for. The right plot shows potential benefits which additional sensors can provide when fusing e.g., a height sensor with visual and inertial cues.

A. Sensor Fusion for State Estimation

Autonomous MAV navigation and control has seen great success over the last couple of years, demonstrating impressive results with the aid of external motion capture systems. However, the complex preparation of the operation space required with such systems is clearly not an option in large scale missions in unknown environments. Tackling this challenge is core in enabling operation for common tasks such as industrial inspection, search-and-rescue and surveillance. As a result, a series of approaches have been proposed, using

* This work has been supported by the European Commission's Seventh Framework Programme (FP7) under grant agreements n. 285417 (ICARUS) and n.266470 (myCopter).

[1] Autonomous Systems Laboratory, ETH Zurich, Switzerland
[2] Computer Vision Group, Nasa JPL, California, USA

[1] A video of the experiments is available at http://youtu.be/neG8iEf8XiQ

on-board sensors such as laser-range finders [13], visible-light [1] or depth cameras, typically fused with readings from an Inertial Measurement Unit (IMU) to provide information about the state of the vehicle.

Most often, these state estimation approaches are designed to use a specific sensor setup for the domain space of the task at hand. While showing unmatched accuracy and consistency [5], [9] they are commonly designed for a particular sensor setup with limited modularity. Despite that GPS signals become available once the robot moves outdoors, they are often ignored in the state estimation [1], [17] as they require a more complex control strategy when moving from a local to global frame of reference. As a result, many current frameworks used on board MAVs fail to utilize all available information, limiting both accuracy and robustness of the state estimate. In [8] handling sensor outages was addressed in the context of fixed-wing aerial navigation with one position sensor, demonstrating successful state estimation during simulated temporary outage of GPS signal. In [11] GPS and visual measurements where used for different periods of the experiment. Here, we present a generic framework, which permits handling online with the effects of a multitude of different sensors. Fusing multiple sensors was also addressed recently in [6] where both relative and absolute sensors are included in a factor graph formulation using non-linear optimization. Such (fixed-lag) smoothers based on non-linear optimization have potentially higher accuracy due to the ability to re-linearize all constraints from both exteroceptive and proprioceptive information sources. However even for recent implementations [6] the computational cost is higher by two orders of magnitude compared to the framework presented here.

这里说，
多传感器
融合有两种方法
1. filter-based
2. graph-based
其实可以
在以后把这两种
框架都和
curvefusion比较
在精度，计算时间
内存方面，
当然要基于松组合
的框架下。

B. Self-Calibration of Sensors and Scale estimation

In navigation frameworks, any vehicle states essential for robot control are commonly estimated at high rates, which is especially critical for platforms like MAVs. In a typical scenario, inertial measurements arriving at rates of several 100 Hz to 2 kHz are fused with lower rate exteroceptive updates ($\sim 5 - 90$ Hz), coming from e.g., GPS or visual odometery, to mitigate drifts. Common fusion approaches are based on indirect formulations of Extended (EKF) [13], or Unscented (UKF) Kalman Filters [14]. In [10], it was shown that additional quantities of interest can be estimated in the same manner; for example, the intrinsic calibration of the proprioceptive sensors, the extrinsic calibration between proprio- and exteroceptive sensors, as well as unknown quantities from the exteroceptive-sensor process such as the scale and drifts of a monocular SLAM system. For the study of inter-sensor calibration we refer to our earlier work [17].

The accuracy of monocular visual-inertial frameworks is dominated by the correct estimation of the scale. In Fig. 1 we show the first 350 m of an 800 m flight of a MAV flying with speeds of up to 4 m/s in circles over grass. To highlight the error in scale we plotted estimate and ground-truth in the x and y directions versus the traveled distance. The left plot highlights the error in the scale estimation of about 5 % while the right plot shows the same data when the scale error is minimized. This demonstrates the potential benefits of fusing additional sources of metric information which then leads to more accurate estimates also in long range missions.

Here, we adopt this idea to achieve online self-calibration of the sensor-suite. Furthermore we adapt our framework to handle relative measurements to avoid the shortcomings of our previous work: In [16], the local map is considered as noise free which leads to an inconsistent state estimate.

C. Relative and absolute pose measurements

In [16], we discussed the un-observability of states such as the relative position and yaw between the SLAM-frame and the world-frame in a visual-inertial navigation system. This problem is commonly addressed by fixing the respective states in the estimation process and applying pose estimates from the visual SLAM algorithms as pseudo-absolute measurements [3], [14], [15].

However, it has been shown [11] that applying the relative pose estimates from a visual odometery system as pseudo-absolute measurements leads to sub-optimal estimates, as the uncertainty of the pose computed by a visual odometry system (or key frame based SLAM with a limited number of key frames) is a relative and not an absolute quantity. This leads to inconsistencies and does not allow the estimator to correct for drifts in the visual SLAM system. Here, we circumvent this problem by adopting *Stochastic Cloning* [12] which allows us to include relative measurements in a relative context only, which also means, that we no longer incorporate *local* estimates of the scale factor (typically effected by drift and jumps) to the *global* position estimation. This contrasts with our previous work where the latest scale estimate was applied to the global pose update, which means that a small local drift in scale would falsely result to large changes in global position estimates.

II. Combining Multiple Sensors

Our framework is based on the indirect formulation of an iterated EKF where the state prediction is driven by IMU measurements. The state consists of number of *core* states:

$$x_{core}^T = \left[{p_w^i}^T, {v_w^i}^T, {q_w^i}^T, b_\omega^T, b_a^T \right]. \quad (1)$$

Namely, these correspond to the relative position p_w^i, velocity v_w^i, and attitude[2] q_w^i of the IMU w.r.t. the world frame expressed in the world frame. Furthermore we estimate IMU acceleration and gyroscope biases b_a and b_ω, respectively. Additional sensors can then be added modularly with respect to the IMU frame.

We use the insight from the observability analysis carried out in [16] for the implementation of our framework in order to design a sensor suite for a particular platform and mission. For example, if we were to use a (differential) GPS receiver in addition to the IMU, we would need to add the translation between these two sensors as extrinsic calibration

[2]Relative rotation is parameterized as a Quaternion of rotation in Hamilton notation.

state. Similarly, a pressure sensor introduces a translational calibration state and an additional bias state in the global z axis. A more complex example is a monocular visual odometery module yielding a six-degrees of freedom (DoF) pose, measured w.r.t. a separate frame of reference which drifts in all six dimensions. As with the pressure sensor, we account for these drifts by adding a six-DoF state describing the drift between the world frame and the frame the sensor measurements are expressed in. Furthermore, we need to add a six-DoF extrinsic calibration state with respect to the IMU. A camera could as well measure optical flow representing a 3D body velocity sensor [17]. Since these are no global position measurements, we do not need to add drift states, but only a six-DoF extrinsic calibration state with respect to the IMU frame.

III. Example: Visual-Inertial-Pressure

As an example of a multi-sensor suite we derive the EKF update formulation of a common setup in MAV navigation consisting of an IMU, a pressure sensor and a monocular camera (whose feeds are processed in a visual SLAM providing relative 6-DoF pose estimates) – forming a loosly coupled visual-inertial-pressure navigation system. Starting from (1) we define the 15-element error state vector for core states as

$$\tilde{x}_{core}^T = \left[\Delta {p_w^i}^T, \Delta {v_w^i}^T, \delta {\Theta_w^i}^T, \Delta {b_w}^T, \Delta {b_a}^T\right], \quad (2)$$

with $\tilde{x}$ representing the difference of an estimate $\hat{x}$ to its true value x, which is defined as $\delta q = q \otimes \hat{q} \approx [1 \ \frac{1}{2}\delta\Theta^T]^T$ for quaternions.

Additional to these *core* states, every sensor adds a number of *auxiliary* states which relate the measured quantities to the *core* states. Therefore the full EKF state is assembled from the core states $\tilde{x}_{core}$ and a series of additional states $\tilde{x}_{s_i}$ which are defined by the sensor type:

$$\tilde{x}^T = \{\tilde{x}_{core}^T, \tilde{x}_{s_1}^T, \tilde{x}_{s_2}^T, ..., \tilde{x}_{s_n}^T\}. \quad (3)$$

Below, we derive the EKF equations for the visual-inertial-pressure suite.

1) Pressure sensor: In order to obtain height estimates from a pressure sensor we need to account for the bias b_{press} resulting from the changes in ambient pressure:

$$\tilde{x}_{press} = [\Delta b_{press}]. \quad (4)$$

For the pressure measurement z_{press}, the following measurement model applies:

$$z_{press} = p_{press} - b_{press} + n_{press} \quad (5)$$

with n_{press} denoting the measurement noise modeled as zero-mean, white and Gaussian and p_{press} denoting the measured altitude. We define the error in the z-position generally as $\tilde{z} = z - \hat{z}$. Which can be linearized to $\tilde{z} = H_{press}\tilde{x} + \eta$, where H_{press} denotes the Jacobian of the pressure measurement w.r.t. the (16-dimensional) error state.

2) Monocular visual SLAM sensor: The quantities to be estimated, are the scale λ and the drifts in position p_v^w and attitude q_v^w of the visual SLAM system w.r.t. the world-frame as well as the camera-to-IMU rotation q_i^c. We do not estimate the camera-to-IMU translational offset p_i^c online, as it is unlikely to change significantly during the mission. The states added by this sensor are:

$$\tilde{x}_{vis}^T = \left[\Delta\lambda, \delta\Theta_i^{cT}, \Delta {p_w^v}^T, \delta {\Theta_w^v}^T\right]. \quad (6)$$

For the camera pose measurement z_{vis}, the following measurement model applies [16]:

$$z_{vis} = \begin{bmatrix} p_v^c \\ q_v^c \end{bmatrix} = \begin{bmatrix} C_{(q_w^v)}(p_w^i + C_{(q_w^i)}^T p_i^c)\lambda + p_w^v + n_{p_v} \\ q_i^c \otimes q_w^i \otimes {q_w^v}^{-1} \otimes \delta q_{n_q v}, \end{bmatrix}; \quad (7)$$

with $C_{(q_w^i)}$ as the rotation matrix corresponding to the IMU's attitude and $C_{(q_w^v)}$, p_w^v the rotation and translation of the world frame to the vision frame expressed in the world frame, respectively. The visual measurement is corrupted by noise n_{p_v} and n_{q_v} which we model as zero-mean, white and Gaussian.

We define the position and attitude error of the vision measurement as

$$\begin{bmatrix} \tilde{z}_p \\ \tilde{z}_q \end{bmatrix} = \begin{bmatrix} C_{(q_v^w)}^T(p_w^i + C_{(q_w^i)}^T)\lambda + p_v^w - \\ (C_{(\hat{q}_v^w)}^T(p_w^i + C_{(\hat{q}_w^i)}^T)\hat{\lambda} + \hat{p}_v^w) \\ \otimes q_v^w \otimes (\hat{q}_i^c \otimes \hat{q}_w^i \otimes \hat{q}_v^w)^{-1} \end{bmatrix} \quad (8)$$

which can be linearized to $\tilde{z} = H_{vis}\tilde{x} + \eta$, where H_{vis} holds the Jacobian of the (visual-) pose measurement w.r.t. the error state.

IV. Generic and Modular Implementation

A. Processing delayed measurements

The proprioceptive sensor readings are used to predict the state at IMU rate in real time, which is crucial for for MAV attitude control. This estimate is updated with other sensor readings (e.g. from SLAM) which are available only at lower rates and arrive with significant (and potentially unknown) time delay. In our previous work [17], we proposed to use a ring buffer for the states so that we could apply state updates in the past. After applying an update to the corresponding state in the past we re-predict the state to the current time to keep the best state prediction available for control at high rate, see Fig. 2.

B. Process multiple delayed measurements

In order to integrate multiple sensor readings, we extend our previous approach in [17]: we maintain a potentially infinitely long buffer for states and measurements, inside which the elements are sorted by time. Whenever new IMU readings become available, a new state-object is instantiated, the state is predicted using the proprioceptive measurements

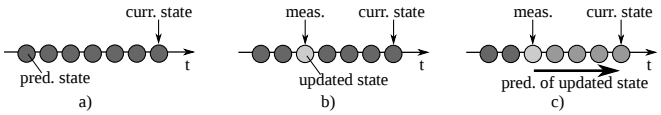

Fig. 2: The ring-buffer scheme we proposed in [17] to compensate for delayed measurements. a) The current state is used for control. The covariance is not required for control and therefore only predicted on demand. b) A delayed measurement arrives upon which the corresponding state is queried in the buffer. The covariance prediction is carried out to the state and both quantities are corrected by the measurement. c) The updated state is predicted to the current time to provide the most recent estimate for control. The covariance is predicted to the state where we anticipate the next measurement to arrive.

and then inserted into the state buffer, illustrated at the bottom row in Fig. 3.

As in Section III, we consider the typical MAV sensor-setup of an IMU, a pressure sensor and a monocular visual-SLAM system. Fusing the metric pressure sensor readings on IMU feeds ensures faster scale convergence and more accurate state estimates, as discussed in Section VI.

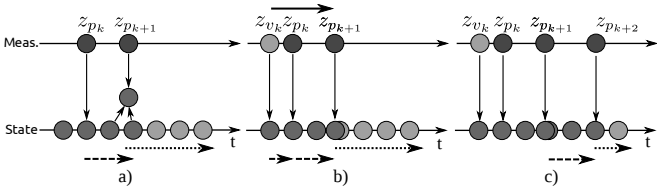

Fig. 3: a) Shows how pressure measurements (z_p) are applied to the closest state in the buffer. Interpolation allows us to use the best available linearization point. In b), delayed vision measurements (z_v) are applied by querying the closest state in the buffer, and any subsequent measurements are re-applied to the updated state. Finally, in c), sensors with different rates can be directly integrated to the framework. The covariance is only predicted on-demand (bold-dash arrows), while the state is always predicted to the current time for control after all pending measurements have been applied (point-dash arrows).

Commonly, the pressure sensor has (constant) measurement rates in the order of 50 Hz, while the visual-SLAM system usually operates at a (varying) rate between 30 and 90 Hz. In the upper row in Fig. 3 a) we illustrate how the pressure measurements arrive and are applied to the state closest to their respective measurement time. In cases where no state is available at the measurement time, we interpolate the respective proprioceptive measurements to get the best available linearization point (See Fig. 3 a.).

Fig. 3 b) illustrates the case where the delayed measurement of the visual-SLAM system becomes available. As before, the closest state (in the past) is retrieved from the state-buffer, which is updated using the information from the visual-SLAM system. Subsequently, the measurement buffer is queried recursively for later measurements, between which both the state and covariance are predicted accordingly.

After applying the last measurement in the measurement-buffer, all subsequent states in the state-buffer are predicted so that the most recent estimate is available for vehicle control (See Fig. 3 c).

C. Delayed state initialization

In multi sensor fusion, it is typical that some sensor feeds might be missed for some time or become unavailable during the mission. For example, indoors the MAV can navigate using the on-board visual and inertial sensors (global position and yaw unobservable) but with the transition to outdoors, GPS measurements become available (rendering global position and yaw observable). The framework allows sending (re-)initialization "measurements" at any time which get integrated to the estimation process seemlessly.

D. Relinearization of the prediction and IEKF window updates

Keeping past states and covariances in the buffer also allows us to employ an IEKF scheme over a window of measurements. When our a-priori state estimate is far from the a-posteriori state estimate, we hold the update back to first employ a set of IEKF iterations over a window of measurements and refine the linearization points before applying the update using the refined linearization point. Given the little computational cost for state and covariance prediction we can thus relinearize the prediction multiple times in order to reduce linearization errors for highly non-linear systems. This extends naturally to applying a set of updates as batch non-linear least squares optimization.

E. Outlier rejection

The framework allows the modular addition of outlier rejection methods to each measurement module. The filter core module then performs then e.g., a Mahalanobis test before applying the update to the state.

F. Compile time calculation of all indices and matrix dimensions

In our implementation we separate the state in core and auxiliary parts (which were added e.g., as bias and calibration terms for a particular sensor) already at the state level (see (3)). This allows us to perform optimizations by exploiting this knowledge in the prediction steps of the EKF. The code related to the core states can therefore be completely separated from the sensor specific implementations, rendering our *MSF-EKF* implementation very easy to extend. The definition of the current sensor suite is done at one place, and then used to unfold the full state and compute the dimensionality of the full state all at *compile time*. This allows transparent sensor integration and highly efficient matrix operations. The design of a new filter setup therefore consists only in the implementation of the measurement Jacobian and residual for the sensor. Additional states can be added with a single line from which the framework computes all indices and derives the necessary calculations at compile time. This state definition also specifies the local parameterizations for Quaternion error state Jacobian and update functions so that the filter can apply the correct parameterization automatically. Since our implementation includes a large set of sensor implementations, in most cases only the specification of the desired sensor setup is necessary.

Knowing the sensor suite and the respective state at compile time, we make extensive use of template meta-programming techniques to let the *compiler* compute all required matrix dimensions and all indices required in the EKF computation. This allows us to exploit the full efficiency of the linear algebra framework *Eigen*[3] we employ.

V. Processing relative measurements

Formulating the visual update as in Section III has drawbacks when using measurements from a visual odometery framework: The latest estimate of the visual scale is used to scale the whole visual odometery path, not taking into account intermediate changes in scale. In addition to that visual odometery systems (i.e. performing SLAM with a limited number of key-frames to bound computational complexity), provide a pose measurement which denotes a *relative* measurement between time-instants k and $k+m$ rendering the measured quantities dependent on the state values x_k and x_{k+m}, as well as the previous measurement. Nevertheless many recent publications ([14], [17], [15]) compute visual updates from a local map with fixed landmarks and apply them as absolute pose measurements leading to inconsistent estimates, prohibiting fusion with absolute measurements like GPS.

By the Markov assumption, in an EKF all information about past states is contained in the latest state estimate and both the state $\tilde{x}_{k|k}$ and the corresponding covariance $\tilde{P}_{k|k}$ are available. The standard EKF formulation, however, does not allow for direct consideration of the correlations between the states at different time steps and therefore, applying a relative measurement is not straightforward.

Similar to the *Stochastic Cloning* approach of [12], we adapt the EKF update equations to handle absolute measurements as in [17] and then to handle the relative measurements, relating two states. The measurement equation for the relative update is:

$$\tilde{z}_k = H_{k+m}\tilde{x}_{k+m|k} + H_k\tilde{x}_{k|k} + \eta, \tag{9}$$

where the subscript $k+m|k$ denotes the predicted quantities at time t_{k+m} and $k|k$ corresponds to the posterior at time-step t_k. Moreover, $\tilde{x}$ is state vector and H the corresponding measurement Jacobian.

A. Updating the state with a relative measurement

In order to account for relative measurements, the authors of [12] propose to add a clone of the state for each sensor providing relative measurements as well as the respective errors for the landmarks used to compute the relative measurements. Since we want to use the sensors in a loosely-coupled manner abstracting the internal algorithms (e.g. SLAM) we don't include the measurement errors for landmarks relating both states in our state vector. To keep the computational complexity low and because in general not all measurements in a multi-sensor setup denote relative quantities we do not in general clone every state but rather make use of our framework design, according to which, we can access the pair $\check{X}$ of past states we want to relate by a given relative measurement, at any time:

$$\check{X} = \begin{bmatrix} \tilde{x}_{k|k}^T & \tilde{x}_{k+m|k}^T \end{bmatrix}^T. \tag{10}$$

We then build the full covariance matrix of this state pair $\check{X}$:

$$\check{P}_{k+m|k} = \begin{bmatrix} P_{k|k} & P_{k|k}\mathcal{F}_{k+m,k}^T \\ \mathcal{F}_{k+m,k}P_{k|k} & P_{k+m|k} \end{bmatrix} \tag{11}$$

with $\mathcal{F}_{k+m,k} = \prod_{i=1}^{m} F_{k+i}$ corresponding to the concatenation of the linearized system dynamic matrix. We store the state transition matrix F_k (given the respective best available linearization point) in the buffer and only carry out the product accumulation and multiplication with $P_{k|k}$ when we want to apply a relative measurement. Thereby all additional measurements arriving within the time spanned by the relative measurements are considered and improve the respective linearization points for F_k. The residual r_{k+m} and the covariance of a relative measurement $\check{S}_{k+m}$ are given by:

$$\begin{aligned} r_{k+m} &= z_{k,k+m} - \hat{z}_{k,k+m} \simeq \check{H}\check{X}, \\ \check{S}_{k+m} &= \check{H}\check{P}_{k+m|k}\check{H}^T + R_r, \end{aligned} \tag{12}$$

where R_r is the covariance of the relative pose coming from the employed SLAM framework and $\check{H} = \begin{bmatrix} H_{k|k} & H_{k+m|k} \end{bmatrix}$ comprises the two corresponding measurement Jacobians. The Kalman gain calculation then is straightforward:

$$\check{K} = \check{P}_{k+m|k}\check{H}^T\check{S}_{k+m}^{-1} = \begin{bmatrix} K_k^T & K_{k+m}^T \end{bmatrix}^T. \tag{13}$$

The final step is the correction of the state and the covariance at time step t_{k+m} given the residual r_{k+m}:

$$\begin{aligned} \hat{x}_{k+m|k+m} &= \hat{x}_{k+m|k} + K_{k+m}r_{k+m}, \\ P_{k+m|k+m} &= P_{k+m|k} - K_{k+m}\check{S}_{k+m}K_{k+m}^T. \end{aligned} \tag{14}$$

Given the better estimate of $\hat{x}_{k+m|k+m}$ we can now re-apply all measurements that arrived after the relative measurements and re-predict the state using the new linearization points. Since we perform relinearization of the prediction, multiple relative measurements are always applied using the best available linearization point.

The main question left is the derivation of the measurement covariance for the relative update.

B. Pose estimation covariance in key-frame based visual SLAM systems

Instead of deriving the uncertainty in the camera pose from the prediction of the pose and the visible landmarks as in EKF SLAM, the covariance of the camera-pose is obtained from *bundle adjustment*. This non-linear optimization technique solves simultaneously for both 3D-map points in the world and the camera locations over time. This is done by minimizing the weighted-least-squares re-projection

[3] http://eigen.tuxfamily.org

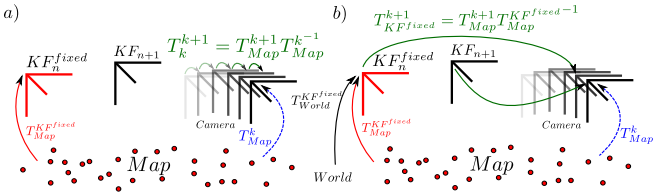

Fig. 4: Part a) on the left shows a straight forward implementation of stochastic cloning using the relative transformations between tracker poses as relative measurements. Pose drift is inevitable because noisy estimates get integrated. This degrades the main benefit of key-frame based SLAM: The absence of temporal drift. Moreover the covariance of this measurement is not available, prohibiting it's fusion with other sensors. b) The relative measurement is expressed w.r.t. a fixed key frame of the map at any time, making it possible to derive a transformation for which we can also get a correct covariance estimate. This estimate is drift free and incorporates the relative covariance correctly.

errors of the 3D map points across all images, which then provides an estimate of the pose covariance [4]. Since the solution of the bundle adjustment problem is costly, the real-time pose estimate (pose tracking) is commonly calculated from an approximation where the map-points are kept fix and the current camera pose is recovered solving the perspective n-point (PnP) [2] problem. The loosely coupled SLAM systems employed in related work [14], [17] therefore provide a covariance of the current pose T^k_{Map} w.r.t. *the local fixed map* which is not the quantity we would need to apply relative measurements. If one would apply these local transformations T^{k+1}_k as relative updates, one would lose of the main benefits of key-frame based SLAM system, namely the absence of temporal drift as shown in Fig. 4 a).

C. Stochastic cloning for key-frame based visual SLAM systems

The covariance calculated w.r.t. the fixed map actually denotes the uncertainty of the current pose T^k_{Map} w.r.t. the fixed key-frame KF^{fixed}_n in the bundle adjustment problem (fixed as to fix gauge freedom) as shown in Fig. 4 b). Therefore to correctly integrate the *relative*-measurements, we need to apply stochastic-cloning to the EKF-state corresponding to the time the *current fixed* key-frame *became fixed*. Since for long term missions we have to drop *old* key frames to keep computational demands low, which key-frame is fixed in the bundle-adjustment problem is changing over time. This means that the current pose needs to be computed relative to a changing reference in the map as shown in Fig. 4 b).

The uncertainty of the fixed key-frame w.r.t. the world frame is changing whenever the fixed key-frame changes. At this moment the uncertainty of the past fixed key-frame w.r.t. the world frame is augmented with the uncertain transformation of the new fixed key-frame which is obtained from bundle-adjustment.

The pose measurement is therefore computed via the uncertain transformation chain from the world frame to the fixed key-frame $T^{KF^{fixed}}_{World}$ and from there to the current camera pose estimate T^k_{Map}. This allows us to take advantage of the non temporal drifting estimate of key-frame based SLAM but at the same time correctly account for the uncertainties in the pose estimate which then allows fusion with additional exteroceptive sensors.

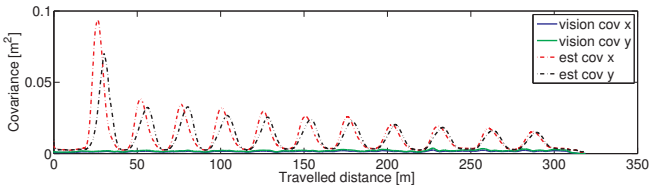

Fig. 5: When applying the covariance of the visual pose as an absolute measurement the covariance of the global x and y positions is decreasing with time despite no global information being available. This prohibits optimal fusion with absolute measurements as provided by e.g., GPS.

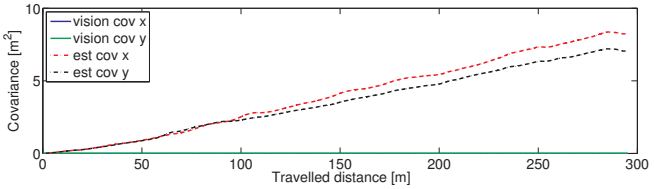

Fig. 6: When applying the covariance of the visual pose as a relative measurement the covariance of the global x and y position is growing as expected.

VI. Experiments

In this section we present a series of experiments which we carried out by flying ~350 m with an Asctec Firefly MAV equipped with a Intel Core2Duo computer [4]. All computation is done on-board and the flights are purely vision based on grass with speeds of up to 4 m/s. Ground truth for all experiments is provided by a sub-mm precision *Leica TotalStation 15i* which continuously tracks the MAV.

A. Covariance of pose estimate for sensor fusion

As discussed earlier, most visual SLAM systems based on key-frames (e.g. PTAM [7]) provide a pose estimate and respective covariance w.r.t. a local map following bundle adjustment. The covariance of such visual pose estimates is most often over-confident since the uncertainty in the pose is estimated assuming a fixed map for computational efficiency (in contrast to EKF SLAM where a full covariance matrix of the pose and the map is jointly estimated). When fusing this visual estimate as an absolute measurement the covariance of the global position does not increase with the distance traveled as shown in Fig. 5 (The oscillations show how applying local PnP based estimates as *global* position measurements multiplied with the current scale estimate leads to wrong covariance estimates). If we would like to fuse this over-confident visual pose estimate with a relatively uncertain GPS pose (roughly spanning an area of $1m^2$), the corrections from GPS would only have minor influence on the state estimate, and as a result, global position drifts of visual SLAM cannot be corrected for.

If the visual SLAM measurements are applied as relative measurements, however, the global pose covariance (given only visual updates) grows over time (Fig. 6), as expected to reflect the true uncertainty of the global position. This allows consistent fusion with global measurements like GPS.

[4]The actual experiment was 800 m but ground truth is only available for a sub-part.

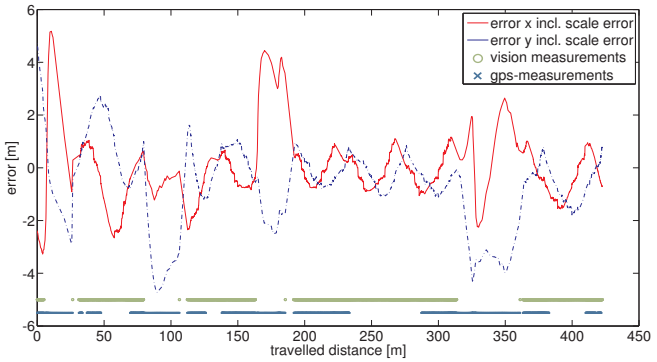

Fig. 7: This plot shows the absolute errors in x and y axes over traveled distance of the proposed framework when switching vision and GPS sensors on and off several times. During the GPS covered parts of the mission, the vehicle passed a cluster of trees several times, which introduced large drift in GPS position. The plot shows that the framework is able to correct for this drift once the measurements from an additional sensor become available again.

B. Online sensor switching

While the improvement of the estimated state is one benefit of including additional sensors, the capability of seamless switching between the elements of the sensor-suite *online* adds both additional fail-safety and versatility. One common case is the change from local vision-based navigation (indoor) to GPS-based global position estimates (outdoor). Current experimental results for MAV navigation are primarily bound by battery life and therefore rather small in scale of operation (usually trajectories of around 1 km). While drifts in position and visual-scale become important at this trajectory length, the local estimate provided by vision systems is still superior to GPS. To demonstrate the capability of online-sensor switching, we added alternating drop-outs of both GPS and vision measurements. Some dropouts of the visual update take place when the MAV was passing a group of trees, where the GPS is highly corrupted by multi-pass and high dilution of precision. In this area, the state estimate then follows the corrupted GPS measurements adding errors as large as 5 m (Fig. 7).

C. Processing time

While the main motivation for the heavy use of template meta-programming was to keep the framework generic and transparent to the employed sensor-suite, there are also significant improvements in terms of computational efficiency[5] as detailed in the table below. The state corresponds to a visual-inertial fusion filter estimating both rotation and translation of the extrinsic IMU to camera calibration.

Cost of function call (EKF: 31 states)	mean	std dev
IMU handling and state prediction	44 μs	23 μs
Covariance prediction	31 μs	24 μs
Get state for delayed measurement	79 μs	53 μs
Apply measurement	65 μs	37 μs
Re-predict state after measurement	21 μs	22 μs
Additional fixed overhead	11 μs	9 μs

[5]clang 3.2-9 -O3 -march=native, i7-2720QM, 16GB M471B5273DH0-CH9

VII. CONCLUSIONS

In this paper, we present our *MSF-EKF* framework for multi-sensor fusion able to handle delayed absolute and relative measurements seamlessly. We discuss how a sensor-suite can be designed according to the requirements of the mission and which combinations of sensors render particular parts of the state observable. The results from this discussion lead to the derivation of our implementation, where we highlight our generic and modular multi-sensor fusion framework. We then show how our framework can be employed to add robustness and fail-safety to long term missions, where not all sensors might be available at any time. In the near future, we plan to open-source our *MSF-EKF* framework for other researchers to employ it on their platforms, while future research will focus on the implementation and evaluation of the key-frame based stochastic cloning for multiple relative sensors.

REFERENCES

[1] G. Chowdhary, E. Johnson, D. Magree, D. Wu, and A. Shein. GPS-Denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft. Journal of Field Robotics (JFR), 2013.

[2] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[3] S. Grzonka, G. Grisetti, and W. Burgard. Towards a navigation system for autonomous indoor flying. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Kobe, Japan, 2009. IEEE.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[5] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Towards consistent vision-aided inertial navigation. In *Algorithmic Foundations of Robotics X*. Springer, 2013.

[6] V. Indelman, S. Williams, M. Kaess, and F. Dellaert. Factor graph based incremental smoothing in inertial navigation systems. In *Information Fusion (FUSION), 2012 15th International Conference on*. IEEE, 2012.

[7] G. Klein. *Visual Tracking for Augmented Reality*. PhD thesis, University of Cambridge, 2006.

[8] S. Leutenegger and R. Y. Siegwart. A low-cost and fail-safe inertial navigation system for airplanes. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2012. IEEE.

[9] M. Li and A. I. Mourikis. High-precision, consistent EKF-based visual–inertial odometry. *The International Journal of Robotics Research, (IJRR)*, 2013.

[10] F. M. Mirzaei and S. I. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *Robotics, IEEE Transactions on*, 24(5):1143–1156, 2008.

[11] A. Mourikis and S. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2007.

[12] A. I. Mourikis, S. I. Roumeliotis, and J. W. Burdick. SC-KF mobile robot localization: a stochastic cloning Kalman filter for processing relative-state measurements. *Robotics, IEEE Transactions on*, 2007.

[13] S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained mav. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2011.

[14] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar. Vision-based state estimation and trajectory control towards aggressive flight with a quadrotor. In *Robotics Science and Systems*, 2013.

[15] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar. Vision-based state estimation for autonomous rotorcraft MAVs in complex environments. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[16] S. Weiss. *Vision based navigation for micro helicopters*. PhD thesis, ETH Zurich, 2012.

[17] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.