Collaborative stereo

Conference Paper *in* Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems · September 2011

DOI: 10.1109/IROS.2011.6048550 · Source: DBLP

5 authors, including:

Markus Wilhelm Achtelik
ETH Zurich
30 PUBLICATIONS **2,315** CITATIONS
SEE PROFILE

Stephan Weiss
Institute of Electrical and Electronics Engineers
42 PUBLICATIONS **2,708** CITATIONS
SEE PROFILE

Margarita Chli
ETH Zurich
31 PUBLICATIONS **3,200** CITATIONS
SEE PROFILE

Frank Dellaert
Georgia Institute of Technology
243 PUBLICATIONS **14,860** CITATIONS
SEE PROFILE

Collaborative Stereo

Markus W. Achtelik*, Stephan Weiss*, Margarita Chli*, Frank Dellaert†, Roland Siegwart*

* ETH Zürich, † Georgia Institude of Technology

{markus.achtelik, stephan.weiss, margarita.chli}@mavt.ethz.ch, dellaert@cc.gatech.edu, rsiegwart@ethz.ch

***Abstract*— In this paper, we propose a method to recover the relative pose of two robots in absolute scale and in real-time using one monocular camera on each robot. We achieve this by fusing measurements from the onboard inertial sensors on each platform with information obtained from feature correspondences between the two cameras using an Extended Kalman Filter (EKF). This forms a flexible stereo rig, providing the ability to treat the two robots as one single dynamic sensor, which can adapt to the environment and thus improve environmental mapping, obstacle avoidance and navigation. We demonstrate the power of this approach on both simulation and real datasets, employing two micro aerial vehicles (MAVs) to illustrate successful operation over general 3D motion.**

I. Introduction

Localization of a robot and estimation of its environment based on visual information are well-explored tasks in the fields of Robotics and Computer Vision. Mounting a single camera or a camera rig on a robot, methods like SFM and SLAM can be employed to estimate the robot's pose and build a map of its workspace which are key to autonomous navigation. Taking this a step further, here we address the problem of collaborative sensing of the environment via autonomous, independently-moving robotic platforms. The core requirement for collaborative building of a unified map is the localization of the robots with respect to each other. While this is often assumed to be given by an external reference (e.g. GPS, VICON system), this paper presents a method for automatic real-time relative position estimation of two robotic platforms undergoing general motion in 3D. Estimation is based primarily on visual data, forming a variable-baseline stereo setup towards efficient, high-quality scene estimation.

A. Vision as a Primary Sensor

The generality and compactness of using a single camera to estimate the scene structure has had major influence on the popularity of this setup [1], [2]. However, the inherent difficulty in estimating depth and absolute scale in monocular solutions imposes great research challenges.

For depth to be estimated from a monocular image, the scene has to be viewed from a sufficiently large baseline, constraining the motion of the camera-carrying platform: not only is the robot required to move before scene estimation is possible, but degenerate cases like motion along the optical axis should be avoided as they slow down the convergence of uncertainty in the map. The feature parametrization proposed in [3] permits well-behaved initialization, however motion parallax is still necessary for obtaining good 3D estimates.

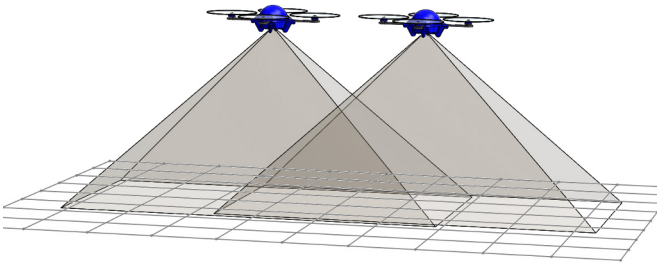

Fig. 1. Using one monocular camera on each MAV, the commonly viewed scenery is used together with inertial data to estimate the relative pose of each MAV. Forming a flexible stereo rig, this introduces the idea of collaborative stereo towards autonomous cooperative navigation.

The unobservability of absolute scale affects all monocular setups – from SLAM systems [2] to pose estimation in n-point algorithms [4]. While relative scale estimation is possible, it is never clear whether an obtained map corresponds to a lilliputian or a gigantic structure. In [2] this was addressed by observing an object of known dimensions at start-up, initializing scale with a known value automatically. In [5] this was tackled by fusing inertial measurements with monocular data, allowing estimation of absolute scale by including it in the EKF state vector.

In stereo setups, scene-depth and absolute scale estimation do not pose such a big challenge as the known baseline usually permits recovery of both. In just one stereo-snapshot of the scene, information about obstacles in close range can immediately become available. However, observing objects at a much greater distance in comparison to the cameras' baseline, reduces a stereo setup to almost a bearing-only sensor suffering similar depth estimation problems as the monocular case. Acknowledging this problem, [6] selected poses along the camera's trajectory in a SFM scenario to optimize the baseline of triangulation and the resolution, for different scene depths. Advocating the power of variable baseline they illustrated significant gains in accuracy over standard, fixed-baseline stereo methods. In the same spirit of exploiting the advantages of variable baseline, here we aim to demonstrate a method for estimation of dynamic baseline to be used *online*.

B. From Individual Operation to Collaborative Sensing

Collaborative sensing and multi-robot scenarios have long been studied in the Robotics community. Interestingly, [7] studied strategic trajectory control in a cooperative SLAM scenario aiming at maximizing information with respect to

the SLAM estimates. Actively guiding the trajectories and motions of vehicles they explored the relative merits of centralized and decentralized data fusion in simulation.

Fusing information from multiple cameras, [8] used a SLAM filter to perform self-calibration, cooperative localization and mapping. Specifying the relative configuration at startup, they demonstrated results on two cameras (using initial baseline to set absolute scale) and presented a quantitative analysis on results obtained from a fixed-baseline stereo rig undergoing motion on a straight line. In this work, we choose to demonstrate our methodology on MAVs tackling the general case of motion in 3D . Fusing visual and inertial data, we present results compared against ground truth obtained from a VICON tracking system.

C. The Aim of this Work

We propose a method to recover the relative configuration of two robots navigating autonomously, in absolute scale and in real-time without prior knowledge on their initial configuration or workspace. Combining the advantages of stereo and monocular vision, we form a flexible stereo rig by equipping each robot with a monocular camera and an inertial measurement unit (IMU). Using feature correspondences in the overlapping field of view, we estimate the relative 6 DoF transformation between the robots' poses up to a scalar factor in translation. We recover the real translation by fusing it with IMU measurements (linear acceleration, agular velocity) inside an EKF formulation. Intuitively – given that there is motion between the robots – changes in the relative transformation ("relative acceleration") will deviate from IMU measurements. This error is fed back to the state and thus allows to find the correct scaling.

II. Problem Setup

A. Notation

Since we are working with multiple vehicles in different coordinate frames and we integrate and differentiate in different coordinate frames, we use the following notation:

$$ {}^{\text{(coordinate frame)}}_{\text{(reference frame)}}\boldsymbol{x}_{\text{(vehicle number)}} \tag{1} $$

For example, the acceleration $\boldsymbol{a}$ of vehicle 1 expressed in the vehicle's coordinate frame, but measured with respect to the inertial (world) frame w, is denoted as ${}^{1}_{w}\boldsymbol{a}_1$. The reference frame is only important for differentiation of position (leading to linear velocity and acceleration) and is therefore omitted for the remaining variables.

To represent rotations, we use either quaternions or rotation matrices depending on their suitability to the problem at hand. For quaternions $\bar{q}$, we use the Hamilton notation as in [9]. Written as a vector $\bar{q} = \begin{bmatrix} q & \boldsymbol{q}^T \end{bmatrix}^T$, the first element is the real part and the remaining elements are the imaginary part. Small bold letters denote a vector and the $\bar{\cdot}$ notation, the (full imaginary) quaternion version, which is needed for rotations with quaternions. $\lfloor \boldsymbol{x} \times \rfloor$ denotes a skew-symmetric matrix of $\boldsymbol{x}$ to compute a cross product with another vector. Further conventions can be found in the appendix.

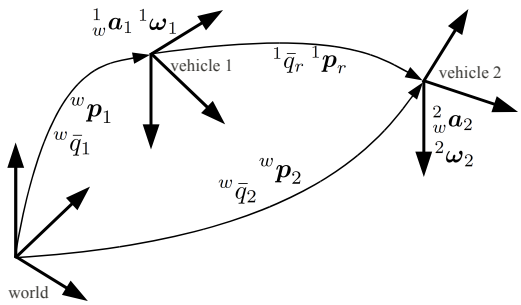

Fig. 2. Setup of the system with its coordinate frames and variables

B. Assumptions

As we aim for relative pose estimation for general 3D trajectories, we explicitly avoid constraints on the robots' motion and derive our method for the full 6 DoF case. However, we still have to make a few realistic assumptions for simplicity which otherwise do not affect the method.

- Acceleration and gyro biases of the IMU readings are known. These can be estimated at startup or at runtime.
- The calibration from the camera to the IMU is known. It can be determined by the method in [10]. Furthermore, we assume that the translation from the camera to the IMU is small to avoid introducing additional dynamics.
- The cameras have an overlapping field of view, as is the case in standard stereo scenarios.

In this paper, we focus on data fusion of visual and inertial information. We therefore treat the vision part as black box providing the rotation and translation between two cameras with overlapping field of view up to a translatory scale. This could be *e.g.* the 5-point algorithm [4] or 4-point homography [11] computing the pose between the camera from feature correspondences in two views. We furthermore assume that there exists scale propagation between two consecutive time steps. This scale propagation may drift, thus we model our filter accordingly.

C. Measurements and Variables to be Estimated

The sensor suite comprises of a single camera and an IMU on each robot. The IMU is equipped with a 3-axis gyroscope to measure angular velocity and a 3-axis linear acceleration sensor. Fig. 2 shows the setup of our system with its variables. We express the variables to be estimated with respect to the reference frame of vehicle 1. In particular, we want to estimate the relative position $\boldsymbol{p}_r$ in absolute scale and the relative orientation $\bar{q}_r$. We achieve this by evaluating:

- ${}^{i}_{w}\boldsymbol{a}_i, i = 1, 2$: Acceleration of the vehicles with respect to the global (inertial) frame, expressed in local coordinates. This measurement also includes the gravity vector. (IMU)
- ${}^{i}\boldsymbol{\omega}_i, i = 1, 2$: Angular velocity of the vehicles expressed in their local coordinate system. (IMU)
- ${}^{1}\boldsymbol{p}_r \cdot \lambda$: relative position up to a scale λ obtained from feature correspondences between the vehicles' cameras. (Vision)
- ${}^{1}\bar{q}_r$: relative orientation from feature correspondences between the vehicles' cameras. (Vision)
- ${}^{w}\bar{q}_i, {}^{w}\boldsymbol{p}_i$: attitude and position of vehicle i w.r.t. world.

III. Relative Pose Estimation

We use an EKF to estimate the variables listed in the previous section from visual measurements and from measurements of the IMUs. Inspired by the work in [5], the main idea is that the acceleration sensors provide metric information, however on the acceleration level. The vision algorithm provides drift free position information between the vehicles, but this measurement is scaled by a factor λ which we refer to as "scale". We avoid integration of acceleration and/or differentiation of the pose to obtain λ by adding it to the state. The scaled vision measurement for position produces a residual during the measurement update and the state including the scale is corrected accordingly.

The state consists of the orientation ${}^1\bar{q}_r$ and metric position ${}^1\boldsymbol{p}_r$ of vehicle 2 w.r.t. vehicle 1. Furthermore, we need an intermediate state variable ${}^1_w\boldsymbol{v}_r$ (relative velocity w.r.t. world, expressed in the coord. frame of vehicle 1) to integrate the accelerations in order to link ${}^1\boldsymbol{p}_r$ with IMU readings.

We also include the angular velocities ${}^1\boldsymbol{\omega}_1$, ${}^2\boldsymbol{\omega}_2$ as well as the linear accelerations ${}^1_w\boldsymbol{a}_1$, ${}^2_w\boldsymbol{a}_2$ into the state vector which allows us to handle measurements from both vehicles arriving out of sync. This yields the following 23-long state:

$$\boldsymbol{x} = \left[{}^1\bar{q}_r{}^T \; {}^1\boldsymbol{\omega}_1{}^T \; {}^2\boldsymbol{\omega}_2{}^T \; {}^1\boldsymbol{p}_r{}^T \; {}^1_w\boldsymbol{v}_r{}^T \; {}^1_w\boldsymbol{a}_1{}^T \; {}^2_w\boldsymbol{a}_2{}^T \; \lambda \right]^T \quad (2)$$

A. State Equations

In order to properly define the differential equations governing the state, we need to study the relations between the moving coordinate frames of the vehicles. The relative orientation ${}^1\bar{q}_r$ of vehicle 2 w.r.t vehicle 1, is defined as:

$${}^w\bar{q}_2 = {}^w\bar{q}_1 \otimes {}^1\bar{q}_r \Leftrightarrow {}^1\bar{q}_r = {}^w\bar{q}_1^* \otimes {}^w\bar{q}_2 \quad (3)$$

From the quaternion derivative (48) and the derivative of its conjugate (49), we get a differential equation for ${}^1\bar{q}_r$:

$$\begin{aligned} {}^1_1\dot{\bar{q}}_r =& \frac{d}{dt}({}^w\bar{q}_1^* \otimes {}^w\bar{q}_2) \\ =& \frac{d}{dt}{}^w\bar{q}_1^* \otimes {}^w\bar{q}_2 + {}^w\bar{q}_1^* \otimes \frac{d}{dt}{}^w\bar{q}_2 \\ =& 0.5 \cdot ({}^1\bar{q}_r \otimes {}^2\bar{\omega}_2 - {}^1\bar{\omega}_1 \otimes {}^1\bar{q}_r) \end{aligned} \quad (4)$$

When differentiating position and velocity in rotating coordinate frames, both rotation and position/velocity depend on time. Therefore, product rules have to be applied. In the following, we use rotation matrices in order to reduce the complexity of the equations:

$$\bar{q}_i \otimes \bar{p} \otimes \bar{q}_i^* \widehat{=} \boldsymbol{R}(\bar{q}_i) \cdot \boldsymbol{p} = \boldsymbol{R}_i \cdot \boldsymbol{p} \quad (5)$$

To obtain a relation for ${}^1\boldsymbol{p}_r$ and to link it to the inertial frame, we look at the sum of translations:

$$\begin{aligned} {}^w\boldsymbol{p}_1 + \boldsymbol{R}_1 \cdot {}^1\boldsymbol{p}_r - {}^w\boldsymbol{p}_2 = 0 \Leftrightarrow {}^w\boldsymbol{p}_2 - {}^w\boldsymbol{p}_1 = \boldsymbol{R}_1 \cdot {}^1\boldsymbol{p}_r \\ {}^w\boldsymbol{p}_r = \boldsymbol{R}_1 \cdot {}^1\boldsymbol{p}_r \end{aligned} \quad (6)$$

Differentiation of (6) and solving for ${}^1_1\dot{\boldsymbol{p}}_r$ yields the desired differential equation for the relative position.

$$\begin{aligned} {}^w_w\dot{\boldsymbol{p}}_r &= \boldsymbol{R}_1 \cdot \lfloor {}^1\boldsymbol{\omega}_1 \times \rfloor \cdot {}^1\boldsymbol{p}_r + \boldsymbol{R}_1 \cdot {}^1_1\dot{\boldsymbol{p}}_r \\ \Leftrightarrow {}^1_1\dot{\boldsymbol{p}}_r &= \boldsymbol{R}_1^T \cdot {}^w_w\dot{\boldsymbol{p}}_r - \lfloor {}^1\boldsymbol{\omega}_1 \times \rfloor \cdot {}^1\boldsymbol{p}_r \\ {}^1_1\dot{\boldsymbol{p}}_r &= {}^1_w\boldsymbol{v}_r - \lfloor {}^1\boldsymbol{\omega}_1 \times \rfloor \cdot {}^1\boldsymbol{p}_r \end{aligned} \quad (7)$$

We obtain an expression for ${}^1_w\dot{\boldsymbol{v}}_r$ in a similar way:

$$\begin{aligned} {}^w_w\boldsymbol{v}_r &= \boldsymbol{R}_1 \cdot {}^1_w\boldsymbol{v}_r \\ {}^w_w\dot{\boldsymbol{v}}_r &= \boldsymbol{R}_1 \cdot \lfloor {}^1\boldsymbol{\omega}_1 \times \rfloor \cdot {}^1_w\boldsymbol{v}_r + \boldsymbol{R}_1 \cdot {}^1_w\dot{\boldsymbol{v}}_r \\ \Leftrightarrow {}^1_w\dot{\boldsymbol{v}}_r &= \boldsymbol{R}_1^T \cdot {}^w_w\dot{\boldsymbol{v}}_r - \lfloor {}^1\boldsymbol{\omega}_1 \times \rfloor \cdot {}^1_w\boldsymbol{v}_r \end{aligned} \quad (8)$$

We replace ${}^w_w\dot{\boldsymbol{v}}_r$ by the difference of accelerations w.r.t. the inertial frame. A multiplication with $\boldsymbol{R}_i^T$ expresses ${}^w_w\boldsymbol{a}_i$ in the local coordinate system of the respective vehicle, and this is what can be measured by the IMUs. We see that these rotations nicely cancel out and ${}^1_w\dot{\boldsymbol{v}}_r$ solely depends on the state variables we defined:

$$\begin{aligned} {}^1_w\dot{\boldsymbol{v}}_r &= \boldsymbol{R}_1^T \cdot ({}^w_w\boldsymbol{a}_2 - {}^w_w\boldsymbol{a}_1) - \lfloor {}^1\boldsymbol{\omega}_1 \times \rfloor \cdot {}^1_w\boldsymbol{v}_r \\ {}^1_w\dot{\boldsymbol{v}}_r &= \boldsymbol{R}_1^T \cdot {}^w_w\boldsymbol{a}_2 - {}^1_w\boldsymbol{a}_1 - \lfloor {}^1\boldsymbol{\omega}_1 \times \rfloor \cdot {}^1_w\boldsymbol{v}_r \\ {}^1_w\dot{\boldsymbol{v}}_r &= \boldsymbol{R}_1^T \cdot \boldsymbol{R}_2 \cdot {}^2_w\boldsymbol{a}_2 - {}^1_w\boldsymbol{a}_1 - \lfloor {}^1\boldsymbol{\omega}_1 \times \rfloor \cdot {}^1_w\boldsymbol{v}_r \\ {}^1_w\dot{\boldsymbol{v}}_r &= \boldsymbol{R}_r \cdot {}^2_w\boldsymbol{a}_2 - {}^1_w\boldsymbol{a}_1 - \lfloor {}^1\boldsymbol{\omega}_1 \times \rfloor \cdot {}^1_w\boldsymbol{v}_r \end{aligned} \quad (9)$$

These relations allow us to link acceleration and turn rate measurements with ${}^1\boldsymbol{p}_r$. In contrast, naively computing an explicit "relative acceleration" between the vehicles would require to transform ${}^2_w\boldsymbol{a}_2$ into the rotating reference frame of vehicle 1. This involves an additional term to compensate for the Euler effect which requires angular acceleration $\dot{\boldsymbol{\omega}}_1$. This cannot be measured with standard IMUs and estimating it would be prone to noise. $\boldsymbol{R}_r$ transforms ${}^2_w\boldsymbol{a}_2$ to vehicle 1's coordinate frame. Therefore the gravity vector, which is also measured by the acceleration sensors, cancels out. This avoids estimation of the global orientation of each vehicle in order to remove the gravity vector from the measurements.

The angular velocities ${}^1\boldsymbol{\omega}_1$, ${}^2\boldsymbol{\omega}_2$ and linear accelerations ${}^1_w\boldsymbol{a}_1$, ${}^2_w\boldsymbol{a}_2$ are modeled as random walk. Since the scale is likely to drift slowly, it is modeled as random walk as well. $\boldsymbol{n}$ is assumed to be white Gaussian noise with zero mean and covariance σ^2.

From this point onwards, we drop the super/subscripts denoting the coordinate/reference frames for clarity. Hence, the state equations are:

$$\begin{aligned} \dot{\bar{q}} &= 0.5 \cdot (\bar{q} \otimes \bar{\omega}_2 - \bar{\omega}_1 \otimes \bar{q}) \\ \dot{\boldsymbol{\omega}}_1 &= \boldsymbol{n}_{\omega_1} \\ \dot{\boldsymbol{\omega}}_2 &= \boldsymbol{n}_{\omega_2} \\ \dot{\boldsymbol{p}} &= \boldsymbol{v} - \lfloor \boldsymbol{\omega}_1 \times \rfloor \cdot \boldsymbol{p} \\ \dot{\boldsymbol{v}} &= \boldsymbol{R}(\bar{q}) \cdot \boldsymbol{a}_2 - \boldsymbol{a}_1 - \lfloor \boldsymbol{\omega}_1 \times \rfloor \cdot \boldsymbol{v} \\ \dot{\boldsymbol{a}}_1 &= \boldsymbol{n}_{a_1} \\ \dot{\boldsymbol{a}}_2 &= \boldsymbol{n}_{a_2} \\ \dot{\lambda} &= n_\lambda \end{aligned} \quad (10)$$

B. Error State Representation

In the following, $\hat{}$ denotes the estimated state and a preceding Δ or δ, the error state for additive error or multiplicative error, respectively. We define the error state vector $\tilde{\boldsymbol{x}}$ as:

$$\tilde{\boldsymbol{x}} = \left[\delta\boldsymbol{\theta}^T \; \Delta\boldsymbol{\omega}_1^T \; \Delta\boldsymbol{\omega}_2^T \; \Delta\boldsymbol{p}^T \; \Delta\boldsymbol{v}^T \; \Delta\boldsymbol{a}_1^T \; \Delta\boldsymbol{a}_1^T \; \Delta\lambda \right]^T \quad (11)$$

Since the quaternion for the relative attitude is enforced to have unit length, the corresponding covariance matrix would

be singular. Furthermore, instead of the difference of two quaternions, we take a small quaternion rotation as the error.

$$\begin{aligned}\bar{q} &= \hat{\bar{q}} \otimes \delta\bar{q} \\ \dot{\bar{q}} &= \dot{\hat{\bar{q}}} \otimes \delta\bar{q} + \hat{\bar{q}} \otimes \dot{\delta\bar{q}} \Leftrightarrow \\ \dot{\delta\bar{q}} &= \hat{\bar{q}}^* \otimes (\dot{\bar{q}} - \dot{\hat{\bar{q}}} \otimes \delta\bar{q})\end{aligned} \tag{12}$$

With the derivative of the relative attitude (4), and assuming an additive error $\bar{\omega}_i = \hat{\bar{\omega}}_i + \Delta\bar{\omega}_i$ for the angular velocities, the derivative of the error quaternion is:

$$\begin{aligned}\dot{\delta\bar{q}} =& \frac{1}{2}(\delta\bar{q} \otimes \hat{\bar{\omega}}_2 - \hat{\bar{\omega}}_2 \otimes \delta\bar{q} + \delta\bar{q} \otimes \Delta\bar{\omega}_2 \\ & - \hat{\bar{q}}^* \otimes \Delta\bar{\omega}_1 \otimes \hat{\bar{q}} \otimes \delta\bar{q})\end{aligned} \tag{13}$$

The rotation of $\Delta\bar{\omega}_1$ by $\hat{\bar{q}}^*$, can also be expressed by a rotation matrix $\widehat{\boldsymbol{R}}^T$ according to (5). Rewriting the quaternion products as matrix-vector multiplications (46), reordering and neglecting higher order terms involving multiplications of the type $\Delta\boldsymbol{\omega}_i \cdot \delta\boldsymbol{q}$, we get:

$$\dot{\delta\bar{q}}_r = \begin{bmatrix} 0 \\ -\lfloor\hat{\boldsymbol{\omega}}_2\times\rfloor\delta\boldsymbol{q} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2}\Delta\boldsymbol{\omega}_2 \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{1}{2}\widehat{\boldsymbol{R}}^T\Delta\boldsymbol{\omega}_1 \end{bmatrix} \tag{14}$$

Applying the small angle approximation for quaternions (47), we finally get the error state for the rotation between both vehicles:

$$\dot{\delta\boldsymbol{\theta}} = -\lfloor\hat{\boldsymbol{\omega}}_2\times\rfloor\delta\boldsymbol{\theta} - \widehat{\boldsymbol{R}}^T\Delta\boldsymbol{\omega}_1 + \Delta\boldsymbol{\omega}_2 \tag{15}$$

For the relative position, we use an additive error model $\dot{\boldsymbol{p}} = \dot{\hat{\boldsymbol{p}}} + \dot{\Delta\boldsymbol{p}}$. With this model and the state equation for position (7), we obtain the error state for the position. Again, higher order terms of the form $\Delta \cdot \Delta$ are neglected:

$$\begin{aligned}\dot{\Delta\boldsymbol{p}} &= \boldsymbol{v} - \lfloor\boldsymbol{\omega}_1\times\rfloor\boldsymbol{p} - \hat{\boldsymbol{v}} + \lfloor\hat{\boldsymbol{\omega}}_1\times\rfloor\hat{\boldsymbol{p}} \quad \Leftrightarrow \\ \dot{\Delta\boldsymbol{p}} &= \lfloor\hat{\boldsymbol{p}}\times\rfloor\Delta\boldsymbol{\omega}_1 - \lfloor\hat{\boldsymbol{\omega}}_1\times\rfloor\Delta\boldsymbol{p} + \Delta\boldsymbol{v}\end{aligned} \tag{16}$$

Similarly to the position, we use an additive error model for the velocity $\dot{\boldsymbol{v}}_r = \dot{\hat{\boldsymbol{v}}}_r + \dot{\Delta\boldsymbol{v}}$. With the state equation for the velocity (9), we get:

$$\dot{\Delta\boldsymbol{v}} = \boldsymbol{R}\boldsymbol{a}_2 - \boldsymbol{a}_1 - \lfloor\boldsymbol{\omega}_1\times\rfloor\boldsymbol{v} - \widehat{\boldsymbol{R}}\hat{\boldsymbol{a}}_2 + \hat{\boldsymbol{a}}_1 + \lfloor\hat{\boldsymbol{\omega}}_1\times\rfloor\hat{\boldsymbol{v}} \tag{17}$$

To yield an expression depending on the error state, we now replace $\boldsymbol{R}$ and apply the small angle approximation (50)

$$\boldsymbol{R} = \widehat{\boldsymbol{R}}\delta\boldsymbol{R} \approx \widehat{\boldsymbol{R}} \cdot (\boldsymbol{I}_3 + \lfloor\delta\boldsymbol{\theta}\times\rfloor) \tag{18}$$

Carrying out the multiplications, neglecting the higher order terms $\Delta\cdot\Delta$, $\delta\cdot\Delta$ finally yields the error state for the velocity:

$$\begin{aligned}\dot{\Delta\boldsymbol{v}} =& -\widehat{\boldsymbol{R}}\lfloor\hat{\boldsymbol{a}}_2\times\rfloor\delta\boldsymbol{\theta} + \lfloor\hat{\boldsymbol{v}}\times\rfloor\Delta\boldsymbol{\omega}_1 \\ & - \lfloor\hat{\boldsymbol{\omega}}_1\times\rfloor\Delta\boldsymbol{v} - \Delta\boldsymbol{a}_1 + \widehat{\boldsymbol{R}}\Delta\boldsymbol{a}_2\end{aligned} \tag{19}$$

For the remaining error states there is no change compared to the state equations, therefore:

$$\dot{\Delta\boldsymbol{\omega}}_i = \boldsymbol{n}_{\omega_i},\ \dot{\Delta\boldsymbol{a}}_i = \boldsymbol{n}_{a_i},\ \dot{\Delta\lambda} = n_\lambda,\ i = 1, 2 \tag{20}$$

C. State Covariance Prediction

To obtain the continuous system matrix $\boldsymbol{F}_c$ and noise matrix $\boldsymbol{G}_c$, we compute the Jacobians $\frac{\partial\dot{\tilde{\boldsymbol{x}}}}{\partial\tilde{\boldsymbol{x}}}$ and $\frac{\partial\dot{\tilde{\boldsymbol{x}}}}{\partial\boldsymbol{n}}$, $\boldsymbol{n} = \begin{bmatrix}\boldsymbol{n}_{\omega_1}^T & \boldsymbol{n}_{\omega_2}^T & \boldsymbol{n}_{a_1}^T & \boldsymbol{n}_{a_2}^T & n_\lambda\end{bmatrix}^T$. $\boldsymbol{F}_c$ is assumed to be constant over the integration period, therefore we can write for the discrete state transition matrix $\boldsymbol{F}_d$ [12]:

$$\boldsymbol{F}_d(\Delta t) = \exp(\boldsymbol{F}_c\Delta t) = \boldsymbol{I} + \boldsymbol{F}_c\Delta t + \frac{1}{2!}\boldsymbol{F}_c^2\Delta t^2 \ldots \tag{21}$$

In our implementation, we stop after the zeroth order term. Having $\boldsymbol{F}_d$, we can compute the noise covariance matrix $\boldsymbol{Q}_d$ for discrete time as proposed in [**?**]. With $\boldsymbol{c}_j = \sigma_j^2 \cdot \boldsymbol{I}_{1\times 3}$ and $\boldsymbol{Q}_c = \text{diag}(\begin{bmatrix}\boldsymbol{c}_{\omega_1} & \boldsymbol{c}_{\omega_2} & \boldsymbol{c}_{a_1} & \boldsymbol{c}_{a_2} & \sigma_\lambda^2\end{bmatrix})$:

$$\boldsymbol{Q}_d = \int_{\Delta t} \boldsymbol{F}_d(\tau) \cdot \boldsymbol{G}_c \cdot \boldsymbol{Q}_c \cdot \boldsymbol{G}_c^T \cdot \boldsymbol{F}_d(\tau)^T d\tau \tag{22}$$

We can now compute the new state covariance matrix according to the EKF equations:

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}_d \cdot \boldsymbol{P}_{k|k} \cdot \boldsymbol{F}_d^T + \boldsymbol{Q}_d \tag{23}$$

D. State Prediction

To perform state prediction, we integrate the state variables according to the equations in Section III-A. For the orientation quaternion, zeroth order integration of (4) yields a solution for $\hat{\bar{q}}_{k+1}$:

$$\hat{\bar{q}}_{k+1} = \hat{\bar{q}}_k + 0.5 \cdot \Delta t \cdot (\hat{\bar{q}}_k \otimes \hat{\boldsymbol{\omega}}_{2,\,k} - \hat{\boldsymbol{\omega}}_{1,\,k} \otimes \hat{\bar{q}}_k) \tag{24}$$

Predictions for $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{v}}$ are obtained similarly by zeroth order integration of (7) and (9):

$$\hat{\boldsymbol{p}}_{k+1} = \hat{\boldsymbol{p}}_k + (\hat{\boldsymbol{v}}_k - \lfloor\hat{\boldsymbol{\omega}}_{1,\,k}\times\rfloor\hat{\boldsymbol{p}}_k)\Delta t \tag{25}$$

$$\hat{\boldsymbol{v}}_{k+1} = \hat{\boldsymbol{v}}_k + (\boldsymbol{R}(\hat{\bar{q}}_k)\hat{\boldsymbol{a}}_{2,\,k} - \hat{\boldsymbol{a}}_{1,\,k} - \lfloor\hat{\boldsymbol{\omega}}_{1,\,k}\times\rfloor\hat{\boldsymbol{v}}_k)\Delta t \tag{26}$$

Since we run the prediction step between 100 and 300 Hz in practice, we consider zeroth order integration as sufficient. The remaining states do not change during state prediction.

E. Measurements

1) Vision: From the vision algorithm, we get an absolute measurement of the orientation $\bar{q}$ between both vehicles and a measurement of the distance $\boldsymbol{p}$ between the vehicles, which is scaled by λ. For the rotation, we use a multiplicative error model again:

$$\delta\bar{q} = \hat{\bar{q}}^* \otimes \bar{q} = \begin{bmatrix} q \\ \delta\boldsymbol{q} \end{bmatrix} \approx \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} \Rightarrow\ \tilde{\boldsymbol{z}}_{\bar{q}} = \delta\boldsymbol{\theta} \tag{27}$$

For the distance, we have an additive error model. Note that here, we have to scale the relative position $\boldsymbol{p}$ by λ to obtain an estimated measurement:

$$\begin{aligned}\tilde{\boldsymbol{z}}_{\boldsymbol{p}} &= \boldsymbol{p}\lambda - \hat{\boldsymbol{p}}\hat{\lambda} \\ \tilde{\boldsymbol{z}}_{\boldsymbol{p}} &= (\hat{\boldsymbol{p}} + \Delta\boldsymbol{p})(\hat{\lambda} + \Delta\lambda) - \hat{\boldsymbol{p}}\hat{\lambda} \quad | \ \Delta\boldsymbol{p}\Delta\lambda \approx 0 \\ \tilde{\boldsymbol{z}}_{\boldsymbol{p}} &= \hat{\boldsymbol{p}}\Delta\lambda + \hat{\lambda}\Delta\boldsymbol{p}\end{aligned} \tag{28}$$

We obtain the vision measurement matrix $\boldsymbol{H}_v$ by computing the Jacobian $\frac{\partial\tilde{\boldsymbol{z}}_v}{\partial\tilde{\boldsymbol{x}}}$ of $\tilde{\boldsymbol{z}}_v = \begin{bmatrix}\tilde{\boldsymbol{z}}_{\bar{q}}^T & \tilde{\boldsymbol{z}}_{\boldsymbol{p}}^T\end{bmatrix}^T$ w.r.t. $\tilde{\boldsymbol{x}}$. The measurement residual computes as follows, where im(.) denotes the imaginary part of the resulting quaternion:

$$\boldsymbol{r}_v = \begin{bmatrix} 2 \cdot \text{im}(\hat{\bar{q}}^* \otimes \bar{q}_{\text{meas}}) \\ \boldsymbol{p}_{\text{meas}} - \hat{\boldsymbol{p}} \end{bmatrix} \quad | \ (27)\ (28) \tag{29}$$

2) IMU: Measurements from the IMUs of the vehicles arrive at different times, therefore we split the updates into one for IMU 1 and one for IMU 2. We have an additive error model for all measurements which behaves linearly, $i = 1, 2$.

$$\tilde{\boldsymbol{z}}_{I,\,i} = \begin{bmatrix} \boldsymbol{\omega}_i - \hat{\boldsymbol{\omega}}_i \\ \boldsymbol{a}_i - \hat{\boldsymbol{a}}_i \end{bmatrix} \;\Rightarrow\; \boldsymbol{r}_{I,\,i} = \begin{bmatrix} \boldsymbol{\omega}_{i,\,\text{meas}} - \hat{\boldsymbol{\omega}}_i \\ \boldsymbol{a}_{i,\,\text{meas}} - \hat{\boldsymbol{a}}_i \end{bmatrix} \tag{30}$$

We obtain the measurement matrices $\boldsymbol{H}_{I,\,i}$ by computing the Jacobian $\frac{\partial \tilde{\boldsymbol{z}}_{I,\,i}}{\partial \tilde{\boldsymbol{x}}}$ of $\tilde{\boldsymbol{z}}_{I,\,i}$ w.r.t. $\tilde{\boldsymbol{x}}$.

3) Filter Update: The following steps are executed for the vision measurement and for both IMU measurements. Compute the Kalman gain $\boldsymbol{K}$ and the state correction $\Delta\boldsymbol{x}(+)$ according to the Kalman equations with the measurement covariance matrix $\boldsymbol{R}$.

$$\boldsymbol{K} = \boldsymbol{P} \cdot \boldsymbol{H}^T \cdot (\boldsymbol{H} \cdot \boldsymbol{P} \cdot \boldsymbol{H}^T + \boldsymbol{R})^{-1} \tag{31}$$

$$\Delta\boldsymbol{x}(+) = \boldsymbol{K} \cdot \boldsymbol{r} \tag{32}$$

Since we have a multiplicative error model for the relative orientation $\bar{q}$, we compute the correction for $\bar{q}$ as a small quaternion rotation. With $\delta\boldsymbol{q}(+) = 0.5 \cdot \delta\boldsymbol{\theta}(+)$:

$$\bar{q}(+) = \begin{bmatrix} \sqrt{1 - \delta\boldsymbol{q}(+)^T \cdot \delta\boldsymbol{q}(+)} & \delta\boldsymbol{q}(+)^T \end{bmatrix}^T \tag{33}$$

We have to ensure that the resulting correction quaternion $\bar{q}(+)$ is of unit length. Therefore, if $\delta\boldsymbol{q}(+)^T \cdot \delta\boldsymbol{q}(+) > 1$, we have to re-normalize the correction quaternion. The state update for the orientation is now computed as:

$$\hat{\bar{q}}_{k+1|k+1} = \hat{\bar{q}}_{k+1|k} \otimes \bar{q}(+) \tag{34}$$

For the remaining states $\boldsymbol{\omega}_1 \dots \boldsymbol{a}_2$, we have an additive error model, where the correction simply computes as:

$$\boldsymbol{x}_{\text{add},\,k+1|k+1} = \hat{\boldsymbol{x}}_{\text{add},\,k+1|k} + \boldsymbol{x}_{\text{add}}(+) \tag{35}$$

Finally, we update the state covariance matrix according to the Kalman equations:

$$\boldsymbol{U} = \boldsymbol{I}_{22} - \boldsymbol{K} \cdot \boldsymbol{H} \tag{36}$$

$$\boldsymbol{P}_{k+1|k+1} = \boldsymbol{U} \cdot \boldsymbol{P}_{k+1|k} \cdot \boldsymbol{U}^T + \boldsymbol{K} \cdot \boldsymbol{R} \cdot \boldsymbol{K}^T \tag{37}$$

IV. Observability Analysis

In order to ensure proper functioning of the filter described above, we apply a non-linear observability analysis. We apply the method of Hermann and Krener suggested in [13]. We refer to the work of Mirzaei [14] and Kelly [10] for details about how to apply this method to a system like ours. In this paper, we only focus on the obtained observability matrix $\boldsymbol{\Omega}$ and prove that it has column full rank. This indicates local weak observability as defined in [13]. We can express the system equations according to Section III in (10) as:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \dot{\bar{q}}^T & \mathbf{0} & \mathbf{0} & \dot{\boldsymbol{p}}^T & \dot{\boldsymbol{v}}^T & \mathbf{0} & \mathbf{0} & 0 \end{bmatrix}^T \tag{38}$$

The measurement equations can be summarized as a vector $\boldsymbol{h}(\boldsymbol{x}) = [\boldsymbol{h}_1(\boldsymbol{x})^T, ..., \boldsymbol{h}_M(\boldsymbol{x})^T]^T$ containing the M measurements as listed below:

$$\boldsymbol{h}_1 = \boldsymbol{p} \cdot \lambda \quad \boldsymbol{h}_2 = \bar{q} \quad \boldsymbol{h}_3 = \bar{q}^T \bar{q} = 1 \quad \boldsymbol{h}_4 = \boldsymbol{\omega}_1 \tag{39}$$

$$\boldsymbol{h}_5 = \boldsymbol{\omega}_2 \quad \boldsymbol{h}_6 = \boldsymbol{a}_1 \quad \boldsymbol{h}_7 = \boldsymbol{a}_2 \tag{40}$$

Using $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$, we can directly apply the method of Hermann and Krener and calculate the necessary Lie derivatives for the observability matrix $\boldsymbol{\Omega}$. The initial matrix is:

$$\boldsymbol{\Omega} = \begin{bmatrix} \nabla L^0\boldsymbol{h}_1 \\ \nabla L^0\boldsymbol{h}_2 \\ \nabla L^0\boldsymbol{h}_3 \\ \nabla L^0\boldsymbol{h}_4 \\ \nabla L^0\boldsymbol{h}_5 \\ \nabla L^0\boldsymbol{h}_6 \\ \nabla L^0\boldsymbol{h}_7 \\ \nabla L^1_{\boldsymbol{f}}\boldsymbol{h}_1 \\ \nabla L^2_{\boldsymbol{f}}\boldsymbol{h}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{B} \\ \boldsymbol{C} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{D} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{E} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{F} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{G} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{H} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{U} & \mathbf{0} & \boldsymbol{J} & \boldsymbol{K} & \mathbf{0} & \mathbf{0} & \boldsymbol{L} \\ \boldsymbol{U} & \boldsymbol{U} & \mathbf{0} & \boldsymbol{M} & \boldsymbol{N} & \boldsymbol{U} & \boldsymbol{U} & \boldsymbol{O} \end{bmatrix}$$

The matrices $\boldsymbol{A}$ to $\boldsymbol{O}$ are expanded as follows, while the matrices $\boldsymbol{U}$ are not required for our analysis:

$$\boldsymbol{A}, \boldsymbol{K} = \boldsymbol{I}_3\lambda \quad \boldsymbol{B} = \boldsymbol{p} \quad \boldsymbol{C} = \boldsymbol{I}_4 \quad \boldsymbol{E}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{H} = \boldsymbol{I}_3 \tag{41}$$

$$\boldsymbol{D} = 2\bar{q}^T \quad \boldsymbol{J} = -\lambda\lfloor\boldsymbol{\omega}_1\times\rfloor \quad \boldsymbol{L} = \boldsymbol{v} - \lfloor\boldsymbol{\omega}_1\times\rfloor\boldsymbol{p} \tag{42}$$

$$\boldsymbol{M} = \lambda\lfloor\boldsymbol{\omega}_1\times\rfloor^2 \quad \boldsymbol{N} = -\lambda\lfloor\boldsymbol{\omega}_1\times\rfloor(\boldsymbol{v} + \boldsymbol{I}) \tag{43}$$

$$\boldsymbol{O} = \lfloor\boldsymbol{\omega}_1\times\rfloor^2\boldsymbol{p} - 2\lfloor\boldsymbol{\omega}_1\times\rfloor\boldsymbol{v} + \boldsymbol{R}(\bar{q})\boldsymbol{a}_2 - \boldsymbol{a}_1 \tag{44}$$

To prove that $\boldsymbol{\Omega}$ has full column rank, we apply Gaussian elimination:

1) Since accelerations and angular rates are measured directly in our setup, their entries are the identity matrices in Ω. We can thus eliminate the columns containing $\boldsymbol{E}$, $\boldsymbol{F}$, $\boldsymbol{G}$, $\boldsymbol{H}$.
2) $\boldsymbol{C}$ is the derivative of the attitude w.r.t. itself, this is again an identity matrix. We can thus eliminate its column. Note that we do not need the constraint of q being a unit quaternion in $\boldsymbol{D}$, since we directly measure the orientation aid of the overlapping images.
3) We still need to prove that columns 4, 5 and 8 have full rank. To this end we analyze the matrix:

$$\boldsymbol{\Omega}' = \begin{bmatrix} \boldsymbol{A} & \mathbf{0} & \boldsymbol{B} \\ \boldsymbol{J} & \boldsymbol{K} & \boldsymbol{L} \\ \boldsymbol{M} & \boldsymbol{N} & \boldsymbol{O} \end{bmatrix} \tag{45}$$

We use Mathematica and show that $\boldsymbol{\Omega}'$ has full column rank if two conditions are met: first, the scale factor λ must be non-zero and second, there must be a "relative acceleration" between the vehicles in at least one axis. Thus we can also eliminate the remaining columns 4, 5 and 8 in $\boldsymbol{\Omega}$, proving that our proposed system is observable, under these conditions.

V. Results

In this section, we present results on simulation and real data. Simulations were performed for the full 3D case with 6 DoF, while real experiments were carried out using two MAVs.

A. Simulation Setup

In simulation, random parallel trajectories were generated such that the two vehicles keep a baseline of around $2\,\text{m}$ with some additional disturbance. Note that this is the most

difficult case for the filter, since there is very limited relative motion between the vehicles. One of the trajectories used in the experiments can be seen in Fig. 3. Yaw angles were also generated randomly, while vehicle position was differentiated to yield the corresponding velocity and acceleration for each vehicle. Given these accelerations and yaw angles, we computed the corresponding roll and pitch angles. Since we also need angular velocities, we chose quintic splines as trajectories which guarantee differentiable fourth derivatives (corresponding to angular velocities). Here, we assume that the rotation and scaled translation between the two cameras are given from an arbitrary vision algorithm that propagates its scale from one "stereo frame" to the next. Since this scale propagation is never perfect in practice, we introduced a scale drift of up to $\pm 10\,\%$ of the ground truth value (this is arbitrarily set to 0.5).

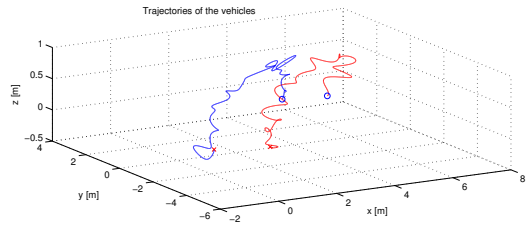

Fig. 3. Simulated trajectories in demonstrating full 3D motion for the position of the vehicles

B. Setup for Real Experiments

Real experiments were performed in the presence of a VICON motion capturing system which provides ground truth for the position of the moving MAVs at each instant within a flying volume of $10 \times 10 \times 10\,\mathrm{m}^3$. Two helicopters, set up similarly as in [15] were used to fly at a height of around $5\,\mathrm{m}$. Each helicopter is equipped with an IMU and a camera facing the ground. An onboard computer on each helicopter captures IMU data at 100 Hz and images at 20 Hz. Time is synchronized between the onboard computer and the IMU on each MAV while the computers of both MAVs are synchronized via the Network Time Protocol (NTP) over a WiFi connection. As the focus of this paper is on data fusion, we treat the vision processing as a black box that provides estimates for the orientation and pose between the MAVs. This makes our method not dependent on the choice of the vision algorithm. For the purpose of evaluating the performance of the proposed data fusion alone, both orientation and pose between the MAVs are provided by the VICON system. Therefore, the capture rate was reduced to 20 Hz, the position was scaled and noise was added. As in the simulation setup, the scale was set to 0.5 with a drift of $\pm 10\,\%$.

C. Discussion

Fig. 4 and Fig. 5 show the results of the simulation and real experiments for the orientation (top) and position (center) between the vehicles as well as the scaling (bottom) of the measured position. To demonstrate the capabilities of our filter, we initialized the scale with an error of 500 %.

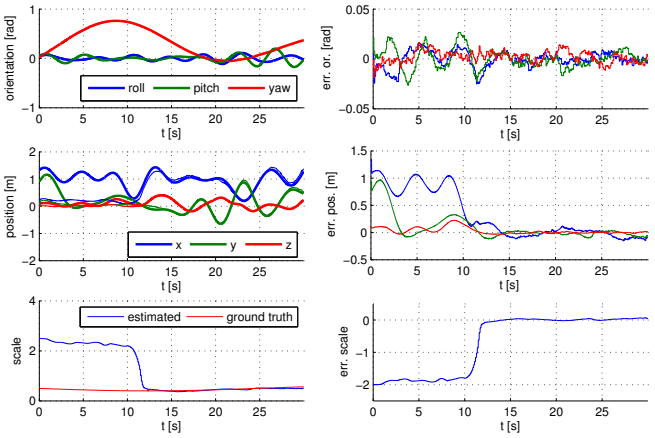

Fig. 4. Simulation results for orientation $\bar{q}$, position $\boldsymbol{p}$ between the vehicles and scale of the measured position. The scale drifted by $\pm 10\,\%$. The bold lines show ground truth while the thin lines show the filter estimates.

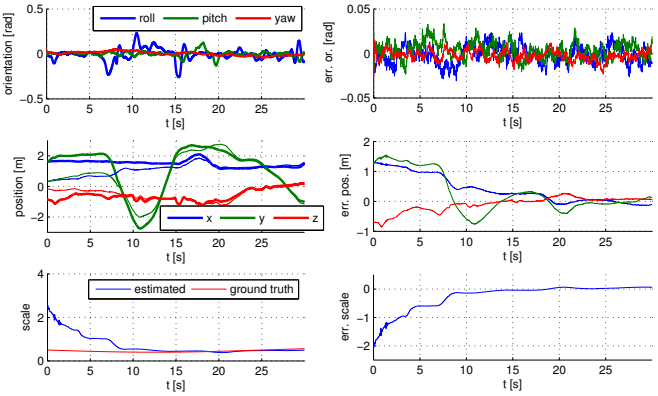

Fig. 5. Experimental results for orientation $\bar{q}$, position $\boldsymbol{p}$ between the vehicles and scale of the measured position. The scale drifted by $\pm 10\,\%$. The bold lines show ground truth while the thin lines show the filter estimates.

The orientation converges quickly to the real value since scale has no influence on it, while the error is not larger than $0.025\ \mathrm{rad}$ both in simulation and in the real experiment. Note the big changes in the yaw angle in the simulation: although this would not make sense in real experiments, it was done intentionally to show that the derived dynamics from Section III-A hold for large (and fast) rotations.

To obtain the absolute scale of the position between the vehicles, we fuse metric information from IMU measurements with the scaled position measurement from the vision algorithm. Since the IMUs provide only acceleration and angular velocity measurements, it becomes clear that we need motion between the vehicles for the filter to converge to the real value of the position and scale. How this convergence depends on motion can be seen in Fig. 4: there is small motion in the position plot until $t \approx 8\,\mathrm{s}$ while the estimated scale changes only slowly. Having sufficient motion, the scale converges quickly to the real value.

Fig. 6 shows the convergence behavior in simulation for different initializations ($\lambda_{init} = 2.5,\ 1.0,\ 0.5,\ 0.25,\ 0.1$) of the scale. On the left side, we allowed large motion and limited acceleration to $2\,\mathrm{m/s}^2$, while on the right side

TABLE I

RMS ERRORS AND MAXIMUM ERRORS.

	RMS $\tilde{\boldsymbol{p}}$	max $\tilde{\boldsymbol{p}}$	RMS $\tilde{\bar{q}}$	max. $\tilde{\bar{q}}$
sim. $\lvert\boldsymbol{a}_{\max}\rvert = 2\,\mathrm{m/s^2}$	.11 m	.35 m	.009 rad	.018 rad
sim. $\lvert\boldsymbol{a}_{\max}\rvert = 0.5\,\mathrm{m/s^2}$	.14 m	.49 m	.008 rad	.016 rad
real $\lvert\boldsymbol{a}_{\max}\rvert = 1.6\,\mathrm{m/s^2}$	.15 m	.29 m	.016 rad	.040 rad
real $\lvert\boldsymbol{a}_{\max}\rvert = 0.7\,\mathrm{m/s^2}$	.29 m	.56 m	.014 rad	.033 rad

acceleration was limited to $0.5\,\mathrm{m/s^2}$. Despite the large initialization error for $\lambda_{init} = 2.5,\ 0.1$ the filter is still able to converge to the real value. This is also the case for small motion, but at the cost of convergence time.

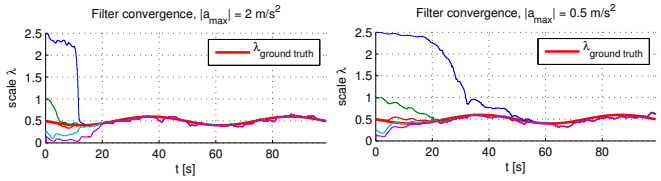

Fig. 6. Convergence behavior for $\lambda_{init} = 2.5,\ 1.0,\ 0.5,\ 0.25,\ 0.1$. Acceleration was limited to $2\,\mathrm{m/s^2}$ on the left and to $0.5\,\mathrm{m/s^2}$ on the right. Note that the filter is still able to converge despite the large initialization error.

Finally, Table I shows the RMS and maximum errors from the experiments for position $\boldsymbol{p}$ and orientation $\bar{q}$ between the vehicles. The errors for $\bar{q}$ were converted to Roll-Pitch-Yaw angles.

VI. CONCLUSIONS

We presented a method for recovering the relative configuration of two robots without any prior information on initialization conditions or their workspace. Fusing information from the onboard monocular camera and inertial sensor, we demonstrate accurate and real-time recovery of relative pose in absolute scale. The power of this method is demonstrated on both simulation and real data compared to ground truth, using MAVs to study performance across all range of 3D configurations.

VII. ACKNOWLEDGMENTS

This research was supported by the EC's 7th Framework Programe (FP7/2001-2013) under grant agreement no. 231855 (sFly). We are grateful to Prof. R. D'Andrea and S. Lupashin for providing access and support for the "Flying Machine Arena" [16]. F. Dellaert wishes to acknowledge Prof. M. Pollefeys for hosting his stay at ETH in summer 2010.

APPENDIX

For sequences of rotations, $\bar{q}_1 \otimes \bar{q}_2$ denotes a rotation of the coordinate system $\bar{q}_1$ by $\bar{q}_2$ in the local frame of $\bar{q}_1$, i.e. a rotation w.r.t. the moving frame $\bar{q}_1$.

Quaternion version of a vector $\boldsymbol{p} \in \mathbb{R}^3$, needed for transformation to another coordinate frame: $\bar{p} = \begin{bmatrix} 0 & \boldsymbol{p}^T \end{bmatrix}^T$.

The transformation of a vector ${}^1\boldsymbol{p}$ w.r.t. $\bar{q}_1$ to world-coordinates ${}^w\boldsymbol{p}$ is given by: ${}^w\bar{p} = \bar{q}_1 \otimes {}^1\bar{p} \otimes \bar{q}_1^*$.

Quaternion Multiplication:

$$\bar{q} \otimes \bar{p} = \begin{bmatrix} q & -\boldsymbol{q}^T \\ \boldsymbol{q} & q\boldsymbol{I}_3 + \lfloor \boldsymbol{q} \times \rfloor \end{bmatrix} \bar{p} = \begin{bmatrix} p & -\boldsymbol{p}^T \\ \boldsymbol{p} & p\boldsymbol{I}_3 - \lfloor \boldsymbol{p} \times \rfloor \end{bmatrix} \bar{q} \quad (46)$$

Small angle approximation for quaternions [17]:

$$\delta\bar{q} = \begin{bmatrix} \delta q_1 \\ \delta\boldsymbol{q} \end{bmatrix} = \begin{bmatrix} \cos(\delta\theta/2) \\ \boldsymbol{k}\sin(\delta\theta/2) \end{bmatrix} \approx \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} \quad (47)$$

Derivative of a quaternion and a rotation matrix, $\boldsymbol{\omega}$ is expressed in body coordinates:

$$\dot{\bar{q}} = \frac{1}{2}\bar{q} \otimes \bar{\omega}; \quad \dot{\boldsymbol{R}} = \boldsymbol{R} \cdot \lfloor \boldsymbol{\omega} \times \rfloor \quad (48)$$

Derivative of the conjugate quaternion [9]:

$$\begin{aligned} (\bar{q}^* \otimes \bar{q}) = 0 &\Leftrightarrow \dot{\bar{q}}^* \otimes \bar{q} + \bar{q}^* \otimes \dot{\bar{q}} = 0 \\ &\Rightarrow \dot{\bar{q}}^* = -\bar{q}^* \otimes \dot{\bar{q}} \otimes \bar{q}^* \end{aligned} \quad (49)$$

Small angle approximation for rotation matrices:

$$\boldsymbol{R}(\delta\bar{q}) \approx \boldsymbol{I}_3 + \lfloor \delta\boldsymbol{\theta} \times \rfloor \quad (50)$$

REFERENCES

[1] 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.

[2] 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.

[3] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics (T-RO)*, vol. 24, no. 5, pp. 932–945, 2008.

[4] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 26, no. 6, pp. 756–777, 2004.

[5] 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.

[6] D. Gallup, J. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[7] M. Bryson and S. Sukkarieh, "Architectures for cooperative airborne simultaneous localisation and mapping," *Journal of Intelligent and Robotic Systems (JIRS)*, vol. 55, no. 4, pp. 267–297, 2009.

[8] J. Solà, A. Monin, M. Devy, and T. Vidal-Calleja, "Fusing monocular information in multicamera SLAM." *IEEE Transactions on Robotics (T-RO)*, vol. 24, no. 5, pp. 958–968, 2008.

[9] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press, 2002.

[10] 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.

[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[12] P. S. Maybeck, *Stochastic models, estimation, and control*. Academic Press, 1979, vol. 1.

[13] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Transactions on Automatic Control*, vol. 22, no. 5, pp. 728 – 740, 1977.

[14] 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.

[15] 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.

[16] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[17] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Computer Science and Engineering, Tech. Rep. 2005-002, 2005.