This is an accepted version of the article: Perez-Paina, G., Paz, C., Kulich, M., Saska, M., and Araguás, G. (2016). **Fusion of Monocular Visual-Inertial Measurements for Three Dimensional Pose Estimation**. *Modelling and Simulation for Autonomous Systems*, 9991, which has been published in a final form at https://doi.org/10.1007/978-3-319-47605-6_20.

Fusion of Monocular Visual-Inertial Measurements for Three Dimensional Pose Estimation*

Gonzalo Perez-Paina[1], Claudio Paz[1], Miroslav Kulich[2], Martin Saska[3], and Gastón Araguás[1]

[1] Center for IT Research, National Technological University, Córdoba Branch, Argentina
{gperez,cpaz,garaguas}@frc.utn.edu.ar
[2] Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague, Czech Republic
kulich@ciirc.cvut.cz
[3] Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic
saska@labe.felk.cvut.cz

Abstract. This work describes a novel fusion schema to estimate the pose of a UAV using inertial sensors and a monocular camera. The visual motion algorithm is based on the plane induced homography using so called spectral features. The algorithm is able to operate with images presenting small amount of corner-like features, which gives more robustness to the state estimation. The key contribution of the paper is the use of this visual algorithm in a fusion schema with inertial sensors, exploiting the complementary properties of these two sensors. Results are presented in simulation with six degrees of freedom motion that satisfies dynamic constraints of a quadcopter. Virtual views are generated from this simulated motion cropped from a real floor image. Simulation results show that the presented algorithm would have enough precision to be used in an on-board algorithm to control the UAV in hovering operations.

Keywords: Sensor fusion · Visual odometry · Inertial sensors · Pose estimation · UAV · Kalman filter.

1 Introduction

Reduced size Unmanned Aerial Vehicles (UAV), also known as Micro-aerial Vehicles (MAV), have gained popularity in the last years. This is due mainly to their low cost and ability to operate either in outdoor and indoor environments. For

* This work was partially founded by the project "Multi-Robot Autonomous System", MINCyT ARC/13/13 & MEYS 7AMB14AR015, and the project "Mobile Robot Localization using Metric and Semantic information", UTN-PID 2173.

the case of multicopters [8], they have the additional ability to fly in a hovering mode, which is useful for inspection tasks.

The state estimation [16] is the problem of determining a set of different variables of interest from measurements of on-board sensors, which presents varying uncertainties and noise. Particularly, for state estimation in UAVs, these interesting variables are usually those that describe the UAV position and orientation in a three dimensional space. Due to this, the state estimation is usually solved robustly integrating or fusing information from different sensors, such as inertial units (accelerometers and gyroscopes), magnetic compasses, altimeters (barometers or sonars), cameras, GPS receivers, etc. Different sensors have their own advantages and disadvantages, depending on the flying conditions, like flight altitude, indoor/outdoor operation, etc. On the other hand, given the limited payload of MAVs which constraints the available computational power, one of typical setups for state estimation is to use inertial sensors (mainly accelerometers and gyroscopes) together with a monocular camera. Moreover, this result is useful for UAV application in GPS-denied areas, like indoor environments, urban canyons, etc.

Even though there are different algorithms for sensor fusion, the most widely used are those based on the Kalman filter and its variants. Recently, the algorithms based on graph-based optimization are gaining popularity. From the camera point of view there are also a variety of algorithms which can be used to compute the six degrees of freedom (6DOF) motion parameters. These are divided into two main categories, those that use image features (like corners, lines, etc.) or those that do not (optical flow, photometry error minimization, or frequency domain). For instance, in [6] an approach for for long term localization, stabilization, and navigation of micro-aerial vehicles (MAVs) in unknown environment is presented. This approach consists of extraction of information from pictures consequently captured using down-looking camera carried by the particular MAV. Corner-like scale invariant visual features are obtained from images of the surface under the MAV, and stored into a map that is represented by these features. The position of the MAV is then obtained through matching with previously stored features.

A general approach for multi-sensor fusion is described in [15], which allows to incorporate different kinds of sensors, either of absolute (GPS, magnetometer, altimeter) or relative (laser-based odometry and visual odometry) measurements. This sensor fusion is based on the unscented Kalman filter (UKF) with the stochastic cloning approach described in [13]. The fusion schema makes use also of a pose-only graph-based SLAM (being in fact a mixed approach using both filtering and optimization techniques) in order to cope with large uncertainty when GPS signal becomes available, after a long time offline. The limitation of the algorithm is related to the visual pose estimation which is highly affected by changes in the illumination for outdoor scenes. Also, the computational complexity of the UKF is higher than the EKF, which limits its applicability in on-board processing. The authors of [4] describe a fusion schema also based on the UKF, where the camera information is extracted from an optical flow algorithm. The

main goal is to achieve a robust estimation of the velocity and angular motion of the vehicle to be applied in dynamic motion control, and not to achieve a high-precision in the estimation as it is stated. The image processing part is based on FAST features (using Shi-Tomasi score) with the KLT tracker and evaluated against the standard epipolar constraint for error determination used by the filter correction. The authors of [5] present a method for increasing the accuracy of the standard visual-inertial odometry (VIO) system by removing the angular drift. The visual-inertial filter is based on the classical EKF from the approach presented in [11], using the unit quaternion for orientation representation and IMU bias estimation. The visual part is based on a line segment classification using inertial-aided RANSAC, where detected vanishing points are used in the update stage of the estimation filter. The limitation of this approach is that it only operates in man-made indoor environments, given that the vanishing points are obtained from the lines detected in the image in corridor like scenarios. The work in [17] describes an EKF-based tightly coupled monocular VIO with direct photometric error minimization, using a sparse (10-20) set of small patches (as small as 3×3 pixels). This allows a fast camera motion estimation and does not depend on corner-like features. The presented results show the comparison against ground truth obtained from a VICON system and they show a better performance in scenes without corner-like features and also with lines structures.

On the other side, using the optimization approach, the most representative work is presented in [14], which uses an IMU-camera fusion schema based on the sliding window graph-based formulation for non-linear optimization. In more details, the algorithm estimates the position and orientation (with unit quaternion representation) based on the standard IMU integration model (without bias estimation), and the image processing is based on KLT tracker with RANSAC for epipolar constraint. The IMU integration model is the same of those used in the fusion schema. The described algorithm was evaluated in different situations including autonomous hovering, autonomous trajectory tracking (both with VICON), and in autonomous flight in indoor environments, the algorithm was also evaluated with respect to the real-time performance.

An important issue in the implementation of a filter-based pose estimation is the parametrization used to represent the orientation. Different orientation representations present some advantages and disadvantages [12]. For instance, the Euler angles representation uses the minimum of three required parameters but they present singularities. On the other hand, the commonly used rotation representation given by a 3×3 orthogonal matrix has nine components with six constraints without singularities. Unit quaternions have the minimum redundant parametrization for orientation representation, which avoids discontinuities with only one constraint. However, using unit quaternions in filter-based orientation estimation presents the dilemma of how to treat the estimation error, which can be additive or multiplicative [10, 9]. Our implementation, as it will be described later, uses the multiplicative error model (multiplicative EKF) in which a non-singular representation is used for the estimated orientation, and a minimal representation for the deviation or error from this estimate.

In the present work, we describe a novel fusion schema to estimate the position and orientation of a UAV using inertial sensors and a monocular camera observing a considered flat scene. In addition, an altimeter is employed to disambiguate the scale factor of the camera measurements and to avoid the divergence of the altitude estimation resulting from the integration of the inertial sensors. The fusion is performed by an extended Kalman filter, in which the orientation is represented using unit quaternions [12, 18]. The motion information from the visual algorithm is based on the plane induced homography using so called spectral features [2]. The algorithm is able to operate with images presenting small amount of corner-like features [1], which gives more robustness to the state estimation. The key contribution of the paper stands in the use of this visual algorithm in a fusion schema with inertial sensors, exploiting the complementary properties of these two sensors. The current fusion approach is an extension of the work presented in [2], where the visual algorithm only returns the yaw angle measurement. In addition, this approach incorporates the complete three dimensional pose from the camera, i.e. position and orientation measurements.

Results are presented in simulation with six degrees of freedom motion that satisfies dynamic constraints of a specific UAV, particularly a quadcopter. Virtual views are generated from this simulated motion cropped from a real floor image. In this case, simulations are useful to validate the estimation when ground truth information is not available, which for the case of experiments with real UAV is usually obtained from an external tracking system. Simulation results show that the presented algorithm would have enough precision to be used in an on-board algorithm to control the UAV in hovering operations.

The model used for the filter-based sensor fusion does not incorporate the estimation of the inertial sensor biases, which is left as future work. Also, a future work will be to compare the performance of our IMU-camera fusion approach with that described in [17]. There, as was previously stated, the image processing is based on the intensity information of small patches instead of using the frequency domain as in our approach.

The rest of the paper is organized as follows. Section 2 gives a short overview of the proposed system. The models for the Kalman filter based sensor fusion are described in Sec. 3, while Sec. 4 details the used visual algorithm. The results are described in Sec. 5, while Sec. 6 finally presents the conclusions and future work.

2 System Overview

The proposed system consists of a down-looking monocular camera and an inertial measurement unit. These sensors are rigidly attached to the UAV and their relative pose is assumed to be known. An extended Kalman filter (EKF) is used to fuse the IMU and the camera information in order to obtain a six degrees of freedom motion estimation, i.e. the three dimensional position and orientation. The orientation is represented using a unit quaternion, which presents some advantages with respect to other representations, like algebraic simplicity and

numerical stability [12]. The vector state to be estimated by the filter is then composed by the position, the linear velocity and the unit quaternion. The inertial sensors (accelerometer and gyroscope) biases are not included in this vector state, thus they are not estimated by the filter. In some applications and mainly for flights of short time, the estimation of these biases averaging accelerometer and gyroscope readings before every flight would be sufficient. However, it has to be experimentally evaluated in real platforms. The prediction stage of the fusion filter integrates the IMU readings at each time step using the classical model, and when camera information is available, the update stage uses it to correct the estimation. Figure 1 shows the block diagram of the estimation filter.

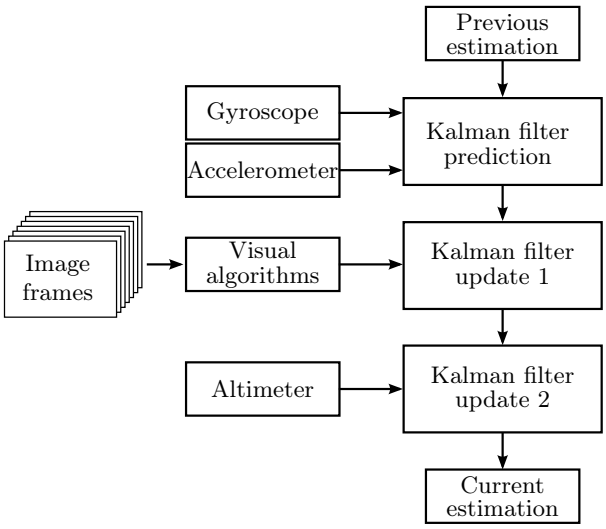


Fig. 1: Sensor fusion diagram

The camera motion is estimated using the homography induced by the observed floor, which is assumed to be flat, as shown schematically in Fig. 2. This is based on corresponding points which are obtained from the frequency domain. This spectral information corresponds to a fixed number of image patches distributed on each image, which we call spectral features. This kind of features performs better than interest points based on the image intensity when observing a floor with homogeneous texture [1]. Moreover, because their positions in the image plane are previously selected, they are always well distributed. The obtained homography is then decomposed into the motion parameters, i.e. the translation vector and the rotation matrix.

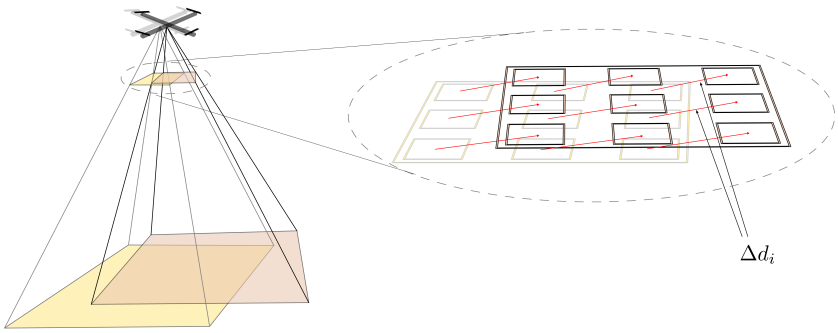


Fig. 2: Homography-based pose estimation

3 Sensor Fusion

In order to estimate a three dimensional pose using an extended Kalman filter, the system has to be represented stochastically by means of a process and a measurement models. These two models are generally expressed as $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w})$ and $\mathbf{z} = h(\mathbf{x}, \mathbf{v})$, respectively. The process model in the current implementation integrates the inertial sensor information, whilst there are two measurement models: one that incorporates the camera information, and the other one that incorporates the altimeter information.

The unknown state vector is composed of the three dimensional position, the linear velocity and the unit quaternion representing the orientation, i.e. $\mathbf{x} = \begin{bmatrix}\mathbf{p}^T & \mathbf{v}^T & \mathbf{q}^T\end{bmatrix}^T$. The measurement of the three axial accelerometer and gyroscope acting as input signal of the process model is defined as $\mathbf{u} = \begin{bmatrix}\mathbf{a}_m^T & \boldsymbol{\omega}_m^T\end{bmatrix}^T$. On the other hand, the camera measurement is composed of relative translation vector and orientation expressed in the unit quaternion between consecutive frames, i.e. ${}^c\mathbf{z} = \begin{bmatrix}{}^c\mathbf{p} & {}^c\mathbf{q}\end{bmatrix}^T$. This is used for the first update stage, while the altimeter measurement ${}^a\mathbf{z}$ is used in the second update.

As it was previously stated, the implementation is based on the multiplicative definition of the orientation error. That is, the estimation error is

$$\mathbf{x} \ominus \hat{\mathbf{x}} = \begin{bmatrix}\delta\mathbf{p} \\ \delta\mathbf{v} \\ \delta\mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{p} - \hat{\mathbf{p}} \\ \mathbf{v} - \hat{\mathbf{v}} \\ \hat{\mathbf{q}}^* \otimes \mathbf{q}\end{bmatrix} \quad , \tag{1}$$

where $\ominus$ is a general difference operator, which is the algebraic difference for the position and velocity vectors, and a Hamilton product for unit quaternion. The quaternion error in (1) for small angles can be expressed as

$$\delta\mathbf{q} = \begin{bmatrix}\delta q_w \\ \delta\mathbf{q}_v\end{bmatrix} = \begin{bmatrix}1 \\ \frac{1}{2}\delta\boldsymbol{\theta}\end{bmatrix}, \quad \mathbf{q}_v = \begin{bmatrix}q_x & q_y & q_z\end{bmatrix}^T \quad , \tag{2}$$

where $\delta\boldsymbol{\theta}$ is the attitude or orientation error. Hence, the final estimation error, used to define the covariance matrix representing the uncertainty in the estimation $\mathbf{P} = \mathbb{E}[\delta\mathbf{x}\delta\mathbf{x}^T]$, is defined as $\delta\mathbf{x} = \begin{bmatrix}\delta\mathbf{p} & \delta\mathbf{v} & \delta\boldsymbol{\theta}\end{bmatrix}^T$.

3.1 Inertial Sensor Models

The continuous-time kinematic model of three dimensional motion is given by

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{R}(\mathbf{q})\mathbf{a} + \mathbf{g}^n \\ \dot{\mathbf{q}} &= \frac{1}{2}\mathbf{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \ , \end{aligned} \tag{3}$$

where $\mathbf{p}$ and $\mathbf{q}$ represent the three dimensional pose in the navigation or earth fixed frame, and the inertial measurements $\mathbf{a}$ and $\boldsymbol{\omega}$ are expressed in the body or moving frame. $\mathbf{R}(\mathbf{q})$ is the representation of the orientation given by the unit quaternion as a rotation matrix, $\mathbf{g}^n$ is the gravitational acceleration expressed in the navigation frame, and the symbol $\otimes$ represents the Hamilton product [18], using the convention of the quaternion in which the first component is the scalar part.

The models of the inertial sensors, i.e. the accelerometer and gyroscope, are given by

$$\mathbf{a}_m = \mathbf{a} + \mathbf{w}_a \tag{4}$$

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{w}_\omega \ , \tag{5}$$

where the subscript m stands for the measured value, which is equal to the true value plus an additive white Gaussian noise $\mathbf{w}_a \sim \mathcal{N}(0, \sigma_a^2\mathbf{I})$ and $\mathbf{w}_\omega \sim \mathcal{N}(0, \sigma_\omega^2\mathbf{I})$. Then, the continuous-time covariance matrix of the process noise of (3) is $\mathbf{Q}^c = diag(\sigma_a^2\mathbf{I}, \sigma_\omega^2\mathbf{I})$.

For the application of the prediction stage of an EKF based on the model given by (3), it is needed to obtain: 1) a discrete solution of (3) to propagate the mean value of the state vector, where the orientation is represented in a non-singular way [9], and 2) a linearized version of (3) to propagate the covariance matrix [16] where the orientation error is expressed with three components.

The discrete-time form of (3), as a first-order approximation with a zero-order holder for the sensor measurements, is given by

$$\begin{aligned} \mathbf{p}_k &= \mathbf{p}_{k-1} + \Delta t \mathbf{v}_{k-1} \\ \mathbf{v}_k &= \mathbf{v}_{k-1} + \Delta t \mathbf{R}(\mathbf{q}_{k-1})\mathbf{a}_{m,k-1} \\ \mathbf{q}_k &= \left(\mathbf{I} + \frac{\Delta t}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_m)\right)\mathbf{q}_{k-1} \ , \end{aligned} \tag{6}$$

where

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & \lfloor \boldsymbol{\omega}_\times \rfloor \end{bmatrix} = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \ , \tag{7}$$

and $\lfloor \mathbf{x}_\times \rfloor$ is the skew-symmetric matrix of the vector $\mathbf{x}$ such as $\mathbf{a} \times \mathbf{b} = \lfloor \mathbf{a}_\times \rfloor \mathbf{b}$.

On the other hand, the first term of the linearized model or the error state model, is given by

$$\begin{aligned}\dot{\delta\mathbf{p}} &= \delta\mathbf{v} \\ \dot{\delta\mathbf{v}} &= -\mathbf{R}(\mathbf{q})\lfloor\mathbf{a}_{m\times}\rfloor\delta\boldsymbol{\theta} - \mathbf{R}(\mathbf{q})\mathbf{w}_a \\ \dot{\delta\boldsymbol{\theta}} &= -\lfloor\boldsymbol{\omega}_{m\times}\rfloor\delta\boldsymbol{\theta} - \mathbf{w}_\omega \ ,\end{aligned} \tag{8}$$

which results to be a linear model of the form $\dot{\delta\mathbf{x}} = \mathbf{A}\delta\mathbf{x} + \mathbf{D}\mathbf{w}$, where the matrices $\mathbf{A}$ and $\mathbf{D}$ are independent of the state $\delta\mathbf{x}$, and are defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{R}(\mathbf{q})\lfloor\mathbf{a}_{m\times}\rfloor \\ \mathbf{0} & \mathbf{0} & -\lfloor\boldsymbol{\omega}_{m\times}\rfloor \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\mathbf{R}(\mathbf{q}) & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix} \ .$$

Using also the first order or Euler approximation, the state transition matrix of the model given in (8) is $\mathbf{F} = e^{\mathbf{A}\Delta t} \approx \mathbf{I} + \Delta t\mathbf{A}$, where Δt is the time step. This transition matrix is used in the prediction state of the filter. The discrete-time covariance matrix [16] of the process noise is then expressed as

$$\mathbf{Q}_{k-1} = \int_{t_{k-1}}^{t_k} e^{\mathbf{A}(t_k-\tau)}\mathbf{D}\mathbf{Q}^c\mathbf{D}^T e^{\mathbf{A}^T(t_k-\tau)}d\tau \ , \tag{9}$$

where the sub-matrices, removing time-step subscript, are given by

$$\begin{aligned}\mathbf{Q}_{11} &= \frac{\sigma_a^2\Delta t^3}{3}\mathbf{I} \\ \mathbf{Q}_{22} &= \sigma_a^2\mathbf{I} - \frac{\sigma_\omega^2\Delta t^3}{3}\mathbf{R}\lfloor\mathbf{a}_{m\times}\rfloor^2\mathbf{R}^T \\ \mathbf{Q}_{33} &= \sigma_\omega^2\Delta t\mathbf{I} - \frac{\sigma_\omega^2\Delta t^3}{3}\lfloor\boldsymbol{\omega}_{m\times}\rfloor \\ \mathbf{Q}_{12} &= \frac{\sigma_a^2\Delta t^2}{2}\mathbf{I} = \mathbf{Q}_{21}, \quad \mathbf{Q}_{13} = \mathbf{Q}_{31} = \mathbf{0} \\ \mathbf{Q}_{23} &= -\frac{\sigma_\omega^2\Delta t^2}{2}\mathbf{R}\lfloor\mathbf{a}_{m\times}\rfloor - \frac{\sigma_\omega^2\Delta t^3}{3}\mathbf{R}\lfloor\mathbf{a}_{m\times}\rfloor\lfloor\boldsymbol{\omega}_{m\times}\rfloor, \quad \mathbf{Q}_{32} = \mathbf{Q}_{23}^T \ .\end{aligned}$$

3.2 Camera Measurement Model

The camera information, after applying the image processing algorithms, is the relative position and orientation of two consecutive frames both expressed in the first frame. This parameters are then used in the update stage of the fusion filter, where the measurement vector is defined as

$${}^c\hat{\mathbf{z}}_k^- = \begin{bmatrix} {}^c\hat{\mathbf{p}}_k^- \\ {}^c\hat{\mathbf{q}}_k^- \end{bmatrix} = \begin{bmatrix} \mathbf{R}(\mathbf{q}_p)^T(\hat{\mathbf{p}}_k^- - \mathbf{p}_p) \\ \mathbf{q}_p^* \otimes \hat{\mathbf{q}}_k^- \end{bmatrix} \ . \tag{10}$$

In (10), $\mathbf{p}_p$ and $\mathbf{q}_p$ are the position and orientation of the camera in the previous frame. It should be noted that the model given in (10) has to take into account

the rigid transformation between the camera and IMU coordinate systems for a real system. However, given that the presented results are based on simulated data, these two coordinate systems are considered to be coincident.

The measurement Jacobian of the camera measurement model (10) needed in the implementation of the Kalman filter is then

$$ {}^{c}\mathbf{H}_k = \left.\frac{\partial h_c(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} = \begin{bmatrix} \partial^c\mathbf{p}_k/\partial\mathbf{x} \\ \partial^c\mathbf{q}_k/\partial\mathbf{x} \end{bmatrix} = \begin{bmatrix} \partial^c\mathbf{p}_k/\partial\mathbf{p}_k & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} \\ \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \partial^c\mathbf{q}_k/\partial\mathbf{q}_k \end{bmatrix} \ , \quad (11) $$

where

$$ \frac{\partial^c\mathbf{p}_k}{\partial\mathbf{p}_k} = \mathbf{R}(\mathbf{q}_p)^T, \quad \frac{\partial^c\mathbf{q}_k}{\partial\mathbf{q}_k} = Q_l(\mathbf{q}_p^*) \ , $$

and $\mathbf{Q}_l(\cdot)$ is the left-matrix representing the quaternion product [18].

The residual or innovation $\boldsymbol{\nu}_k = \mathbf{z}_k \ominus \hat{\mathbf{z}}_k^-$ is computed as ${}^{c}\mathbf{p}_k - {}^{c}\hat{\mathbf{p}}_k^-$ for the position, and as $({}^{c}\mathbf{q}_k)^* \otimes {}^{c}\hat{\mathbf{q}}_k^-$ for the orientation quaternion. Then, this residual is used in the filter update stage taking into account also the multiplicative model, $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- \oplus \mathbf{K}_k\boldsymbol{\nu}_k$.

3.3 Altimeter Measurement Model

The sonar measurement is the distance from the sonar position and the floor plane. Here, we assume that the sonar position is at the center of the body frame, and its orientation is such that the measured distance is in the body frame z-axis direction. In this way, the measurement to be modeled is the distance defined in a 3D line in the direction of the body frame z-axis, from the body frame center to the point where this line intersects the floor plane. This is shown schematically in Fig. 3.

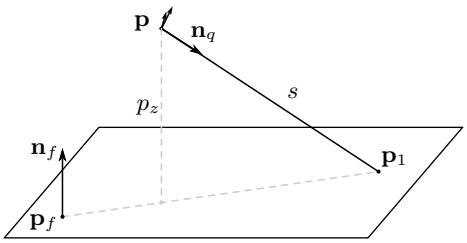


Fig. 3: Geometry of sonar measurement model

The floor plane is defined by the norm vector $\mathbf{n}_f = \begin{bmatrix}0\ 0\ 1\end{bmatrix}^T$ and the point $\mathbf{p}_f = \begin{bmatrix}0\ 0\ 0\end{bmatrix}^T$. The direction vector of the measurement is then

$$ \mathbf{n}_q = \mathbf{R}(\mathbf{q}) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2(q_x q_z + q_y q_w) \\ 2(q_y q_z - q_x q_w) \\ q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \ , $$

therefore the sought distance can be obtained as

$$s = \frac{-\mathbf{n}_f \cdot (\mathbf{p} - \mathbf{p}_f)}{\mathbf{n}_f \cdot \mathbf{n}_q} \ ,$$

where $\mathbf{p} = \begin{bmatrix} p_x \ p_y \ p_z \end{bmatrix}^T$ is the position of the body frame. Then, the sonar measurement model, removing time-step subscript, is

$$^a z = h_a(\mathbf{x}) = \frac{p_z}{q_w^2 - q_x^2 - q_y^2 + q_z^2} \ . \tag{12}$$

Finally, the measurement Jacobian of the altimeter measurement model is

$$^a\mathbf{H}_k = \left. \frac{\partial h_a(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} = \begin{bmatrix} \frac{\partial^a z}{\partial \mathbf{p}_k} \ \mathbf{0}_{1\times 3} \ \frac{\partial^a z}{\partial \mathbf{q}_k} \end{bmatrix} \ , \tag{13}$$

where

$$\frac{\partial^a z}{\partial \mathbf{p}} = \frac{1}{r}, \quad \frac{\partial^a z}{\partial \mathbf{q}} = \begin{bmatrix} -\frac{2p_z q_w}{r} & \frac{2p_z q_x}{r} & \frac{2p_z q_y}{r} & -\frac{2p_z q_z}{r} \end{bmatrix} \ ,$$

with $r = q_w^2 - q_x^2 - q_y^2 + q_z^2$.

4 Visual Algorithm for Pose Computation

The visual pose estimation is based on the principle that two consecutive images of a planar scene are related by a homography. The planar scene corresponds to the floor surface, which is assumed to be relatively flat, observed by the down-looking camera on the UAV. The spatial transformation of the camera, and therefore of the UAV, is encoded in this homography. Knowing the homography matrix that relates both images, the transformation parameters that describe the camera rotation and translation can be obtained.

In order to estimate the homography induced by the planar surface, a set of corresponding points on two consecutive images has to be obtained. This process is performed selecting a set of features in the first image and finding the corresponding set of features in the second one. Then, the image coordinates of each feature in both images conform the set of corresponding image points needed to calculate the homography.

The image features used in our approach are the so-called spectral features, a Fourier domain representation of an image patch. Selecting a set of patches in both images (the same number, with the same size and position), the displacement between them is proportional to the phase shift between the associated spectral features, and can be obtained using the Fourier shift theorem. This displacement, in addition to the feature center, determines the correspondence between features in both images: that is, the set of corresponding points needed to estimate the homography. An evaluation of spectral features for motion estimation can be seen in [3].

4.1 Plane-induced Homography and Decomposition

Given a 3D scene point $\mathbf{P}$, and two coordinate systems, CS_A and CS_B, the coordinates of the point $\mathbf{P}$ on each one can be denoted by $\mathbf{X}_A$ and $\mathbf{X}_B$ respectively. If $\mathbf{R}_A^B$ is the rotation matrix that changes the representation of a point in CS_A to CS_B, and $\mathbf{T}_B$ is the translation vector of the origin of CS_A w.r.t CS_B, then the representations of the point $\mathbf{P}$ relate to each other as

$$\mathbf{X}_B = \mathbf{R}_A^B \mathbf{X}_A + \mathbf{T}_B \ . \tag{14}$$

We suppose now that the point $\mathbf{P}$ belongs to a plane π, denoted in the coordinate system CS_A by its normal $\mathbf{n}_A$ and its distance to the coordinate origin d_A. Therefore, the following plane equation holds

$$(\mathbf{n}_A)^T \mathbf{X}_A = d_A \qquad \Rightarrow \qquad \frac{(\mathbf{n}_A)^T \mathbf{X}_A}{d_A} = 1 \ . \tag{15}$$

Replacing (15) into (14) we have

$$\mathbf{X}_B = \left(\mathbf{R}_A^B + \frac{\mathbf{T}_B}{d_A} (\mathbf{n}_A)^T \right) \mathbf{X}_A = \mathbf{H}_A^B \mathbf{X}_A \ , \tag{16}$$

with

$$\mathbf{H}_A^B = \left(\mathbf{R}_A^B + \frac{\mathbf{T}_B}{d_A} (\mathbf{n}_A)^T \right) \ . \tag{17}$$

The matrix $\mathbf{H}_A^B$ is a plane-induced homography, in this case induced by the plane π. As can be seen, this matrix encodes the transformation parameters that relate both coordinate systems ($\mathbf{R}_A^B$ and $\mathbf{T}_B$), and the structure parameters of the environment ($\mathbf{n}_A$ and d_A).

Considering now a moving camera associated to the coordinate system CS_A at time t_A and to CS_B at time t_B, according to the central projection model the relations between the 3D points and their projections on the camera normalized plane are given by

$$\lambda_A \mathbf{x}_A = \mathbf{X}_A, \quad \lambda_B \mathbf{x}_B = \mathbf{X}_B \ , \tag{18}$$

where $\lambda_A, \lambda_B > 0$. Using (18) in (16) we have

$$\lambda_B \mathbf{x}_B = \mathbf{H}_A^B \lambda_A \mathbf{x}_A \qquad \Rightarrow \qquad \mathbf{x}_B = \lambda \mathbf{H}_A^B \mathbf{x}_A \ , \tag{19}$$

with $\lambda = \frac{\lambda_A}{\lambda_B}$. Given that both vectors $\mathbf{x}_B$ and $\lambda \mathbf{H}_A^B \mathbf{x}_A$ have the same direction

$$\mathbf{x}_B \times \lambda \mathbf{H}_A^B \mathbf{x}_A = \lfloor \mathbf{x}_{B\times} \rfloor \mathbf{H}_A^B \mathbf{x}_A = 0 \ , \tag{20}$$

with $\lfloor \mathbf{x}_{B\times} \rfloor$ the skew-symmetric matrix associated to $\mathbf{x}_B$. Equation (20) is known as the planar epipolar restriction, and holds for all 3D points belonging to the plane π. Assuming that the camera is pointing to the ground (downward-looking

camera) and that the scene structure is approximately a planar surface, all the 3D points captured by the camera will fulfill this restriction.

The homography $\mathbf{H}_A^B$ represents the transformation of the camera coordinate systems between instant t_A and t_B, hence, it contains the information of the camera rotation and translation between these two instants. This homography can be estimated knowing at least four corresponding points of two images. In our case the correspondence between these points is calculated in the spectral domain, by means of the spectral features.

Following [**?**], a homography matrix $\mathbf{H}$ can be decomposed in four possible solutions, that is

$$\begin{aligned} &\left\{\mathbf{R}_1, \mathbf{n}_1, \tfrac{\mathbf{T}_1}{d_1}\right\}, \quad \left\{\mathbf{R}_1, -\mathbf{n}_1, \tfrac{-\mathbf{T}_1}{d_1}\right\} \ , \\ &\left\{\mathbf{R}_2, \mathbf{n}_2, \tfrac{\mathbf{T}_2}{d_2}\right\}, \quad \left\{\mathbf{R}_2, -\mathbf{n}_2, \tfrac{-\mathbf{T}_2}{d_2}\right\} \ . \end{aligned} \tag{21}$$

Then, in order to ensure that the plane inducing the homography $\mathbf{H}$ appears in front of the camera, each normal vector $\mathbf{n}_i$ must fulfill $n_z < 0$, and therefore only two solutions remain. These two solutions are both physically possible, but given that most of the time the camera on the UAV is facing-down, we choose the solution with the normal vector $\mathbf{n}$ closer to $[0, 0, -1]^T$ in terms of the norm L_2. A detailed description of the homography decomposition is presented in [1].

4.2 Spectral Features Correspondence

The so-called spectral feature refers to the Fourier domain representation of an image patch of $2^n \times 2^n$, where n is set accordingly to the allowed image displacement. The power of 2 of this patch size is selected based on the efficiency of the Fast Fourier Transform (FFT) algorithm. The number and position of spectral features in the image are set beforehand. Even though a minimum of four points is needed to estimate the homography, a higher number of features is used to increase the accuracy, and the RANSAC algorithm is used for outliers elimination.

Consider two consecutive frames, where spectral features on each image were computed. To determine the correspondence between features is equivalent to determine the displacement between them. This displacement can be obtained using the spectral information by means of the Phase Correlation Method (PCM). This method is based on the Fourier shift theorem, which states that the Fourier transforms of two identical but displaced images differ only in a phase shift.

Given two images i_A and i_B differing only in a displacement (u, v), such as $i_A(x, y) = i_B(x - u, y - v)$, their Fourier transforms are related by

$$I_A(\omega_x, \omega_y) = e^{-j(u\omega_x + v\omega_y)} I_B(\omega_x, \omega_y) \ , \tag{22}$$

where I_A and I_B are the Fourier transforms of images i_A and i_B, respectively; u and v are the displacements for each axis. From (22), the amplitudes of both transformations are the same and only differ in phase which is directly related

to the image displacement (u, v). Therefore, this displacement can be obtained using the cross-power spectrum (CPS) of the given transformations I_A and I_B. The CPS of two complex functions is defined as

$$\mathcal{C}(F, G) = \frac{F(\omega_x, \omega_y)G^*(\omega_x, \omega_y)}{|F(\omega_x, \omega_y)||G^*(\omega_x, \omega_y)|} \ , \tag{23}$$

where G^* is the complex conjugate of G. Using (22) in (23) over the transformed images I_A and I_B, gives

$$\frac{I_A I_B^*}{|I_A||I_B^*|} = e^{-j(u\omega_x + v\omega_y)} \ . \tag{24}$$

The inverse Fourier transform of (24) is an impulse located exactly in (u, v), which represents the displacement between the two images

$$\mathcal{F}^{-1}[e^{-j(u\omega_x + v\omega_y)}] = \delta(x - u, y - v) \ . \tag{25}$$

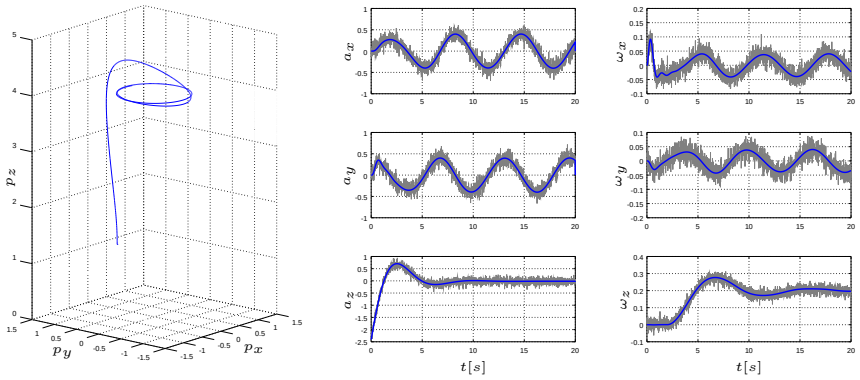
Using the discrete Fast Fourier Transform (FFT) algorithm instead of the continuous version, the result will be a pulse signal centered at (u, v) [19].

5 Results

5.1 Synthetic Dataset

A synthetic dataset is obtained from the dynamic simulation of a quadcopter based on the model described in [7]. The simulated quadcopter performs a given flight from which ground truth variables are obtained, these include position, linear velocity and orientation. The simulated flight is also used to obtain the ground truth inertial measurements, i.e. linear acceleration and angular rate in the vehicle frame, which are then affected by additive noise. Additionally, a large image represented a flat floor is cropped in order to obtain the camera image sequence, where the camera pose is also based on ground truth information. The ground truth pose of the UAV is also used to evaluate the performance of the proposed estimation filter. Figure 4 shows the data obtained from the simulation. Specially, Fig. 4a depicts the path followed by the quadcopter, whilst Fig. 4b presents the ground truth inertial sensor measurement in blue, and noisy measurement in gray.

The noise parameters of inertial sensors are the same of a real sensor, particularly the MicroStrain model 3DM-GX1, which are obtained from the vendor specifications. The continuous-time variance of the accelerometers is $\sigma_a = 0.4mg/\sqrt{Hz}$ and for the gyroscopes $\sigma_\omega = 3.5°/\sqrt{hour}$, which are converted to discrete time, dividing them by the square root of the sampling interval $\sqrt{\Delta t}$. As usually done, the accelerometer noise is affected by a factor of 10, to reflect the noise increase when the sensor is on-board a quadcopter. The simulated camera is of 640×480 with a pixel size of $5.6\mu m$. The image processing algorithm is set to use 42 patches of 128×128 pixels, equally distributed in the image, and with



(a) Ground truth path (b) IMU measurements

Fig. 4: Ground truth position and inertial sensor measurements

this setup the visual algorithms process 20 frames per second. Figure 5 shows some images obtained from the simulated camera, and processed by the visual algorithm. Figures 5a and 5b show two consecutive cropped images, whereas Fig. 5c shows the detected image feature displacements.

(a) Frame n (b) Frame $n+1$ (c) Features displacements

Fig. 5: Consecutive cropped images and detected feature displacements

As can be seen in Fig. 4a, in this dataset the quadcopter flights from a high of 1.5m to 4.0m, and describes a circular path in the $x-y$ plane. The initial high is not zero in order to obtain an appropriate image from the simulated camera.

5.2 Pose Estimation

The obtained results from the estimation filter using the described dataset represented in Fig. 4 and Fig. 5, are shown in Fig. 6 to 8. Figure 6 shows the estimated position, linear velocity and orientation expressed in Euler angles (ϕ, θ, ψ), whereas in Fig. 7 this orientation is represented as a unit quaternion. In these

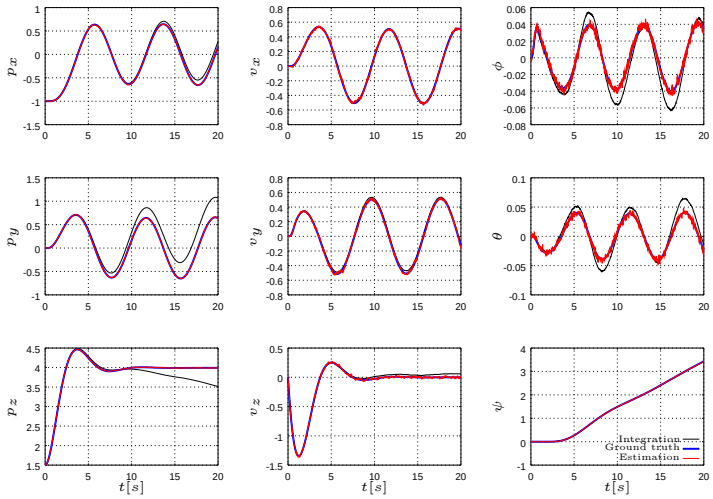


Fig. 6: Estimated position, linear velocity and orientation expressed in Euler angles

figures, the thick blue line is the ground-truth information obtained from the simulated dataset, the black line corresponds to the integration given by the inertial sensor model, and the red line is the estimation given by the filter. The integrated result is obtained from the filter prediction stage without any correction, whereas the final estimation is obtained with the correction stages, i.e. fusing all sensor measurements together. The orientation result given in Euler angles of Fig. 6 is obtained converting the unit quaternion of Fig. 7, which is a component of the filter state vector. As can bee seen in Fig. 6 and Fig. 7, the estimation obtained from the fusion of all the sensors avoids a typical unbounded integration error when using only inertial sensors.

Figure 8 shows the filter estimation error, together with the $\pm 3\sigma$ error bound given by the diagonal elements of the covariance matrix. In this case, the estimation error in orientation is expressed in Euler angles, given that the filter covariance matrix is based on this minimal representation, as previously stated. This figure shows that the initial uncertainties reach their final values in a few time steps and they remain bounded for the next time steps.

Figure 9 shows inertial sensor measurements corrupted by a constant bias. The bias for each axis of the accelerometer is $b_a = 0.5 m/s^2$, and for each axis of the gyroscope is $b_\omega = 0.1 rad/s$. As previously mentioned, the biases in the inertial sensors can be estimated before every flight by means of averaging accelerometer and gyroscope reading. Figure 10 shows the estimation error for the cases in which the biases are partially and completely removed. The red line corresponds to a remaining of 25% of the bias, whereas the blue line is for a

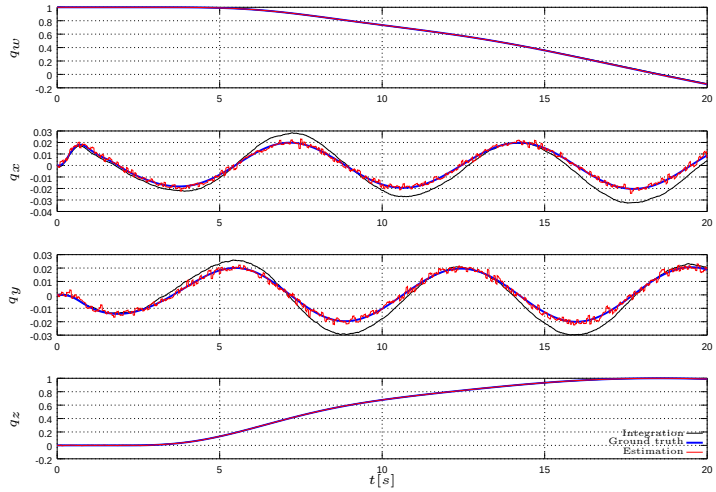


Fig. 7: Estimated orientation expressed in unit quaternion

remaining of 50%. The gray line is the estimation error without any bias, as it was shown in Fig. 8.

As it can be observed in Fig. 10, the estimation error in the orientation given by the Euler angles (three rightmost plots) are negligibly affected by the

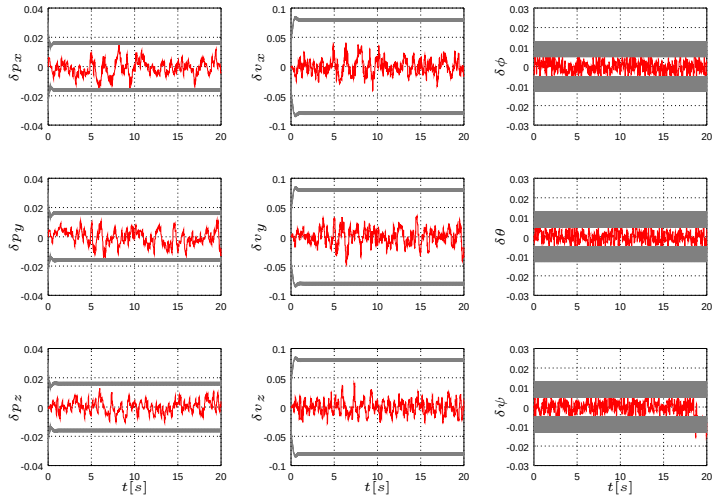


Fig. 8: Estimation error and $\pm 3\sigma$ uncertainty in position, linear velocity and Euler angles

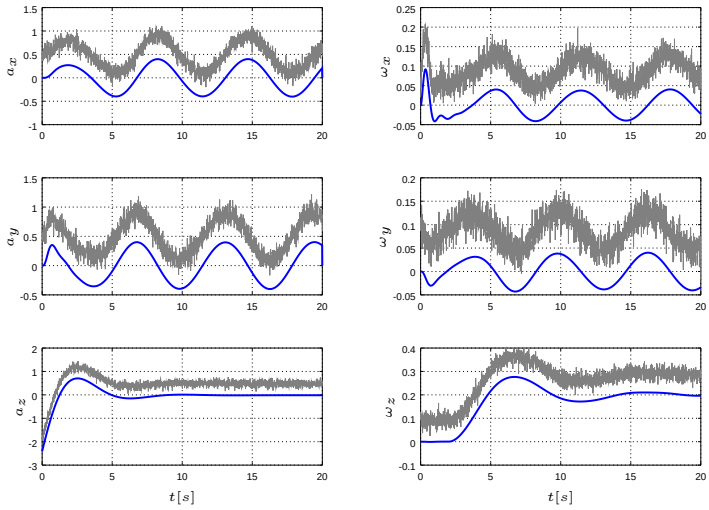


Fig. 9: IMU measurements corrupted with constant bias

gyroscope bias, whereas the linear velocity and position are affected more significantly. This is mainly due to that the position and linear velocity are affected for the biases of both sensors, i.e. the accelerometer and gyroscope.

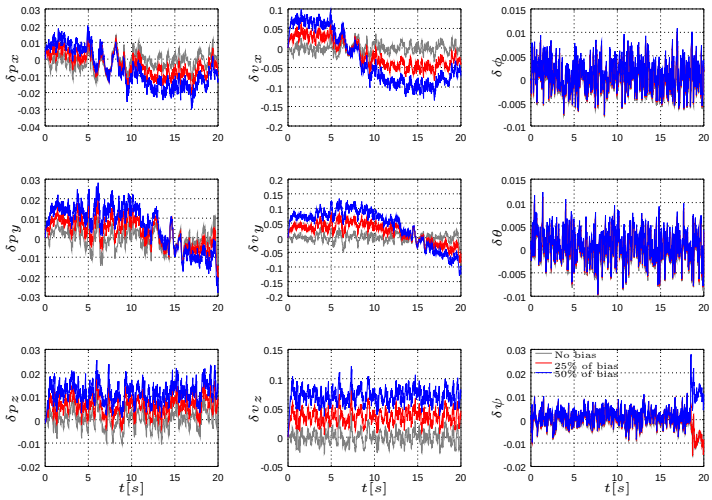


Fig. 10: Estimation error for IMU measurement corrupted constant with bias

6 Conclusion and Future Work

In this work a novel schema for estimating the position and orientation of a UAV using inertial sensors and a monocular camera has been presented. The visual algorithm uses the so called spectral features, and is based on the plane-induced homography obtained when observing a flat floor. The presented results are based on a dynamic simulation of a quadcopter, which allows to obtain the ground truth position and orientation, together with the ideal inertial sensor measurements. This ground truth information is also useful for the performance evaluation of the proposed fusion algorithm.

As the results show, the three dimensional pose estimation has a precision that seems promising to be applied in UAV onboard algorithms for control purposes. Also, it was shown that a not precise bias correction, as high as 50% of the real bias, does not affect pose estimation considerably, specially for short time flights.

As future work, the next step is to test the proposed fusion schema with a real UAV, in order to evaluate how the estimation is affected by real sensor noise and unmodeled sensor bias. Also, a new filter model can be formulated incorporating the sensor biases in the state vector in order to be estimated. This has the main disadvantage of a high dimensional state vector requiring a higher computational power to compute the filter equations. Finally, the proposed visual algorithm can be compared with other similar approaches, particularly those that does not use corner-like features.

References

1. Araguás, G., Paz, C., Paina, G.P., Canali, L.: Visual homography-based pose estimation of a quadrotor using spectral features. In: 2015 Latin America Congress on Computational Intelligence (LA-CCI). pp. 1–6 (Oct 2015)
2. Araguás, G., Paz, C., Gaydou, D., Perez Paina, G.: Quaternion-based orientation estimation fusing a camera and inertial sensors for a hovering UAV. Journal of Intelligent & Robotic Systems 77(1), 37–53 (2015), http://dx.doi.org/10.1007/s10846-014-0092-z
3. Araguás, G., Paz, C., Perez Paina, G., Canali, L.: Visual Homography-based Pose Estimation of a Quadrotor using Spectral Features, chap. 2. Designing with Computational Intelligence, Springer Book Series "Studies in computational intelligence" ((in press))
4. Bloesch, M., Omari, S., Fankhauser, P., Sommer, H., Gehring, C., Hwangbo, J., Hoepflinger, M., Hutter, M., Siegwart, R.: Fusion of optical flow and inertial measurements for robust egomotion estimation. In: Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on. pp. 3102–3107 (Sept 2014)
5. Camposeco, F., Pollefeys, M.: Using vanishing points to improve visual-inertial odometry. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on. pp. 5219–5225 (May 2015)
6. Chudoba, J., Kulich, M., Saska, M., Báča, T., Přeučil, L.: Exploration and mapping technique suited for visual-features based localization of mavs. Journal of

Intelligent & Robotic Systems pp. 1–19 (2016), http://dx.doi.org/10.1007/s10846-016-0358-8
7. Corke, P.I.: Robotics, Vision & Control: Fundamental Algorithms in Matlab. Springer (2011)
8. Gaydou, D., Suarez, G., Paz, C., Perez Paina, G., Araguás, G.: Robot volador no tripulado QA3. Diseño y construcción de un cuatrirrotor para experimentación. In: Proceedings of the VIII Jornadas Argentinas de Robótica (JAR) (2014)
9. Markley, F.L.: Attitude Error Representations for Kalman Filtering. Journal of Guidance, Control, and Dynamics 26, 311–317 (Mar 2003)
10. Markley, F.L.: Multiplicative vs. additive filtering for spacecraft attitude determination. In: in Dynamics and Control of Systems and Structures in Space (2004)
11. Mourikis, A., Roumeliotis, S.: A multi-state constraint kalman filter for vision-aided inertial navigation. In: Robotics and Automation, 2007 IEEE International Conference on. pp. 3565–3572 (April 2007)
12. Pucheta, M.A., Paz, C.J., Pereyra, M.E.: Representaciones cinemáticas de orientación y ecuaciones de estimación. In: XXI Congreso sobre Métodos Numéricos y sus Aplicaciones ENIEF. vol. XXXIII, pp. 2303–2324 (2014)
13. Roumeliotis, S., Burdick, J.: Stochastic cloning: a generalized framework for processing relative state measurements. In: Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on. vol. 2, pp. 1788–1795 vol.2 (2002)
14. Shen, S., Michael, N., Kumar, V.: Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on. pp. 5303–5310 (May 2015)
15. Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V.: Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft mav. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on. pp. 4974–4981 (May 2014)
16. Simon, D.: Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. Wiley-Interscience (2006)
17. Tanskanen, P., Naegeli, T., Pollefeys, M., Hilliges, O.: Semi-direct ekf-based monocular visual-inertial odometry. In: Intelligent Robots and Systems (IROS 2015), 2015 IEEE/RSJ International Conference on (2015)
18. Trawny, N., Roumeliotis, S.I.: Indirect Kalman filter for 3D attitude estimation. Tech. Rep. 2005-002, University of Minnesota, Dept. of Comp. Sci. & Eng. (Mar 2005)
19. Zitová, B., Flusser, J.: Image registration methods: a survey. Image and Vision Computing 21(11), 977–1000 (2003)