Low computational-complexity algorithms for vision-aided inertial navigation of micro aerial vehicles

Chiara Troiani, Agostino Martinelli, Christian Laugier, Davide Scaramuzza

Low Computational-Complexity Algorithms for Vision-Aided Inertial Navigation of Micro Aerial Vehicles

Chiara Troiani[a], Agostino Martinelli[a], Christian Laugier[a], Davide Scaramuzza[b]

[a] *INRIA Rhone Alpes, Montbonnot, France*
[b] *Robotics and Perception Group, University of Zurich, Switzerland*

Abstract

This paper presents low computational-complexity methods for micro-aerial-vehicle localization in GPS-denied environments. All the presented algorithms rely only on the data provided by a single onboard camera and an Inertial Measurement Unit (IMU). This paper deals with outlier rejection and relative-pose estimation. Regarding outlier rejection, we describe two methods. The former only requires the observation of a single feature in the scene and the knowledge of the angular rates from an IMU, under the assumption that the local camera motion lies in a plane perpendicular to the gravity vector. The latter requires the observation of at least two features, but it relaxes the hypothesis on the vehicle motion, being therefore suitable to tackle the outlier detection problem in the case of a 6DoF motion. We show also that if the camera is rigidly attached to the vehicle, motion priors from the IMU can be exploited to discard wrong estimations in the framework of a 2-point-RANSAC–based approach. Thanks to their inherent efficiency, the proposed methods are very suitable for resource-constrained systems. Regarding the pose estimation problem, we introduce a simple algorithm that computes the vehicle pose from the observation of three point features in a single camera image, once that the roll and pitch angles are

[☆]This research was supported by the Swiss National Science Foundation through project number 200021-143607 ("Swarm of Flying Cameras"), the National Centre of Competence in Research (NCCR) Robotics, and by The French National Research Agency ANR 2014 through the project VIMAD.

estimated from IMU measurements. The proposed algorithm is based on the minimization of a cost function. The proposed method is very simple in terms of computational cost and, therefore, very suitable for real-time implementation. All the proposed methods are evaluated on both synthetic and real data.
Keywords: Outlier detection, Micro Aerial Vehicle, Quadrotor, Vision-Aided Inertial Navigation, Camera pose estimation, GPS-denied navigation, Structure from Motion.

1. Introduction

In recent years, flying robotics has received significant attention from the robotics community. The ability to fly allows easily avoiding obstacles and quickly having an excellent birds eye view. These navigation facilities make flying robots the ideal platform to solve many tasks like exploration, mapping, reconnaissance for search and rescue, environment monitoring, security surveillance, inspection etc. In the framework of flying robotics, micro aerial vehicles (MAV) have a further advantage. Due to the small size they can also be used in narrow out- and indoor environment and they represent only a limited risk for the environment and people living in it. However, for such operations today's systems navigating on GPS information only are not sufficient any more. Fully autonomous operation in cities or other dense environments requires the MAV to fly at low altitude or indoors where GPS signals are often shadowed.

A relevant issue for MAVs is the limited autonomy and payload. This brings researchers to focus their attention on low computational complexity algorithms and low-weight sensors.

Recent works on autonomous navigation of micro helicopters in GPS-denied environments have demonstrated the ability to perform basic maneuvers using as little as a single camera and an Inertial Measurement Unit (IMU) onboard the vehicle [1], [2], [3]. These systems rely on well-known theory of Visual Odometry [4], [5] which consists of incrementally estimating the pose of a vehicle by examining the changes that motion induces on visually-tracked interest points.

These points consist of salient and repeatable features that are extracted and matched across consecutive images according to their similarity.

One of the primary problems in Visual Odometry is wrong data associations. Matched features between two different camera views are usually affected by outliers. This is due to the fact that changes in viewpoint, occlusions, image noise, illumination changes and image noise are not modeled by feature-matching techiniques. To perform a robust motion estimation, it is essential to remove the outliers. The outlier detection task is usually very expensive from a computational point of view and is based on the exploitation of the geometric constraints induced by the motion model.

The standard method for model estimation from a set of data affected by outliers is RANSAC (RANdom SAmple Consensus) [6]. It consists of randomly selecting a set of data points, computing the corresponding model hypothesis, and verifying this hypothesis on all the other data points. The solution is the hypothesis with the highest consensus. The number of iterations (N) necessary to guarantee a robust outlier removal is [6]:

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)} \tag{1}$$

where s is the number of data points from which the model can be computed, ε is the percentage of outliers in the dataset, p is the probability of success requested. Table 1 shows the number of iterations (N) with respect to the number of points necessary to estimate the model (s). The values are computed for $p = 0.99$ and $\epsilon = 0.5$. Note that N is exponential in the number of data points s; this means that it is extremely important to look for minimal parametrizations of the model, in order to reduce the number of iterations, which is of utmost importance for vehicles equipped with a computationally-limited embedded computer.

In this paper, which is an extension of our previous works [7], [8], we present low computational complexity algorithms to tackle the problem of Micro Aerial Vehicle motion estimation in GPS denied environment and outlier detection between two different views. All the methods rely on the measurements provided by an onboard monocular camera and an IMU. The rest of the paper is organized

Number of points (s)	1	2	3	5	8
Number of Iterations (N)	7	16	35	145	1177

Table 1.

as follows.

The next section provides the state of the art in outlier detection and pose estimation respectively. Section 3 describes the proposed methods to detect outliers between two consecutive views of a camera rigidly attached to an IMU and presents the extension of our previous work [8] which consist in relaxing the hypothesis on the camera motion and making the approach suitable for any 6DoF motion. The specific case of a camera mounted onboard a quadrotor is also presented to show that motion priors provided by the IMU can be used to discard wrong estimations in the framework of a 2-point RANSAC approach. Section 4 tackles the problem of pose estimation providing a simple algorithm able to estimate the vehicle pose from the observation of three point features in a single camera image, once that the roll and pitch angles are obtained by the inertial measurements. Section 5 presents the performance evaluation of the proposed methods on synthetic and real data. Finally, conclusions are provided in Section 6.

2. Related works

2.1. Outlier detection

When the camera is calibrated, its six degrees of freedom (DoF) motion can be inferred from a minimum of five-point correspondences, and the first solution to this problem was given in 1913 by Kruppa [9]. Several five-point minimal solvers were proposed later in [10],[11],[12], but an efficient implementation, based on [11], was found only in 2003 by Nister [13] and later revised in [14]. Before that, the six- [15], seven- or eight- solvers were commonly used. However, the five-point solver has the advantage that it works also for planar scenes. A

more detailed analysis of the state of the art can be found in [4].

Despite the five-point algorithm represents the minimal solver for 5DoF motion of calibrated cameras, in the last few decades there have been several attempts to exploit different cues to reduce the number of motion parameters. In [16], the authors proposed a three-point minimal solver for the case of two known camera-orientation angles. For instance, this can be used when the camera is rigidly attached to a gravity sensor (in fact, the gravity vector fixes two camera-orientation angles). Later, the work in [17] improved on [16] by showing that the three-point minimal solver can be used in a four-point (three-plus-one) RANSAC scheme. The three-plus-one stands for the fact that an additional far scene point (ideally, a point at infinity) is used to fix the two orientation angles. Using their four-point RANSAC, they also showed a successful 6 DoF VO. A two-point minimal solver for 6-DoF Visual Odometry was proposed in [18] and further employed in [19] to achieve high-accuracy localization. This method uses the full rotation matrix from an IMU rigidly attached to the camera. In our work we exploit motion priors from IMU in order to discard wrong estimates. In the case of planar motion, the motion model complexity is reduced to 3 DoF and can be parameterized with two points as described in [20]. For wheeled vehicles, the work in [21, 22] showed that the motion can be locally described as planar and circular, and, therefore, the motion model complexity is reduced to 2 DoF, leading to a one-point minimal solver. Additionally, it was shown that, by using a simple histogram voting technique, outliers can be found in as little as a single iteration. In [19] the authors propose a one-point algorithm for RGBD or stereo cameras which relies on IMU measurements to recover the relative rotation. A performance evaluation of five-, two-, and one-point RANSAC algorithms for Visual Odometry was finally presented in [23].

2.2. Pose estimation

In [24], inertial and visual sensors are used to perform egomotion estimation. The sensor fusion is obtained by an Extended Kalman Filter (EKF) and by an Unscented Kalman Filter (UKF). The approach proposed in [25] extends

the previous one by also estimating the structure of the environment where the motion occurs. In particular, new landmarks are inserted on line into the estimated map. This approach has been validated by conducting experiments in a known environment where a ground truth was available. Also, in [26] an EKF has been adopted. In this case, the proposed algorithm estimates a state containing the robot speed, position and attitude, together with the inertial sensor biases and the location of the features of interest. In the framework of airbone SLAM, an EKF has been adopted in [27] to perform $3D-$SLAM by fusing inertial and vision measurements. It was observed that any inconsistent attitude update severely affects any SLAM solution. The authors proposed to separate attitude update from position and velocity update. Alternatively, they proposed to use additional velocity observations, such as air velocity observation. More recently, a vision based navigation approach in unknown and unstructured environments has been suggested [28].

Recent works investigate the observability properties of the vision-aided inertial navigation system [29], [30], [31], [32], [33], [34] and [35]. In particular, in [33], the observable modes are expressed in closed-form in terms of the sensor measurements acquired during a short time-interval.

Visual UAV pose estimation in GPS-denied environments is still challenging. Many implementations rely on visual markers, such as patterns or blobs, located in known positions [36], [37], [38]. Those approaches have the drawback that can work only in structured environment. In [39] Visual-Inertial Attitude Estimation is performed using image line segments for the correction of accumulated errors in integrated gyro rates when an unmanned aerial vehicle operates in urban areas. The approach will not work in environments that do not present a strong regularity in structure.

In [40], [41] the authors developed a very robust Vision Based Navigation System for micro helicopters. Their pose estimator is based on a monocular VSLAM framework (PTAM, Parallel Tracking and Mapping [42]). This software was originally developed for augmented reality and improved with respect to robustness and computational complexity. The resulting algorithm can be used

in order to make a monocular camera a real-time onboard sensor for pose estimates. This allowed the first aerial vehicle that uses onboard monocular vision as a main sensor to navigate through an unknown GPS-denied environment and independently of any external artificial aids [43], [41].

Natraj et al. [44] proposed a vision based approach, close to structured light, for roll, pitch and altitude estimation of UAV. They use a fisheye camera and a laser circle projector, assuming that the projected circle belongs to a planar surface. The latter must be orthogonal to the gravity vector in order to allow the estimation of the aforementioned quantities. The attitude estimation of the planar surface becomes crucial in order to extend the operational environment of UAVs. Shipboard operations, search and rescue cooperation between ground and aerial robots, low altitude manoeuvres, require to attenuate the position error and to track the platform attitude.

3. Outlier detection

In this section we present two low computational complexity methods to perform the outlier detection task between two different views of a monocular camera rigidly attached to an inertial measurement unit. The first one only requires the observation of a single feature in the scene and the knowledge of the angular rates provided by an inertial measurement unit, under the assumption that the local camera motion lies on a plane perpendicular to the gravity vector. In the second one we relax the hypothesis on the camera motion. The observation consists of two features in the scene (instead of only one) and of angular rates from inertial measurements. We show that if the camera is onboard a quadrotor vehicle, motion priors from inertial measurements can be used to discard wrong data association. Both the methods are evaluated on synthetic and real data.

3.1. Epipolar Geometry

Before going on, we would like to recall some definitions about epipolar geometry. When a camera is calibrated, it is always possible to project the

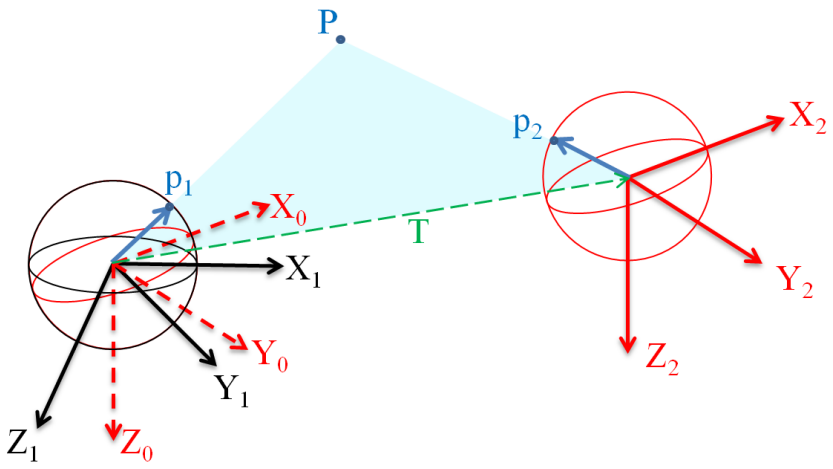


Figure 1: Epipolar constraint. $\mathbf{p_1}$, $\mathbf{p_2}$, T and P lie on the same plane (the *epipolar plane*).

feature coordinates onto a unit sphere. This allows us to make our approach independent of the camera model.

Let $\mathbf{p_1} = (x_1, y_1, z_1)$ and $\mathbf{p_2} = (x_2, y_2, z_2)$ be the image coordinates of a point feature seen from two camera positions and back projected onto the unit sphere (i.e., $\| \mathbf{p_1} \| = \| \mathbf{p_2} \| = 1$) (Figure 1).

The image coordinates of point features relative to two different unknown camera positions must satisfy the *epipolar constraint* (Figure 1) [45].

$$\mathbf{p_2}^T \mathbf{E} \mathbf{p_1} = 0 \tag{2}$$

where $\mathbf{E}$ is the *essential matrix*, defined as $\mathbf{E} = [\mathbf{T}]_\times \mathbf{R}$. $\mathbf{R}$ and $\mathbf{T} = [T_x, T_y, T_z]^T$ describe the relative rotation and translation between the two camera positions, and $[\mathbf{T}]_X$ is the skew symmetric matrix:

$$[\mathbf{T}]_\times = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \tag{3}$$

According to equation (2), the essential matrix can be computed given a set

of image coordinate points. $\mathbf{E}$ can then be decomposed into $\mathbf{R}$ and $\mathbf{T}$ [45].

The minimum number of feature correspondences needed to estimate the essential matrix is function of the degrees of freedom of the camera's motion. In the case of a monocular camera performing a 6DoF motion (three for the rotation and three for the translation), considered the impossibility to recover the scale factor, a minimum of five correspondences is needed.

3.2. 1-point algorithm

In this subsection we propose a novel method to estimate the relative motion between two consecutive camera views, which only requires the observation of a single feature in the scene and the knowledge of the angular rates from an inertial measurement unit, under the assumption that the local camera motion lies in a plane perpendicular to the gravity vector. Using this 1-point motion parametrization, we provide two very efficient algorithms to remove the outliers of the feature-matching process. Thanks to their inherent efficiency, the proposed algorithms are very suitable for computationally-limited robots. We test the proposed approaches on both synthetic and real data, using video footage from a small flying quadrotor. We show that our methods outperform standard RANSAC-based implementations by up to two orders of magnitude in speed, while being able to identify the majority of the inliers.

3.2.1. Parametrization of the camera motion

Considering that the camera is rigidly attached to the vehicle, two camera orientation angles are known (they correspond to the *Roll* and *Pitch* angles provided by the IMU).

If $R_x(\gamma)$, $R_y(\gamma)$, $R_z(\gamma)$ are the orthonormal rotation matrices for rotation of γ about the x-, y- and z-axes, the matrices

$$\begin{aligned} {}^{Cp1}R_{B_1} &= (R_x(Roll_1) \cdot R_y(Pitch_1))^T \\ {}^{Cp2}R_{B_2} &= (R_x(Roll_2) \cdot R_y(Pitch_2))^T \end{aligned} \tag{4}$$

allow us to virtually rotate the two camera frames into two new frames $\{C_{p_1}\}$ and $\{C_{p_2}\}$ (Figure 2). $Pitch_i$ and $Roll_i$, $(i = 1, 2)$ are the angles provided by

the IMU relative to two consecutive camera frames.

The two new image planes are parallel to the ground ($z_{C_{p_1}} \parallel z_{C_{p_2}} \parallel g$).

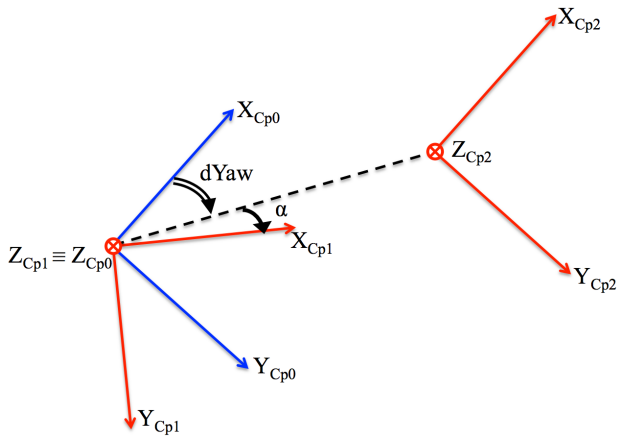


Figure 2: C_{p1} and C_{p2} are the reference frames attached to the vehicle's body frame but which z-axis is parallel to the gravity vector. They correspond to two consecutive camera views. C_{p0} corresponds to the reference frame C_{p1} rotated according to $dYaw$.

If the vehicle undergoes perfect planar motion, the essential matrix depends only on 2 parameters. Integrating the gyroscopic data within the time interval relative to two consecutive camera frames (i.e. the camera framerate), we can obtain the relative rotation of the two frames about Z_{C_p}-axis. We define a third reference frame C_{p_0}, that corresponds to the reference frame C_{p_1} rotated according to $dYaw$, in order to have the same orientation of C_{p2} (Figure 2)). The matrix that describes this rotation is the following:

$$^{Cp0}R_{Cp1} = R_z(dYaw)^T \tag{5}$$

To recap we can express the image coordinates into the new reference frames

according to:

$$\begin{aligned} \mathbf{p_{C_{p_0}}} &= {}^{Cp0}R_{Cp1} \cdot {}^{Cp1}R_{B_1} \cdot \mathbf{p_1} \\ \mathbf{p_{C_{p_2}}} &= {}^{Cp2}R_{B_2} \cdot \mathbf{p_2} \end{aligned} \tag{6}$$

At this point the transformation between $\{C_{p0}\}$ and $\{C_{p2}\}$ is a pure translation:

$$\begin{aligned} \mathbf{T} &= \rho[cos(\alpha) \quad -sin(\alpha) \quad 0]^T \\ \mathbf{R} &= I_3 \end{aligned} \tag{7}$$

and it depends only on α and on ρ (the scale factor). The essential matrix results therefore notably simplified:

$$E = [\mathbf{T}]_\times \mathbf{R} = \rho \begin{bmatrix} 0 & 0 & -sin(\alpha) \\ 0 & 0 & -cos(\alpha) \\ sin(\alpha) & cos(\alpha) & 0 \end{bmatrix} \tag{8}$$

At this point, being $\mathbf{p_{C_{p_0}}} = [x_0 \quad y_0 \quad z_0]^T$ and $\mathbf{p_{C_{p_2}}} = [x_2 \quad y_2 \quad z_2]^T$, we impose the epipolar constraint according to (2) and we obtain the homogeneous equation that must be satisfied by all the point correspondences.

$$(x_0 z_2 - z_0 x_2)sin(\alpha) + (y_0 z_2 - z_0 y_2)cos(\alpha) = 0 \tag{9}$$

where $\mathbf{p_0} = [x_0 \quad y_0 \quad z_0]^T$ and $\mathbf{p_2} = [x_2 \quad y_2 \quad z_2]^T$ are the directions (or unit-sphere coordinates) of a matched feature in $\{C_{p0}\}$ and $\{C_{p2}\}$ respectively. Equation 9 depends only on one parameter (α). This means that the relative vehicle motion can be estimated using only a single image feature correspondence.

At this point we can recover the angle α from 9:

$$\alpha = tan^{-1}\left(\frac{z_0 y_2 - y_0 z_2}{x_0 z_2 - z_0 x_2}\right) \tag{10}$$

3.2.2. 1-point RANSAC

One feature correspondence is randomly selected from the set of all the matched features. The motion hypothesis is computed according to (13). Without loss of generality we can set $\rho = 1$. Inliers are, by definition, the correspondences which satisfy the model hypothesis within a defined threshold. The

number of inliers in each iteration is computed using the reprojection error. We used an error threshold of 0.5 pixels. The minimum number of iterations to guarantee a good outlier detection, considering $p = 0.99$ and $\varepsilon = 0.5$ is 7 (according to (1)).

3.2.3. Me-RE (Median + Reprojection Error)

The angle α is computed from all the feature correspondences according to (10). A distribution $\{\alpha_i\}$ with $i = 1, 2, \ldots, N_f$ is obtained, where N_f is the number of correspondences between the two consecutive camera images.

The best angle α^* is computed as the median of the afore-mentioned distribution $\alpha^* = \text{median}\{\alpha_i\}$.

The inliers are then detected by using the reprojection error. Unlike the 1-point RANSAC, this algorithm is not iterative. Its computational complexity is linear in N_f.

3.3. 2-point algorithm

In this subsection we present a novel method to perform the outlier rejection task between two different views of a camera rigidly attached to an Inertial Measurement Unit (IMU). Only two feature correspondences and gyroscopic data from IMU measurerments are used to compute the motion hypothesis. By exploiting this 2-point motion parametrization, we propose two algorithms to remove wrong data associations in the feature-matching process for case of a 6DoF motion. We show that in the case of a monocular camera mounted on a quadrotor vehicle, motion priors from IMU can be used to discard wrong estimations in the framework of a 2-point-RANSAC based approach. The proposed methods are evaluated on both synthetic and real data.

3.3.1. Parametrization of the camera motion

Let us consider a camera rigidly attached to an Inertial Measurement Unit (IMU) consisting of three orthogonal accelerometers and three orthogonal gyroscopes. The transformation between the camera reference frame $\{C\}$ and the IMU frame $\{I\}$ can be computed using [46]. Without loss of generality, we can

therefore assume that these two frames are coincident ($\{I\} \equiv \{C\}$). The $\Delta\phi$, $\Delta\theta$ and $\Delta\psi$ angles characterizing the relative rotation between two consecutive camera frames can be calculated by integrating the high frequency gyroscopic measurements, provided by the IMU. This measurement is affected only by a slowly-changing drift term and can safely be recovered if the system is in motion.

If $R_x(\Delta)$, $R_y(\Delta)$, $R_z(\Delta)$ are the orthonormal rotation matrices for rotations of Δ about the x-, y- and z-axes, the matrix

$$^{C_0}R_{C_1} = (R_x(\Delta\phi) \cdot R_y(\Delta\theta) \cdot R_z(\Delta\psi))^T \tag{11}$$

allows us to virtually rotate the first camera frame $\{C_1\}$ into a new frame $\{C_0\}$ (Figure 1) having the same orientation of the second one $\{C_2\}$.

The matrix $^{C_0}R_{C_1}$ allows us to express the image coordinates relative to C_1 into the new reference frame C_0:

$$\mathbf{p_0} = ^{C_0} R_{C_1} \cdot \mathbf{p_1}. \tag{12}$$

At this point, the transformation between $\{C_0\}$ and $\{C_2\}$ is a pure translation

$$\begin{aligned} \mathbf{T} &= \rho[s(\beta) \cdot c(\alpha) \quad -s(\beta) \cdot s(\alpha) \quad c(\beta)]^T \\ \mathbf{R} &= I_3, \end{aligned} \tag{13}$$

which depends only on the angles α and β and on the scale factor ρ. The essential matrix results therefore simplified:

$$E = [\mathbf{T}]_\times \mathbf{R} = \rho \begin{bmatrix} 0 & -c(\beta) & -s(\beta) \cdot s(\alpha) \\ c(\beta) & 0 & -s(\beta) \cdot c(\alpha) \\ s(\beta) \cdot s(\alpha) & s(\beta) \cdot c(\alpha) & 0 \end{bmatrix}. \tag{14}$$

With $s(\cdot)$ and $c(\cdot)$ we denote the $\sin(\cdot)$ and $\cos(\cdot)$ respectively. At this point, being $\mathbf{p_0} = [x_0 \quad y_0 \quad z_0]^T$ and $\mathbf{p_2} = [x_2 \quad y_2 \quad z_2]^T$, the coordinates of a feature matched between two different camera frames and backprojected onto the unit sphere, we impose the epipolar constraint according to (2) and we obtain the homogeneous equation that must be satisfied by all the point correspondences.

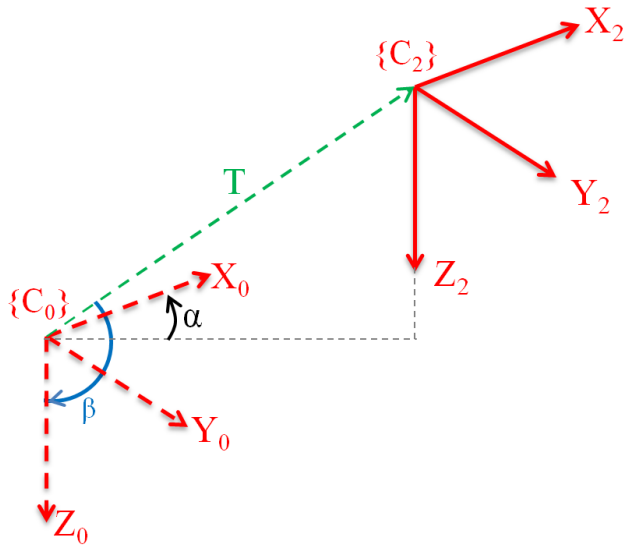


Figure 3: The reference frame C_0 and C_2 differ only for the translation vector T. $\rho = |T|$ and the angles α and β allow us to express the origin of the reference frame C_2 in the reference frame C_0.

$$\begin{aligned} x_2(y_0c(\beta) + z_0s(\alpha)s(\beta)) - y_2(x_0c(\beta) - z_0c(\alpha)s(\beta)) + \\ -z_2(y_0c(\alpha)s(\beta) + x_0s(\alpha)s(\beta)) = 0. \end{aligned} \tag{15}$$

Equation (15) depends on two parameters (α and β). This means that the relative vehicle motion can be estimated using only two image feature correspondences that we will identify as $\mathbf{p_A}$ and $\mathbf{p_B}$, where $\mathbf{p_{i_j}} = [x_{i_j} \quad y_{i_j} \quad z_{i_j}]^T$ with $i = A, B$ and $j = 0, 2$ indicate the direction of the feature i in the reference frame j.

At this point, we can recover the angles α and β solving (15) for the features $\mathbf{p_A}$ and $\mathbf{p_B}$:

$$\begin{aligned} \alpha &= -tan^{-1}\left(\tfrac{c_4c_2 - c_1c_5}{c_4c_3 - c_1c_6}\right), \\ \beta &= -tan^{-1}\left(\tfrac{c_1}{c_2c(\alpha) + c_3s(\alpha)}\right), \end{aligned} \tag{16}$$

where

$$
\begin{aligned}
c_1 &= x_{A_2} y_{A_0} - x_{A_0} y_{A_2}, \\
c_2 &= -y_{A_0} z_{A_2} + y_{A_2} z_{A_0}, \\
c_3 &= -x_{A_0} z_{A_2} + x_{A_2} z_{A_0}, \\
c_4 &= x_{B_2} y_{B_0} - x_{B_0} y_{B_2}, \\
c_5 &= -y_{B_0} z_{B_2} + y_{B_2} z_{B_0}, \\
c_6 &= -x_{B_0} z_{B_2} + x_{B_2} z_{B_0}.
\end{aligned}
\tag{17}
$$

Finally, without loss of generality, we can set the scale factor ρ to 1 and estimate the essential matrix according to (14).

3.3.2. Hough

The angles α and β are computed according to (16) from all the feature pairs matched between two consecutive frames and distant from each other more than a defined threshold (see Section 5). A distribution $\{\alpha_i, \beta_i\}$ with $i = 1, 2, \ldots, N$ is obtained, where N is a function of the position of the features in the environment.

To estimate the best angles α^* and β^*, we build a Hough Space (Figure 4) which bins the values of $\{\alpha_i, \beta_i\}$ into a grid of equally spaced containers. Considering that the angle β is defined in the interval $[0, \pi]$ and that the angle α is defined in the interval $[0, 2\pi]$, we set 360 bins for the variable α and 180 bins for the variable β. The number of bins of the Hough Space encodes the resolution of the estimation.

The angles α^* and β^* are therefore computed as

$$< \alpha^*, \beta^* >= \text{argmax}\{H\},$$

where H is the Hough Space.

The factors that influence the distribution are the error on the estimation of the relative rotation, the image noise, and the percentage of outliers in the data. The closer we are to ideal conditions (no noise on the IMU measurements), the narrower will be the distribution. The wider is the distribution, the more uncertain is the motion estimate.

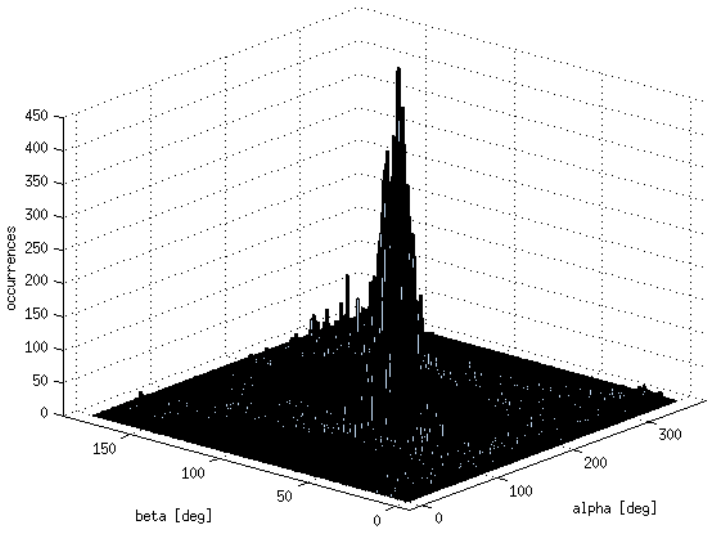


Figure 4: Hough Space in α and β computed with real data.

To detect the outliers, we calculate the reprojection error relative to the estimated motion model.

The camera motion estimation can be then refined processing the remaining subset of inliers with standard algorithms [14], [45].

3.3.3. 2-point RANSAC

Using (13) we compute the motion hypothesis that consists of the translation vector $\mathbf{T}$ and the rotation matrix $\mathbf{R} = \mathbf{I}_3$ by randomly selecting two features from the correspondence set. To have a good estimation, we check that the distance between the selected features is below a defined threshold (see Section 5). If it is not the case, we randomly select another pair of features. Constraints on the motion of the camera can be exploited to discard wrong estimations. The inliers are than computed using the reprojection error. The hypothesis that shows the highest consensus is considered to be the solution.

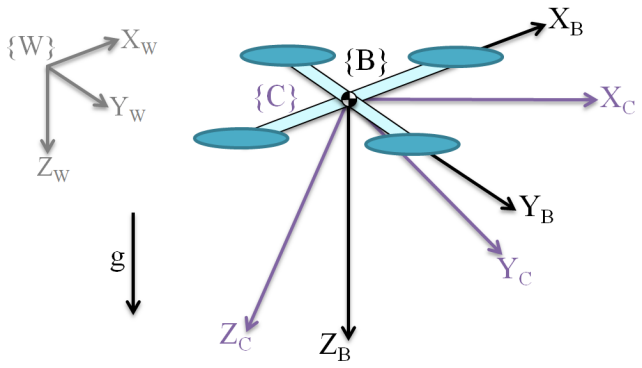


Figure 5: Notation. Vehicle's body frame$B$, camera frame C, world frame W, gravity vector g.

3.3.4. Quadrotor motion model

We consider a quadrotor equipped with a monocular camera and an IMU.

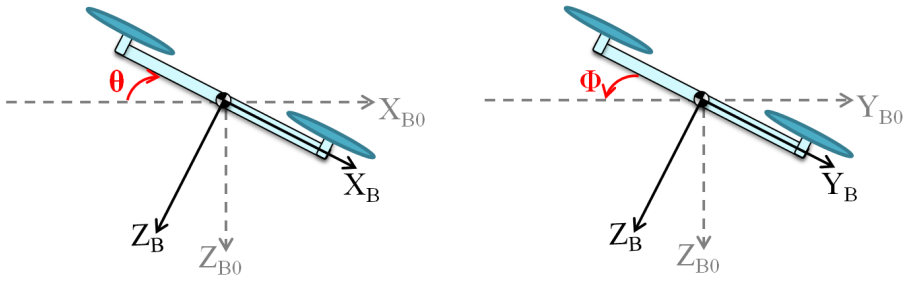
The vehicle body-fixed coordinate frame $\{B\}$ has its Z_B-axis pointing downward (following aerospace conventions [47]). The X_B-axis defines the forward direction and the Y_B-axis follows the right-hand rule.

Without loss of generality we can consider the IMU reference frame $\{I\}$ coinciding with the vehicle body frame $\{B\}$.

The modelization of the vehicle rotation in the World frame $\{W\}$ follows the $Z-Y-X$ Euler angles convention: being ϕ, θ, ψ respectively the $Roll$, $Pitch$ and Yaw angles of the vehicle, to go from the World frame to the Body frame, we first rotate about z_W axis by the angle ψ, then rotate about the intermediate y-axis by the angle θ, and finally rotate about the X_B-axis by the angle ϕ.

The transformation between the camera reference frame $\{C\}$ and the IMU frame $\{I\}$ can be computed using [46]. Without loss of generality, we can therefore assume that also these two frames are coincident ($\{I\} \equiv \{C\} \equiv \{B\}$).

A quadrotor has 6DoF, but its translational and angular velocity are strongly coupled to its attitude due to dynamic constraints. If we consider a coordinate frame $\{B_0\}$ with the origin coincident with the one of the vehicle's body frame



(a) (b)

Figure 6: Motion constraints on a quadrotor relative to its orientation. $\Delta\phi > 0$ implies a movement along Y_{B_0} positive direction, $\Delta\theta < 0$ implies a movement along Y_{B_0} positive direction.

$\{B\}$ and the X_{B_0} and Y_{B_0} axes parallel to the ground, we observe that, in order to move in the X_{B_0} direction, the vehicle must rotate about the y-axes axis (*Pitch* angle), while, in order to move in the Y_{B_0} direction, it must rotate about the x-axis (*Roll* angle) (Figure 6).

These motion constraints allow us to discard wrong estimations in a RANSAC based outlier detection approach. By looking at the relation between the x and y component of the estimated translation vector and the $\Delta\phi$, $\Delta\theta$ angles provided by the IMU measurements (the same used in (11)), we are able to check the consistency of the motion hypothesis. If the estimated motion satisfies the condition

$$\begin{aligned} &((|\Delta\phi| > \epsilon)\&(\Delta\phi \cdot T_y > 0)) \quad \| \\ &((|\Delta\theta| > \epsilon)\&(\Delta\theta \cdot T_x < 0)) \quad \| \\ &((|\Delta\phi| < \epsilon)\&(|\Delta\theta| < \epsilon)), \end{aligned} \tag{18}$$

we count the number of inliers (the number of correspondences that satisfy the motion hypothesis according to a predefined threshold) by using the reprojection error, otherwise we select another feature pair. The condition in (18) is satisfied if the x and y components of the motion hypothesis are coherent with the orientation of the vehicle. If both the angles $\Delta\phi$ and $\Delta\theta$ are below the threshold

ϵ, we cannot infer nothing about the motion and we proceed in the evaluation of the model hypothesis using the reprojection error.

The value of the threshold ϵ is a function of the vehicle dynamics and of the controller used.

Using (1) and considering $p = 0.99$ and $\varepsilon = 0.5$, we calculate the minimum number of iterations necessary to guarantee a good performance to our algorithm and we set it to 16.

4. Pose estimation

In this section we propose a new approach to perform MAV localization by only using the data provided by an Inertial Measurement Unit (IMU) and a monocular camera. The goal of our investigation is to find a new pose estimator which minimizes the computational complexity. We focus our attention on the problem of relative localization, which makes possible the accomplishment of many important tasks (e.g. hovering, autonomous take off and landing). In this sense, we minimize the number of point features which are necessary to perform localization. While 2 point features is the minimum number which provides full observability, by adding an additional feature, the precision is significantly improved, provided that the so-called planar ground assumption is honoured. This assumption has recently been exploited on visual odometry with a bundle adjustment based method [48]. The proposed method does not use any known pattern but only relies on three natural point features belonging to the same horizontal plane. The first step of the approach provides a first estimate of the roll and pitch (through the IMU data) and then the vehicle heading by only using two of the three point features and a single camera image. In particular, the heading is defined as the angle between the MAV and the segment made by the two considered point features. Then, the same procedure is repeated two additional times, i.e., by using the other two pairs of the three point features. In this way, three different heading angles are evaluated. On the other hand, these heading angles must satisfy two geometrical constraints, which are

fixed by the angles characterizing the triangle made by the three point features. These angles are estimated in parallel by an independent Kalman Filter. The information contained in the geometrical constraints is then exploited by minimizing a suitable cost function. This minimization provides a new and very precise estimate of the roll and pitch and consequently of the yaw and the robot position. Note that the minimization of the cost function does not suffer from an erroneous initialization since a first estimate of the roll and pitch is provided by the IMU.

4.1. The System

Let us consider an aerial vehicle equipped with a monocular camera and IMU sensors. We assume that the transformation among the camera frame and the IMU frame is known (we can assume that the vehicle frame coincides with the camera frame).

We assume that three reliable point-features are detected on the ground (i.e. they belong to the same horizontal plane). As we will see, two is the minimum number of features necessary to perform localization. Figure 7 shows our global frame G, which is defined by only using two features, P_1 and P_2. First, we define P_1 as the origin of the frame. The z_G-axis coincides with the gravity vector but with opposite direction. Finally, P_2 defines the x_G-axis [1].

Then, by applying the method in [33], the distance between these point features can be roughly determined by only using visual and inertial data (specifically, at least three consecutive images containing these points must be acquired).

[1] Note that the planar assumption is not necessary to define a global frame. It is sufficient that P_1 and P_2 do not lie on the same vertical axis (defined by the gravity). The X_G-axis can be defined assuming that P_2 belongs to the $x_G - z_G$-plane. In other words, P_2 has zero y_G coordinate.

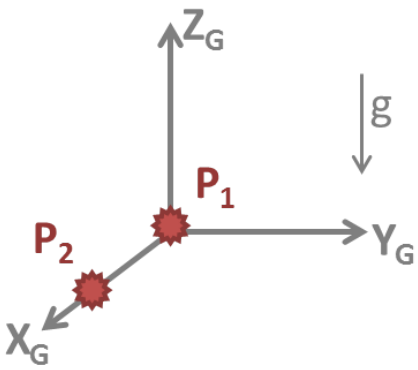


Figure 7: Global frame. Two is the minimum number of point features which allows us to uniquely define a global reference frame. P_1 is the origin, the x_G-axis is parallel to the gravity and P_2 defines the x_G-axis

4.2. The method

The first step of the method consists in estimating the Roll and the Pitch angles. This is performed by an Extended Kalman Filter (EKF) which estimates the gravity in the local frame by only using inertial data. In particular, in this EKF the prediction is done by using the data from the gyroscopes, while the perception by using the data from the accelerometers. Note that the accelerometers alone cannot distinguish the gravity from the inertial acceleration. In particular to distinguish the gravity from the inertial acceleration it is necessary to also use vision (see for instance [33]). However, in the case of micro aerial vehicles, the inertial acceleration is much smaller than the gravity. Additionally, since we know that the speed is bounded, we know that the inertial acceleration, averaged on a long time interval, is almost zero and can be considered as a noise in this EKF. Note also that for micro aerial vehicles this is exactly what has always been done to estimate the roll and pitch. Finally, in our approach, the roll and pitch estimated by this EKF are only a first estimate which is then improved by using also the camera measurements.

Once the direction of the gravity vector in the local frame is estimated, the

Roll and Pitch angles are obtained.

The second step returns the yaw angle and the position of the vehicle taking as input the Roll and Pitch angles and a single camera image.This is obtained by running the *3p-algorithm* (sec. 4.2.2). This algorithm starts by running three times the *2p-algorithm*, which is described in sec. 4.2.1.

4.2.1. 2p-Algorithm

This algorithm only uses two point features. Figure 8 shows the algorithm's inputs and outputs.

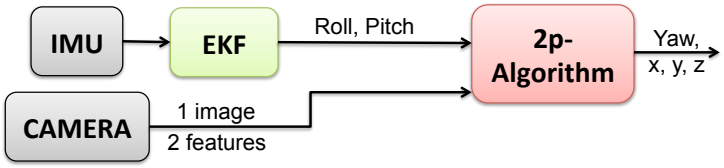


Figure 8: The 2p-algorithm.

For each feature, the camera provides its position in the local frame up to a scale factor. The knowledge of the absolute Roll and Pitch, allows us to express the position of the features in a new vehicle frame N, which Z_N-axis is parallel to the gravity vector. Figure 9 displays all the reference frames: the global frame G, the vehicle frame (represented by V) and the new vehicle frame N. Our goal is to determine the coordinates of the origin of the vehicle frame in the global frame and the orientation of the X_N-axis with respect to the x_G-axis (which corresponds to the Yaw angle of the vehicle in the global frame).

Let us denote with $[x_1, y_1, z_1]^T$ and $[x_2, y_2, z_2]^T$ the coordinates of P_1 and P_2 in the new local frame. The camera provides $\mu_1 = \frac{x_1}{z_1}$, $\nu_1 = \frac{y_1}{z_1}$, $\mu_2 = \frac{x_2}{z_2}$ and $\nu_2 = \frac{y_2}{z_2}$. Additionally, the camera also provides the sign of z_1 and ${z_2}$[2].

Since the Z_N-axis has the same orientation as the z_G-axis, and the two features P_1 and P_2 belongs to a plane perpendicular to the gravity vector, $z_1 =$

[2]For a camera with a field of view smaller than $180deg$ the $z-$component is always positive in the original camera frame.

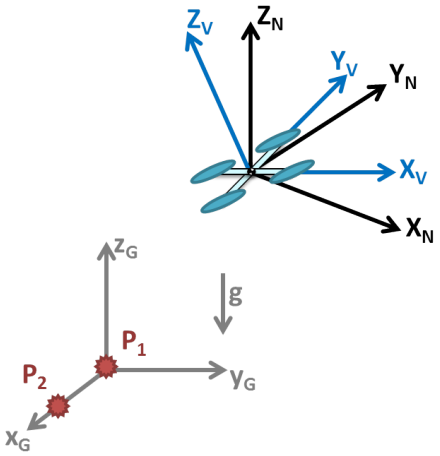


Figure 9: The three reference frames adopted in our derivation.

$z_2 = -z$, where z is the position of the origin of the vehicle frame in the global frame. We obtain:

$$P_1 = -z \begin{bmatrix} \mu_1 \\ \nu_1 \\ 1 \end{bmatrix} \qquad P_2 = -z \begin{bmatrix} \mu_2 \\ \nu_2 \\ 1 \end{bmatrix} \tag{19}$$

Let us denote by D the distance between P_1 and P_2. We have:

$$z = \pm \frac{D}{\sqrt{\Delta\mu_{12}^2 + \Delta\nu_{12}^2}} \tag{20}$$

with $\Delta\mu_{12} \equiv \mu_2 - \mu_1$ and $\Delta\nu_{12} \equiv \nu_2 - \nu_1$. In other words, z can be easily obtained in terms of D. The previous equation provides z up to a sign. This ambiguity is solved considering that the camera provides the sign of z_1 and z_2. Then, we obtain

$$x_1 = -z\mu_1 \quad y_1 = -z\nu_1 \quad x_2 = -z\mu_2 \quad y_2 = -z\nu_2 \tag{21}$$

It is therefore easy to obtain $\alpha = \arctan 2(\Delta\nu_{12},\ \Delta\mu_{12})$ (Figure 10). Hence,

$$Yaw = -\alpha = -\text{atan}(\Delta\nu_{12}/\ \Delta\mu_{12}) \tag{22}$$

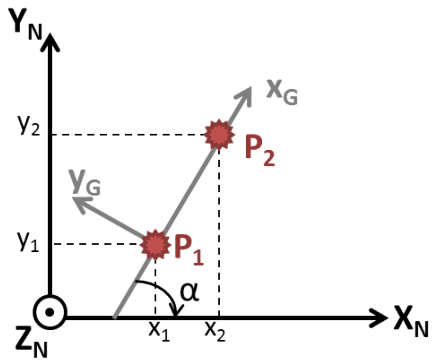


Figure 10: The yaw angle $(-\alpha)$ is the orientation of the X_N-axis in the global frame.

Finally we obtain the coordinates of the origin of the vehicle frame in the global frame,

$$\begin{aligned} x &= -\cos(\alpha)\ x_1 - \sin(\alpha)\ y_1 \\ y &= \sin(\alpha)\ x_1 - \cos(\alpha)\ y_1 \\ z &= \pm\frac{D}{\sqrt{\Delta\mu_{12}^2+\Delta\nu_{12}^2}} \end{aligned} \tag{23}$$

Note that the position x, y, z is obtained in function of the distance D. Specifically, according to equations 21 and 23, the position scales linearly with D. As previously said, a rough knowledge of this distance is provided by using the method in [33] and described in section 4.2.3. We remark that a precise knowledge of this distance is not required to accomplish tasks like hovering on a stable position.

4.2.2. 3p-Algorithm

The three features form a triangle in the (x_G, y_G)-plane. For the sake of clarity, we start our analysis supposing that we know the angles characterizing the triangle (γ_1 and γ_2 in Figure 11). Then, we will show how we estimate on line these angles (Section 4.2.4).

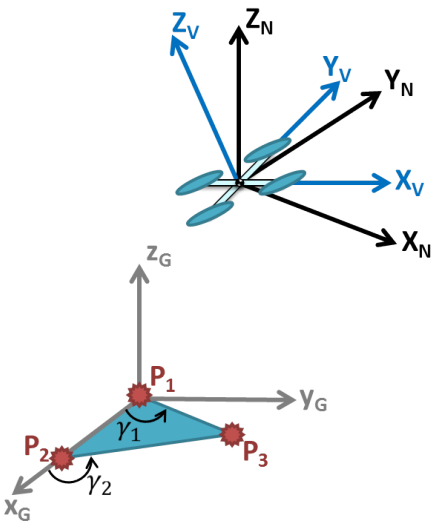


Figure 11: The triangle made by the 3 point features.

We run the $2p-$algorithm three times, respectively with the sets of features (P_1, P_2), (P_1, P_3) and (P_2, P_3) as input. We obtain three different angles. Yaw_{12} is the Yaw of the vehicle in the global frame given in (22) while the other expressions are:

$$\begin{aligned} Yaw_{12} &= -\text{atan}(\Delta\nu_{12}/\Delta\mu_{12}) \\ Yaw_{13} &= -\text{atan}(\Delta\nu_{13}/\Delta\mu_{13}) \\ Yaw_{23} &= -\text{atan}(\Delta\nu_{23}/\Delta\mu_{23}) \end{aligned} \tag{24}$$

The three above-mentioned angles must satisfy the following constraints:

$$\begin{aligned} \gamma_1 &= Yaw_{13} - Yaw_{12} \\ \gamma_2 &= Yaw_{23} - Yaw_{12} \end{aligned} \tag{25}$$

Note that the angles Yaw_{ij} are obtained by using equation 22. This equation uses $\Delta\nu_{ij}$ and $\Delta\mu_{ij}$ which are obtained by rotating the camera measurements according to the roll and pitch provided by the IMU. In other words, Yaw_{ij} can be considered a function of the roll and pitch.

Let us denote the known values of these angles with γ_1^0 and γ_2^0. We correct the estimation of the roll and pitch angles by exploiting these constraints. We solved a nonlinear least-squares problem minimizing the following cost function:

$$c(\zeta) = [(Yaw_{13} - Yaw_{12} - \gamma_1^0)^2 + (Yaw_{23} - Yaw_{12} - \gamma_2^0)^2] \tag{26}$$

in which the variables Yaw_{ij} are nonlinear functions of $\zeta = [Roll, Pitch]^T$.

Once the least-squares algorithm finds the Roll and Pitch angles that minimize the cost function, we can estimate the Yaw angle and the coordinates x, y and z as described in $2p-$algorithm (Figure 12).

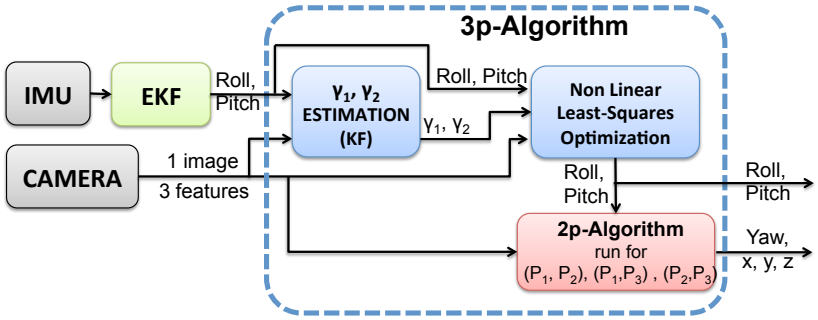


Figure 12: Flow chart of the proposed pose estimator

4.2.3. Scale factor initialization

Recent works on visual-inertial structure from motion have demonstrated its observability properties [29], [30], [31], [32], [33], [34], and [35]. It has been proved that the states that can be determined by fusing inertial and visual information are: the system velocity, the absolute scale, the gravity vector in the local frame, and the biases that affects the inertial measurements. The works in [33], [49] express all the observable modes at a given time T_{in} in closed-form

and only in function of the visual and inertial measurements registered during the time interval $[T_{in}, T_{fin}]$.

The position r of the system is:

$$\mathbf{r}(t) = \mathbf{r}(T_{in}) + \mathbf{v}(T_{in})\Delta t + \int_{T_{in}}^{t}\int_{T_{in}}^{\tau} \mathbf{a}(\xi)d\xi d\tau \tag{27}$$

where $t \in [T_{in},\ T_{fin}]$.

Integrating by part we obtain:

$$\boldsymbol{r}(t) = \boldsymbol{r}(T_{in}) + \boldsymbol{v}(T_{in})\Delta t + \int_{T_{in}}^{t} (t-\tau)\boldsymbol{a}(\tau)d\tau \tag{28}$$

where $\boldsymbol{v} \equiv \frac{d\boldsymbol{r}}{dt}$, $\boldsymbol{a} \equiv \frac{d\boldsymbol{v}}{dt}$ and $\Delta t \equiv t - T_{in}$.

The accelerometers provide the acceleration in the local frame and it also perceives the gravitational acceleration. The measurements are also corrupted by a constant term ($\mathbf{B}$) usually called bias. We can therefore write the accelerometer measurement like this:

$$\mathbf{A}_\tau(\tau) \equiv \mathbf{A}^i_\tau(\tau) - \mathbf{G}_\tau + \mathbf{B} \tag{29}$$

where $\mathbf{A}^i_\tau(\tau)$ is the inertial acceleration and $\mathbf{G}_\tau$ is the gravity acceleration in the local frame (depending on time because the local frame can rotate). Rewriting equation (28) by highlighting the vector $\boldsymbol{A}_\tau(\tau)$ provided by the accelerometer and neglecting the bias term $\mathbb{B}$:

$$\boldsymbol{r}(t) = \boldsymbol{r}(T_{in}) + \boldsymbol{v}(T_{in})\Delta t + \boldsymbol{g}\frac{\Delta t^2}{2} + C^{T_{in}}\left[\boldsymbol{S}_{T_{in}}(t)\right] \tag{30}$$

where:

$$\boldsymbol{S}_{T_{in}}(t) \equiv \int_{T_{in}}^{t} (t-\tau)C^{\tau}_{T_{in}}\boldsymbol{A}_\tau(\tau)d\tau;$$

The matrix $C^{\tau}_{T_{in}}$ can be obtained from the angular speed during the interval $[T_{in},\ \tau]$ provided by the gyroscopes [50].The vector $\boldsymbol{S}_{T_{in}}(t)$ can be obtained by integrating the data provided by the gyroscopes and the accelerometers delivered during the interval $[T_{in},\ t]$.

The visual measurements related to the observation of N point-features are recorded simultaneously with the inertial measurements. Let us denote the

feature position in the physical world with $\boldsymbol{p}^i,\ i = 1,\ ...,\ N$. $\boldsymbol{P}^i_t(t)$ denotes their position at time t in the local frame at time t. We have:

$$\boldsymbol{p}^i = \boldsymbol{r}(t) + C^{T_{in}} C^t_{T_{in}} \boldsymbol{P}^i_t(t) \tag{31}$$

Writing this equation for $t = T_{in}$ we obtain:

$$\boldsymbol{p}^i - \boldsymbol{r}(T_{in}) = C^{T_{in}} \boldsymbol{P}^i_{T_{in}}(T_{in}) \tag{32}$$

By inserting the expression of $\boldsymbol{r}(t)$ provided in (30) into equation (31), by using (32) and by pre multiplying by the rotation matrix $(C^{T_{in}})^{-1}$, we obtain:

$$C^t_{T_{in}} \boldsymbol{P}^i_t(t) = \boldsymbol{P}^i_{T_{in}}(T_{in}) - \boldsymbol{V}_{T_{in}}(T_{in})\Delta t - \boldsymbol{G}_{T_{in}} \frac{\Delta t^2}{2} - \boldsymbol{S}_{T_{in}}(t) \tag{33}$$
$$i = 1,\ 2,\ ...,\ N$$

A single image processed at time t, provides the position of the N features up to a scale factor, which correspond to the the vectors $\boldsymbol{P}^i_t(t)$. The data provided by the gyroscopes during the interval $(T_{in},\ T_{fin})$ allow us to build the matrix $C^t_{T_{in}}$. At this point, having the vectors $\boldsymbol{P}^i_t(t)$ up to a scale, allows us to also know the vectors $C^t_{T_{in}} \boldsymbol{P}^i_t(t)$ up to a scale.

We assume that the camera provides n_i images of the same N point-features at consecutive image stamps: $t_1 = T_{in} < t_2 < ... < t_{n_i} = T_{fin}$. For the sake of simplicity, we adopt the following notation:

- $\boldsymbol{P}^i_j \equiv C^{t_j}_{T_{in}} \boldsymbol{P}^i_{t_j}(t_j),\ i = 1,\ 2,\ ...,\ N;\ j = 1,\ 2,\ ...,\ n_i$
- $\boldsymbol{P}^i \equiv \boldsymbol{P}^i_{T_{in}}(T_{in}),\ i = 1,\ 2,\ ...,\ N$
- $\boldsymbol{V} \equiv \boldsymbol{V}_{T_{in}}(T_{in})$
- $\boldsymbol{G} \equiv \boldsymbol{G}_{T_{in}}$
- $\boldsymbol{S}_j \equiv \boldsymbol{S}_{T_{in}}(t_j),\ j = 1,\ 2,\ ...,\ n_i$

The vectors $\mathbb{P}_j^i$ can be written as $\mathbb{P}_j^i == \lambda_j^i \mu_j^i$. Without loss of generality we can set $T_{in} = 0$. Equation (33) can be written as follows:

$$\boldsymbol{P}^i - \boldsymbol{V} t_j - \boldsymbol{G}\frac{t_j^2}{2} - \lambda_j^i \boldsymbol{\mu}_j^i = \boldsymbol{S}_j \tag{34}$$

The corresponding linear system is:

$$\begin{cases} -\boldsymbol{G}\frac{t_j^2}{2} - \boldsymbol{V} t_j + \lambda_1^1 \boldsymbol{\mu}_1^1 - \lambda_j^1 \boldsymbol{\mu}_j^1 & = \boldsymbol{S}_j \\ \lambda_1^1 \boldsymbol{\mu}_1^1 - \lambda_j^1 \boldsymbol{\mu}_j^1 - \lambda_1^i \boldsymbol{\mu}_1^i + \lambda_j^i \boldsymbol{\mu}_j^i & = 0_3 \end{cases} \tag{35}$$

where $j = 2, ..., n_i$, $i = 2, ..., N$ and 0_3 is the 3×1 zero vector. This linear system consists of $3(n_i - 1)N$ equations in $Nn_i + 6$ unknowns. The two column vectors $\boldsymbol{X}$ and $\boldsymbol{S}$ and the matrix Ξ are defined as following:

$$\boldsymbol{X} \equiv [\boldsymbol{G}^T,\ \boldsymbol{V}^T,\ \lambda_1^1,\ ...,\ \lambda_1^N,\ ...,\ \lambda_{n_i}^1,\ ...,\ \lambda_{n_i}^N]^T$$

$$\boldsymbol{S} \equiv [\boldsymbol{S}_2^T,\ 0_3,\ ...,\ 0_3,\ \boldsymbol{S}_3^T,\ 0_3,\ ...,\ 0_3,\ ...,\ \boldsymbol{S}_{n_i}^T,\ 0_3,\ ...,\ 0_3]^T$$

$$\Xi \equiv \tag{36}$$

$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c|c} T_2 & S_2 & \boldsymbol{\mu}_1^1 & 0_3 & 0_3 & -\boldsymbol{\mu}_2^1 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_{33} & 0_{33} & \boldsymbol{\mu}_1^1 & -\boldsymbol{\mu}_1^2 & 0_3 & -\boldsymbol{\mu}_2^1 & \boldsymbol{\mu}_2^2 & 0_3 & 0_3 & 0_3 & 0_3 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ 0_{33} & 0_{33} & \boldsymbol{\mu}_1^1 & 0_3 & -\boldsymbol{\mu}_1^N & -\boldsymbol{\mu}_2^1 & 0_3 & \boldsymbol{\mu}_2^N & 0_3 & 0_3 & 0_3 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ T_{n_i} & S_{n_i} & \boldsymbol{\mu}_1^1 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & -\boldsymbol{\mu}_{n_i}^1 & 0_3 & 0_3 \\ 0_{33} & 0_{33} & \boldsymbol{\mu}_1^1 & -\boldsymbol{\mu}_1^2 & 0_3 & 0_3 & 0_3 & 0_3 & -\boldsymbol{\mu}_{n_i}^1 & \boldsymbol{\mu}_{n_i}^2 & 0_3 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ 0_{33} & 0_{33} & \boldsymbol{\mu}_1^1 & 0_3 & -\boldsymbol{\mu}_1^N & 0_3 & 0_3 & 0_3 & -\boldsymbol{\mu}_{n_i}^1 & 0_3 & \boldsymbol{\mu}_{n_i}^N \end{array}\right]$$

where $T_j \equiv -\frac{t_j^2}{2} I_3$, $S_j \equiv -t_j I_3$ and I_3 is the identity 3×3 matrix; 0_{33} is the 3×3 zero matrix. The linear system in (35) can be written in a compact

format:

$$\Xi \boldsymbol{X} = \boldsymbol{S} \tag{37}$$

The linear system in 37 contains completely the sensor information. By adding the following equation to the system:

$$|\Pi \boldsymbol{X}|^2 = g^2 \tag{38}$$

where $\Pi \equiv [I_3,\ 0_3\ ...\ 0_3]$, it is possible to exploit the information related to the fact that the magnitude of the gravitational acceleration is known.

The Visual-Inertial Structure from Motion problem consists in the determination of the vectors: $\boldsymbol{P}^i$, $(i = 1,\ 2,\ ...,\ N)$, $\boldsymbol{V}$, $\boldsymbol{G}$ and it can be solved by finding the vector $\boldsymbol{X}$, which satisfies (37) and (38). The scale factors are the quantities λ_j^i for $i = 1,\ 2,\ ...,\ N$, $j = 1,\ 2,\ ...,\ n_i$ contained in the state vector $\mathbb{X}$.

In our case to initialize the scale factor we need at least three consecutive images containing the two points P_1 and P_2. This is enough considering that we know the gravity magnitude and that we know in advance we will not occur in degenerative cases (none of the camera poses will be aligned along with the two features, and the three camera poses and the two features will not belong to the same plane) [49].

4.2.4. Estimation of γ_1 and γ_2

In order to estimate the angles characterizing the triangle γ_1 and γ_2 (Figure 11), we run a Kalman filter. The state that we want to estimate is $\Gamma = [\gamma_1, \gamma_2]^T$. During the prediction step the filter does not update neither the state Γ nor its covariance matrix because the angles are constant in time. For the observation step we need the estimated Roll and Pitch (which allow us to virtually rotate the vehicle frame V into the the new frame N) and the observations of the three features in the current camera image $[x_i, y_i, z]^T = z[\mu_i, \nu_i, 1]^T$ for $i = 1, 2, 3$. At this point the sides of the triangle can be computed according to: $a = z\sqrt{\Delta\mu_{12}^2 + \Delta\nu_{12}^2}$, $b = z\sqrt{\Delta\mu_{13}^2 + \Delta\nu_{13}^2}$, $c = z\sqrt{\Delta\mu_{23}^2 + \Delta\nu_{23}^2}$.

Applying the law of cosine we can easily compute the two required angles:

$$\gamma_1 = \text{acos}\left(\frac{a^2+b^2-c^2}{2ab}\right)$$
$$\gamma_2 = \pi - \text{acos}\left(\frac{a^2+c^2-b^2}{2ac}\right)$$

Note that these angles are independent from z. γ_1 and γ_2 represent the observation of the state Γ of the Kalman Filter.

5. Performance evaluation

5.1. Outlier detection

To evaluate the performance of our algorithms, we run Monte Carlo simulations and experiments on real data. We compared the proposed approaches with the 5-point RANSAC [13] on synthetic data, and with the 5-point RANSAC [13] and the 8-point RANSAC [51] on real data.

Experiments on synthetic data. We simulated different trajectories of a quadrotor moving in indoor scenarios (Figure 13 and 18). The simulations have been performed using the *Robotics and Machine Vision Toolbox* for Matlab [47].

To make our simulations as close as possible to the experiments, we simulated a quadrotor vehicle moving in indoor environment, equipped with a downlooking monocular camera. We randomly generated 1600 features on the ground plane (Figure 13). Note that no assumptions are made on the feature's depth.

We simulated a perspective camera with the same parameters of the one we used for the experiments and added a Gaussian noise with zero mean and standard deviation of 0.5 pixels to each image point. To evaluate the performance of the 1-point algorithm, we generated a circular trajectory (easily repeatable in our flying arena) with a diameter of $1.5m$ (Figure 13). The vehicle was flying at the fix height of $2m$ above the ground. The period for one rotation is $10s$. The camera framerate is $15Hz$, its resolution is 752 x 480. For the 1-point RANSAC and the Me-RE, we set a threshold of 0.5 pixels. For the 5-point RANSAC we set a minimum number of trials of 145 iterations, and a threshold of 0.5 pixels as well.

In Figure 14 we present the results obtained along the aforementioned trajectory in the case of perfect planar motion (the helicopter is flying always at the same height above the ground, and the Roll and Pitch angles are not affected by noise).

Figure 15 represents the results when the *Roll* and *Pitch* angles are affected by a Gaussian Noise with standard deviation of 0.3 degrees.

We evaluated as well the case in which the measure of the angle $dYaw$ is affected by a Gaussian Noise with standard deviation of 0.3 degrees. The relative results are shown in Figure 16

We finally evaluated the case of non perfect planar motion introducing a sinusoidal noise (frequency 4 rad/s and with amplitude of $0.02m$) on the z_W-component of motion of the vehicle. Figure 17 represents the relative results.

We can observe that the *Median + Reprojection Error* (Me-RE) performs always better than the 1-point RANSAC, and requires no iterations (its computational complexity is linear in the number of features).

In the case of perfect planar motion (Figure 14), the Me-RE algorithm finds more inliers than the 5-point RANSAC. The latter algorithm requires at least 145 iterations according to Table 1 to insure a good performance.

When the variables *Roll*, *Pitch* and $dYaw$ are affected by errors (Figures 15 and 16), the performance of our algorithms drops, but they can still find almost the 50% of inliers.

As expected, if the vehicle's motion is not perfectly planar (Figure 17), the performances of the 1-point RANSAC and the Me-RE get worse. The oscillations that we can see in the plots are related to the fact that when the vehicle is approaching the ground, less features are in the field of view of its onboard camera.

To evaluate the performance of the 2-point algorithm, we generated a trajectory consisting of a take-off and of a constant-height maneuver (Figure 18).

Figure 19 shows the results of a simulation run along the trajectory depicted in Figure 18, in the ideal case of no noisy IMU measurements. The helicopter takes off and performs a constant height maneuver.

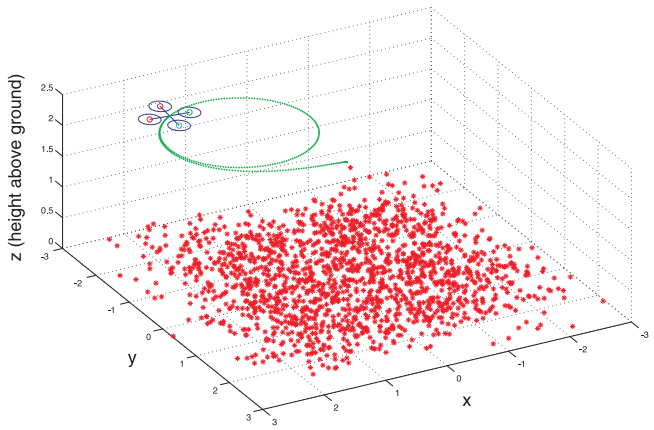


Figure 13: Synthetic scenario for the evaluation of the 1-point algorithm.

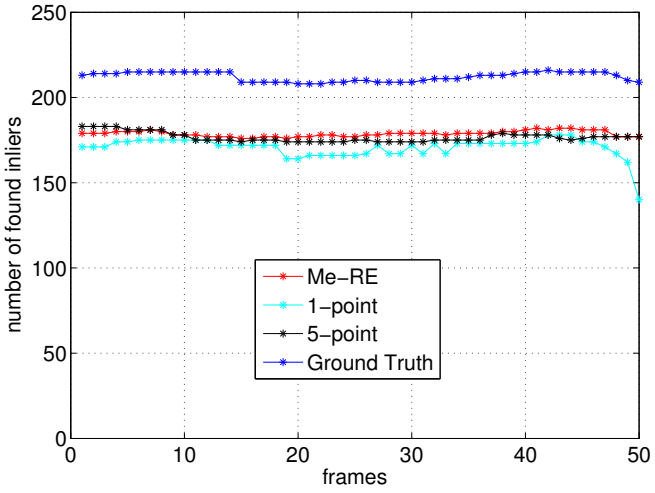


Figure 14: Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) for a perfect planar motion.

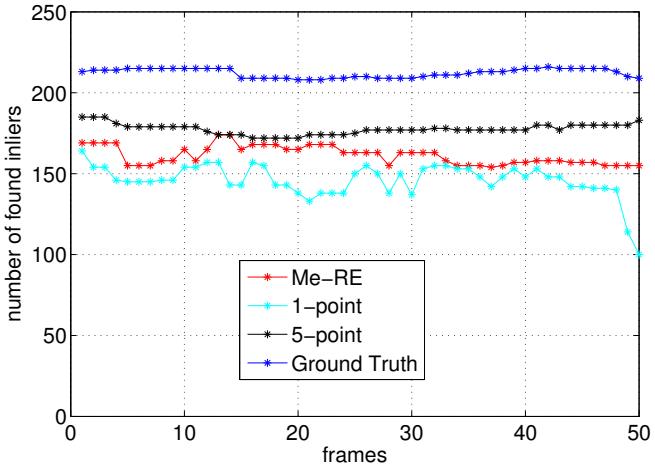


Figure 15: Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) in presence of perturbations on the *Roll* and *Pitch* angles.

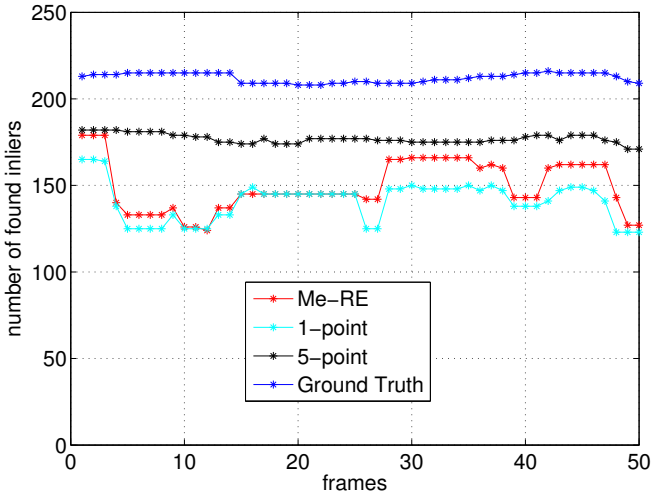


Figure 16: Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) in presence of perturbations on the $dYaw$ angle.

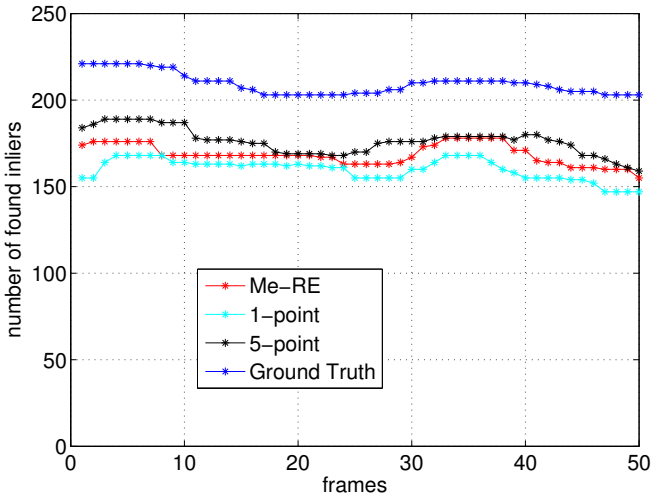


Figure 17: Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) for a non-perfect planar motion ($s_1 = 0.02 * sin(8 * w_c \cdot t)$).

In Figure 20, we present the results related to simulations where the quantities $\Delta\phi$, $\Delta\theta$ and $\Delta\psi$ are affected by a Gaussian noise with standard deviation of 0.3 degrees. Those errors do not affect the performance of the 5-point algorithm (that does not use IMU readings to compute the motion hypothesis). In this case, the Hough and the 2-point RANSAC approaches can still detect more than half of the inliers. The motion hypothesis can then be computed on the obtained set of correspondences by using standard approaches [14], [45].

In Figure 21, we present the results related to simulations where the quantities $\Delta\phi$ and $\Delta\theta$ are affected by a Gaussian noise with standard deviation of 0.3 degrees and in Figure 22 only the angle $\Delta\psi$ is affected by a Gaussian noise with standard deviation of 0.3 degrees. These two plots show that errors on rotations about the camera optical axis (that in our case coincides with rotations about the vehicle Z_B axis, i.e. errors on $\Delta\psi$) affects more the performances of both the algorithms than errors on $\Delta\phi$ and $\Delta\theta$.

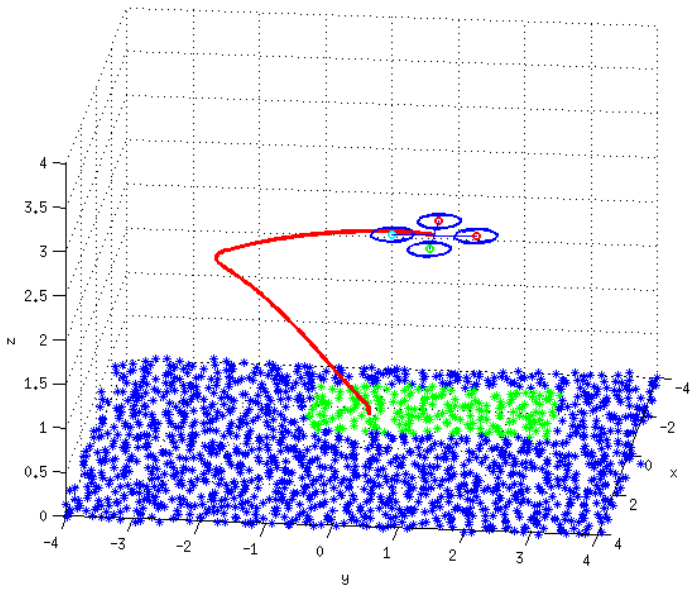


Figure 18: Synthetic scenario for the evaluation of the 2-point algorithm. The red line represents the trajectory and the blue dots represents the simulated features. The green dots are the features in the current camera view.

Experiments on real data. We tested our method on a nano quadrotor (Figure 23)[3] equipped with a MicroStrain 3DM-GX3 IMU (250 Hz) and a Matrix Vision mvBlueFOX-MLC200w camera (FOV: 112 *deg*).

The monocular camera has been calibrated using the *Camera Calibration Toolbox for Matlab* [52]. The extrinsic calibration between the IMU and the camera has been performed using the *Inertial Measurement Unit and Camera Calibration Toolbox* [46]. The dataset was recorded in our flying arena and ground truth data have been recorded using an Optitrack motion capture system with sub-millimeter accuracy.

The trajectories have been generated using the TeleKyb Framework [53]

[3] http://KMelRobotics.com

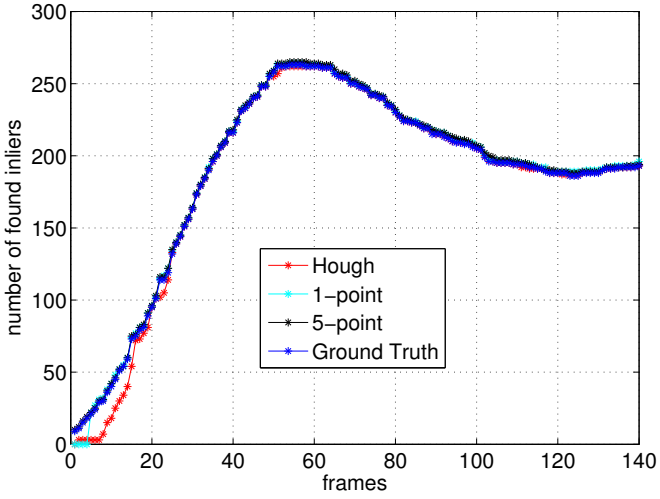


Figure 19: The IMU measurements are not affected by noise (ideal conditions).

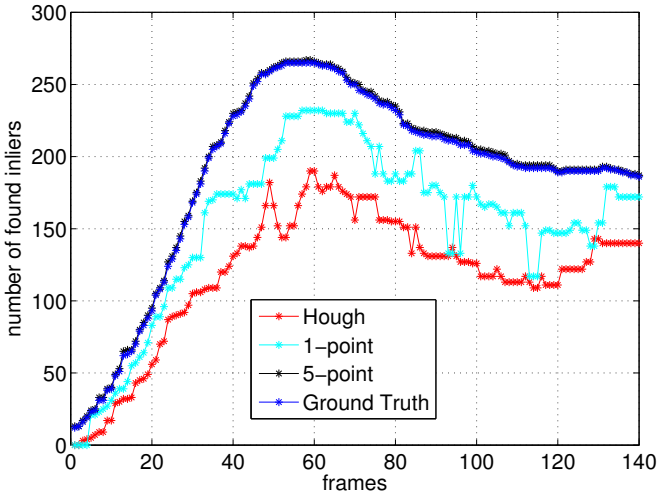


Figure 20: The angles $\Delta\phi$, $\Delta\theta$ and $\Delta\psi$ are affected by noise.

(Figure 24 and 28). The trajectory generated in order to evaluate the perfor-
590 mance of the 1-point algorithm is a circular trajectory ($1.5m$ of diameter, period

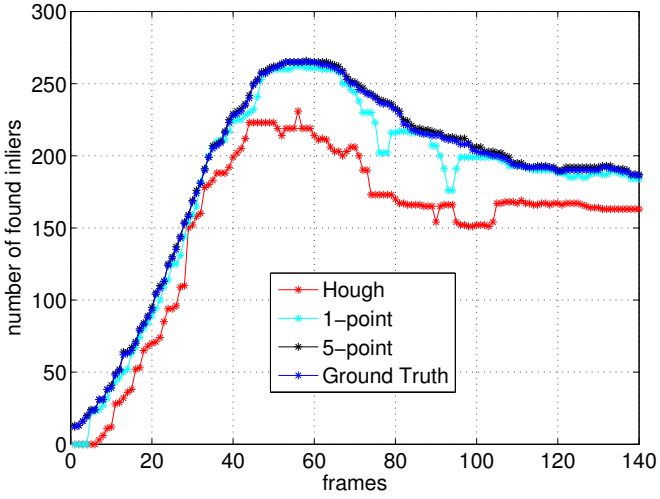


Figure 21: Only the angles $\Delta\phi$ and $\Delta\theta$ are affected by noise.

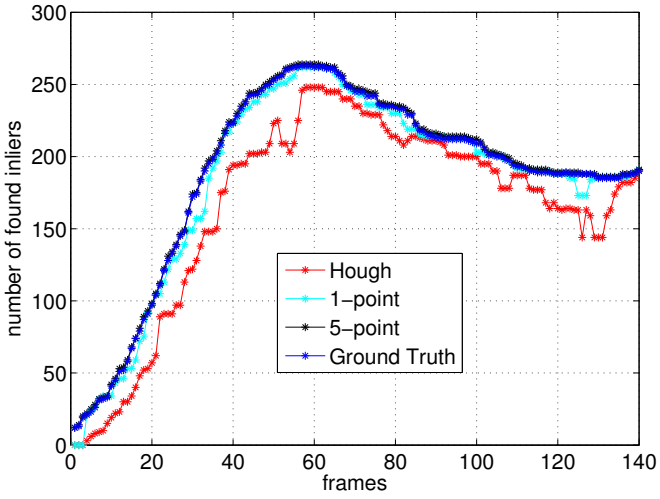


Figure 22: Only the angle $\Delta\psi$ is affected by noise.

of $10s$) with fixed height above the ground of $1.5m$. We computed SURF features (Speeded Up Robust Feature). The feature detection and matching tasks

Figure 23: Our nano quadrotor from KMelRobotics: a 150g and 18cm sized platform equipped with an integrated Gumstix Overo board and MatrixVision VGA camera.

has been performed using the *Machine Vision Toolbox* from [47].

To evaluate the performance of our methods, we compared the number of inliers found by the proposed methods with the number of inliers found by the 5-point RANSAC and the 8-point RANSAC methods.

Figure 25 presents the result of this comparison for the case of the 1-point algorithm. We observe that in the interval [380 : 490] the Me-RE algorithm has a very good performance (it finds even more inliers than the 5-points RANSAC). On the contrary the performance drops in the intervals [350 : 380] and [490 : 540]. The last plot in Figure 26 shows the height of the vehicle above the ground during the trajectory. We can notice that in the interval [380 : 490] the motion of the vehicle along the z-World axis is smoother than in the other intervals, therefore it affects less the performance of the 1-point and of the Me-RE methods.

Table 2 shows the computation time of the compared algorithms, implemented in Matlab and run on an *Intel Core i7-3740QM Processor*. According to our experiments, the 5-point RANSAC takes about 67 times longer than the

8-point. The reason of this is that for each candidate point set, the 5-point RANSAC returns up to ten motion solutions and this involves both Singular Value Decomposition (SVD) and Groebner-basis decompositions. Instead, the 8-point RANSAC only returns 1 solution and has only one SVD, no Groebner-basis decomposition.

The Me-RE algorithm is not considered as a complete alternative to the 5-point RANSAC. However, thanks to its negligible computation time (Table 2), it can be run at each frame. If the resulting number of inliers will be below a defined threshold, it will be more suitable to switch to the 5-point algorithm.

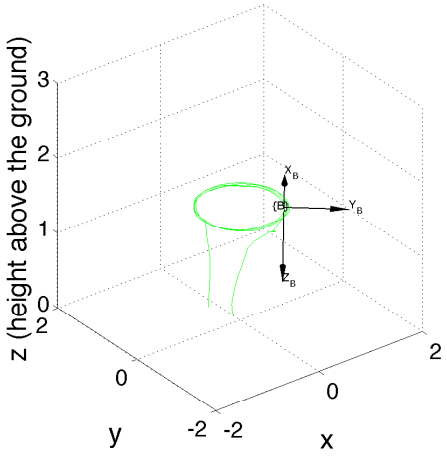


Figure 24: Plot of the real trajectory. The vehicle's body frame is depicted in black and the green line is the trajectory followed.

To evaluate the performance of the 2-point algorithm, we realized a trajectory consisting of a take-off and a constant-height maneuver above the ground, as shown in Figure 28 by using the TeleKyb Framework [53]. We recorded a dataset composed of camera images, IMU measurements and ground truth data provided by the Optitrack.

We processed our dataset with SURF features, matching them in consecutive camera frames. We run the 8-point RANSAC method on each correspondences

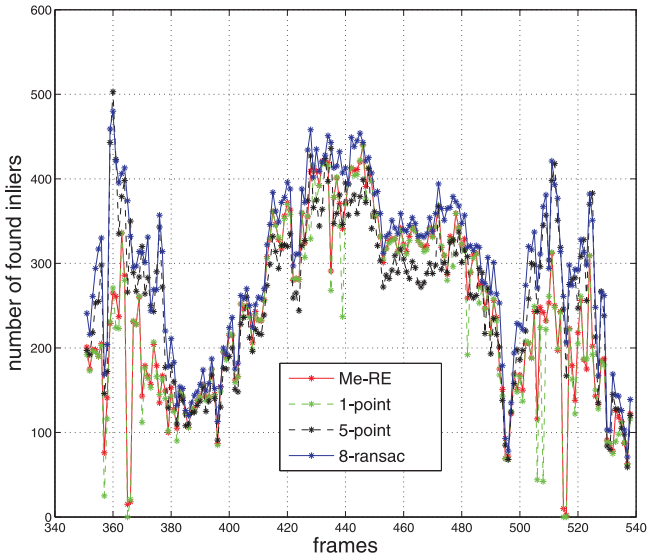


Figure 25: Number of found inliers by Me-RE (red), 1-point RANSAC (green), 5-point RANSAC (black), 8-point RANSAC (blue) along the trajectory depicted in Figure 24.

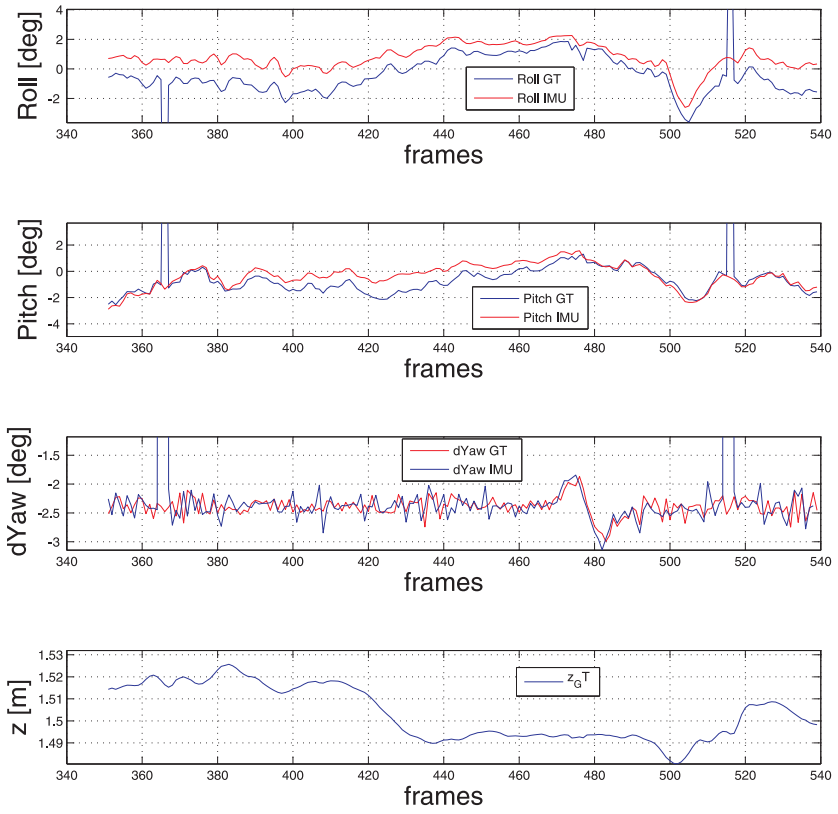


Figure 26: From the top to the bottom: *Roll*, *Pitch* and $dYaw$ angles [deg] estimated with the IMU (red) versus *Roll*, *Pitch* and $dYaw$ angles [deg] estimated with the Optitrack system (blue). The last plot shows the height of the vehicle above the ground (non perfect planarity of motion).

Algorithm	Me-Re	1-point	5-points	8-points
Time [s]	0.0028	0.0190	2.6869	0.0396

Figure 27: Table 2: Computation time.

set to have an additional term of comparison.

To evaluate the performance of our methods, we compared the number of inliers detected using the Hough and the 2-point RANSAC methods with 5-point and an 8-point RANSAC. For the 2-point RANSAC we set $\epsilon = 0.1$ deg. The results of this comparison are shown in Figure 29.

Figure 30 shows the error characterizing the estimated relative rotation between two consecutive camera frames obtained by IMU measurements and the ground truth values.

Looking at both Figure 29 and Figure 30, we can notice that the smaller are the errors on the angles estimations, the higher is the number of inliers detected by the Hough and the 2-point RANSAC method.

Our algorithms and the algorithms that we used for the comparison, are implemented in Matlab and run on an *Intel Core i7-3740QM Processor*. We summarize their computation time in Table 3. We can notice that the computation time of the 5-point RANSAC is almost 67 times the computation time of the 8-point RANSAC. This is due to the fact that the 5-points returns up to 10 motion solutions for each candidate set. Singular Value Decomposition (SVD) and Groebner-basis decompositions are involved and this explains the high computation time.

The computation time of the Hough algorithm is function of the number of feature pairs used to compute the distribution in Figure (4). In our experiments, we choose all the feature pairs distant more than a defined threshold one to each other. We experimentally set this threshold to 30 degrees on the unit sphere.

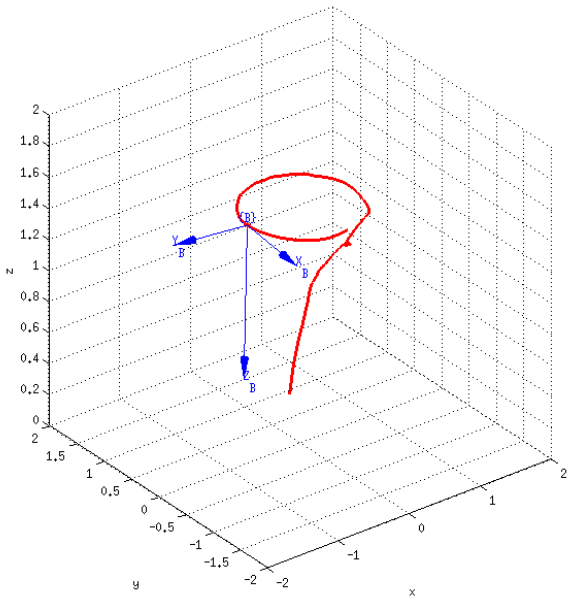


Figure 28: Real scenario. The vehicle body frame is represented in blue, while the red line represents the followed trajectory.

5.2. Pose estimation

Experiments on synthetic data. In order to evaluate the performance of the presented method, we simulated different $3D$ trajectories and scenarios.

The considered scenarios to test the 2p-Algorithm is shown in Figure 7. The features are $P1 = [0, 0, 0]$, $P2 = D * [1, 0, 0]$, where $D = 0.1m$. To compare the 2p-Algorithm with the 3p-Algorithm, we added a third feature $P3 = D * [0.5, \sqrt{3}/2, 0]$ (Figure 11). The angles γ_1 and γ_2 are respectively $60deg$ and $120deg$.

The trajectories are generated with a quadrotor simulator that, given the initial conditions, the desired position and desired Yaw, performs a hovering task [54]. The initial vehicle position is $x = y = z = 0\ m$, the initial vehicle speed is $v_x = v_y = v_z = 0\ ms^{-1}$ in the global frame.

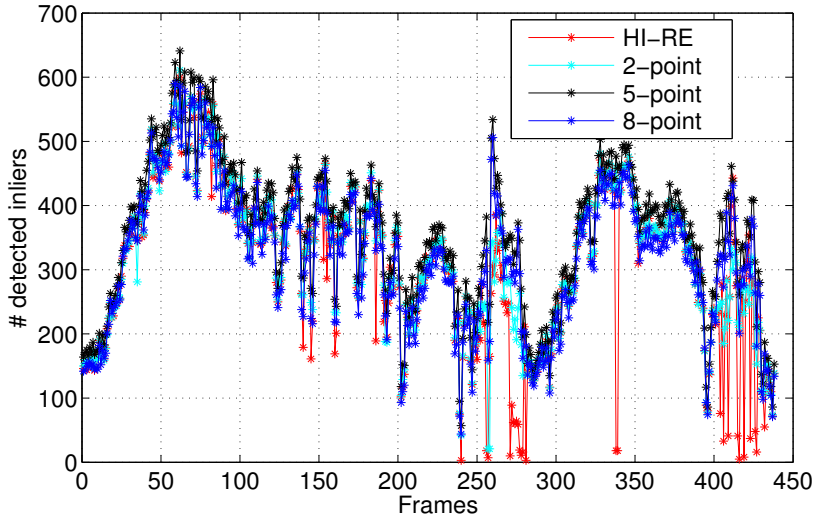


Figure 29: Number of inliers detected with the Hough approach (red), the 2-point RANSAC (cyan), the 5-point RANSAC (black) and the 8-point RANSAC (blue) along the trajectory depicted in Figure 28.

Algorithm	Hough	2-points	5-points	8-points
Time [s]	0.498	0.048	2.6869	0.0396

Table 3: Computation time.

Starting from the performed trajectory, the true angular speed and the linear acceleration are computed each $0.01s$ We denote with $\boldsymbol{\Omega_i^{true}}$ and $\boldsymbol{A_{v\ i}^{true}}$ the true value of the body rates and linear accelerations at time stamp i. The IMU readings are generated as following: $\boldsymbol{\Omega_i} = N\left(\boldsymbol{\Omega_i^{true}} - \boldsymbol{\Omega_{bias}}, P_{\Omega_i}\right)$ and $\boldsymbol{A_i} = N\left(\boldsymbol{A_{v\ i}^{true}} - \boldsymbol{A_g} - \boldsymbol{A_{bias}}, P_{A_i}\right)$ where:

- N indicates the Normal distribution whose first entry is the mean value and the second one is the covariance matrix;
- P_{Ω_i} and P_{A_i} are the covariance matrices characterizing the accuracy of the IMU;

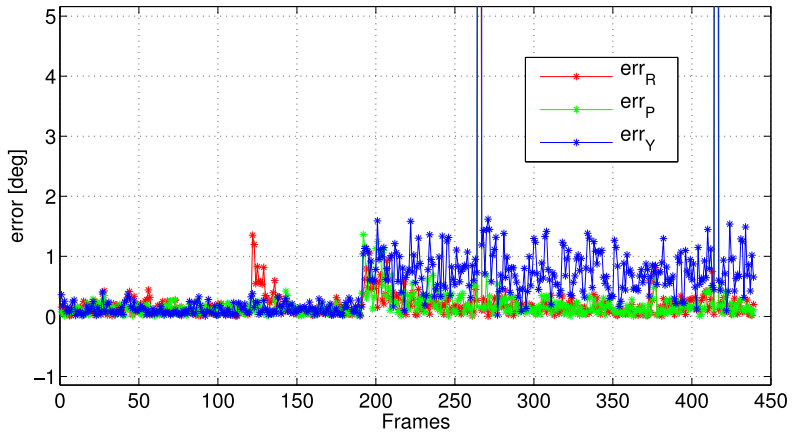


Figure 30: Errors between the relative rotations $\Delta\phi$ (err_R), $\Delta\theta$ (err_P), $\Delta\psi$ (err_Y) estimated with the IMU and estimated with the Optitrack.

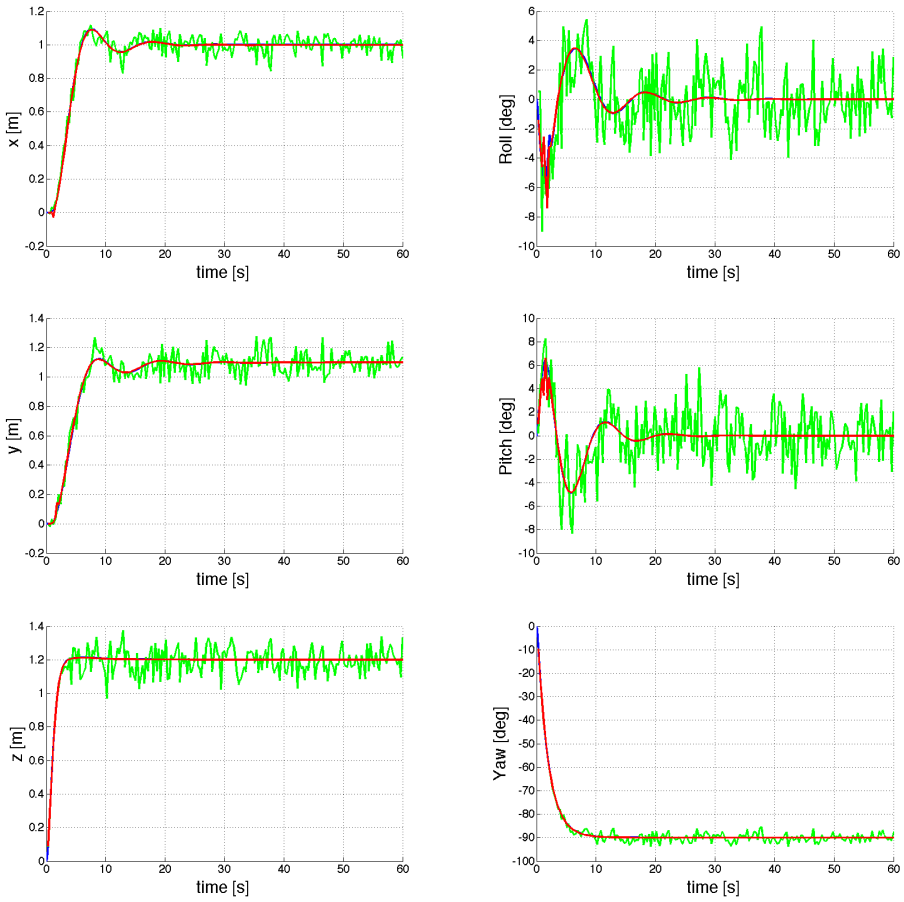
- $\boldsymbol{A_g}$ is the gravitational acceleration in the local frame and $\boldsymbol{A_{bias}}$ is the bias affecting the accelerometer's data;

- $\boldsymbol{\Omega_{bias}}$ is the bias affecting the gyroscope's data.

In all the simulations we set both the matrices P_{Ω_i} and P_{A_i} diagonal and in particular: $P_{\Omega_i} = \sigma^2_{gyro} I_3$ and $P_{A_i} = \sigma^2_{acc} I_3$, where I_3 is the identity 3×3 matrix. We considered several values for σ_{gyro} and σ_{acc}, in particular: $\sigma_{gyro} = 1\ deg\ s^{-1}$ and $\sigma_{acc} = 0.01\ ms^{-2}$.

The camera is simulated as follows. Knowing the true trajectory of the vehicle, and the position of the features in the global frame, the true bearing angles of the features in the camera frame are computed each $0.3s$. Then, the camera readings are generated by adding zero-mean Gaussian errors (whose variance is set to $(1\ deg)^2$) to the true values.

Figures 31.a show the results regarding the estimated x, y and z. Figures 31.b show the results regarding the estimated $Roll$, $Pitch$ and Yaw. In each figure we represent the ground truth values in blue, the values estimated with the 2p-Algorithm in green and the values estimated with the 3p-algorithm in



a *b*

Figure 31: Estimated x, y, z (a), and $Roll$, $Pitch$, Yaw (b). The blue line indicate the ground truth, the green one the estimation with the 2p-Algorithm and the red one the estimation with the 3p-Algorithm

	x	y	z	Roll	Pitch	Yaw
3p-Algorithm	0.26 %	0.24 %	0.08 %	0.07 deg	0.04 deg	0.01 deg
2p-Algorithm	4.08 %	5.41 %	5.23 %	1.63 deg	1.72 deg	1.36 deg

Table 4: Mean error on the estimated states in our simulations. For the position the error is given in %.

Figure 32: AscTec Pelican quadcopter [55] equipped with a monocular camera.

red.

Table 4 summarizes these results by providing the mean error on the estimated position and attitude.

Experiments on real data. This section describes our experimental results. The robot platform is a *Pelican* from *Ascending Technologies* [55] equipped with an Intel Atom processor board (*1.6 GHz, 1 GB RAM*) (Figure 32).

Our sensor suite consists of an Inertial Measurement Unit (*3-Axis Gyro, 3-Axis Accelerometer*) belonging to the Flight Control Unit (FCU) AscTec Autopilot , and a monocular camera (*Matrix Vision mvBlueFOX*, FOV: 130 *deg*). The camera is calibrated using the Camera Calibration Toolbox for Matlab [52]. The calibration between the IMU and the camera has been performed using the Inertial Measurement Unit and Camera Calibration Toolbox in [46]. The IMU provides measurements update at a rate of $100Hz$, while the camera framerate

is $10Hz$.

The Low Level Processor (LLP) of our Pelican is flashed with the *2012 LLP Firmware* [55] and performs attitude data fusion and attitude control. We flashed the High Level Processor (HLP) with the asctec_hl_firmware [56]. The onboard computer runs linux 10.04 and ROS (Robot Operating System). We implemented our method using ROS as a middleware for communication and monitoring . The HLP communicates with the onboard computer through a FCU-ROS node. The communication between the camera and the onboard computer is achieved by a ROS node as well. The presented algorithms are running online and onboard at $10Hz$.

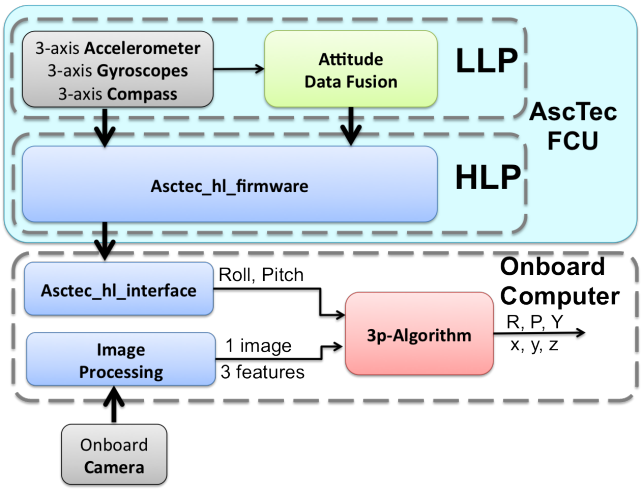
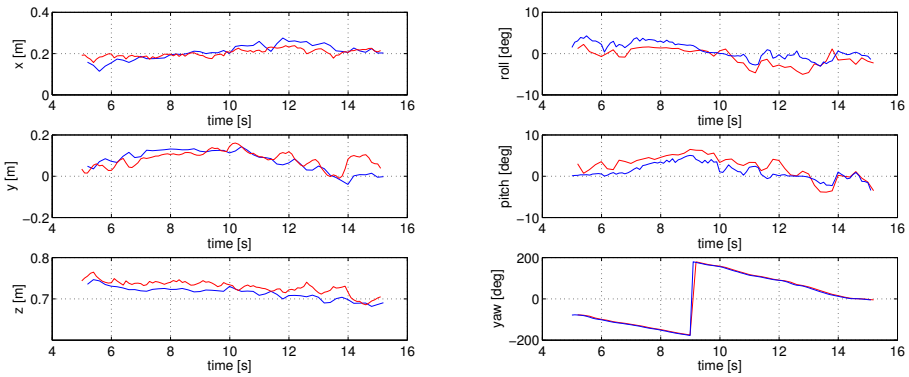


Figure 33: Our Pelican quadcopter: a system overview

A motion capture system is used to evaluate the performance of our approach. Note that the estimations of the camera pose provided by the motion capture system is not used to perform the estimation. Three reflective markers are positioned according to Figure 11. The three features considered are the center of the three reflective markers. The use of three blob markers instead of natural features is only related to the need to get a ground truth. The information related to the pattern composed by the 3 features ($D = 0.25m$, $\gamma_1 = 60deg$,

$\gamma 2 = 120deg$) is only used to evaluate the performance of our approach. The algorithm does not require any information about the features configuration.

Figures 34.a and 34.b show respectively the position and the attitude estimated by using the proposed approach and compared with the ground truth obtained with the motion capture system. From Figure 34.a we see that the difference between our estimates and the ground truth values is of the order of $2cm$ for x and y and less than $1cm$ for z. From Figure 34.b we see that the difference between our estimates and the ground truth values is of the order of $2deg$ for $Roll$, $Pitch$ and Yaw.



a b

Figure 34: Estimated position (a), respectively x, y, z and estimated attitude (b), respectively $Roll$, $Pitch$, Yaw. The red lines represent the estimated values with the 3p-Algorithm, the blue ones represent the ground truth values.

6. Conclusions

This paper provides two main contributions. The former is the presentation of two methods to perform outlier detection on computationally-constrained micro aerial vehicles. The algorithms rely on onboard IMU measurements to calculate the relative rotation between two consecutive camera frames and the

reprojection error to detect the inliers. The first method assumes that the vehicle's motion is locally planar, while the second method generalizes to unconstrained (i.e., 6DoF) motion. Although the 5-point RANSAC is the "golden standard method" for 6DoF motion estimation, experimental results show that the proposed Me-RE and 2-point RANSAC algorithms can be used as a first choice before committing to the 5-point RANSAC due to their very low computational complexity. Considering that the Me-RE algorithm relies on the local planar motion assumption, we remark that it can replace the 5-point algorithm when the motion of the vehicle is smooth and the camera framerate is high. The motion can then be refined applying standard methods [14], [45] to the remaining inliers. Considering that the parameter $\alpha*$ is estimated as the median of the distribution of the α computed from all the feature correspondences (10), the standard deviation of this distribution can be considered as an measure of reliability of the Me-RE algorithm. We show that in the case of a monocular camera mounted on a quadrotor vehicle, motion priors from IMU can be used to discard wrong estimations in the framework of a 2-point-RANSAC–based approach.

The latter contribution is a new approach to perform MAV localization by only using the data provided by an Inertial Measurement Unit and a monocular camera. The approach exploits the so-called planar ground assumption and only needs three natural point features. It is based on a simple algorithm, which provides the vehicle pose from a single camera image, once the roll and the pitch angles are obtained by the inertial measurements. The very low computational cost of the proposed approach makes it suitable for pose control in tasks, such as hovering, and autonomous take-off and landing.

References

[1] S. Weiss., D. Scaramuzza, R. Siegwart, Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments, Journal of Field Robotics 28 (6).

[2] M. Achtelik, S. Lynen, S. Weiss, L. Kneip, M. Chli, R. Siegwart, Visual-inertial slam for a small helicopter in large outdoor environments, in: Video Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.

[3] D. Scaramuzza, M. Achtelik, L. Doitsidis, F. Fraundorfer, E. Kosmatopoulos, A. Martinelli, M. Achtelik, Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. Lee, S. Lynen, L. Meier, M. Pollefeys, A. Renzaglia, R. Siegwart, J. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments, ieee robotics and automation magazine. 21 (3).

[4] D. Scaramuzza, F. Fraundorfer, Visual odometry: Part I - the first 30 years and fundamentals, Robotics and Automation Magazine, IEEE 18 (4) (2011) 80–92.

[5] F. Fraundorfer, D. Scaramuzza, Visual odometry: Part II - matching, robustness, optimization, and applications, Robotics and Automation Magazine, IEEE 19 (2) (2011) 80–92.

[6] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[7] C. Troiani, S. Al Zanati, A. Martinelli, A 3 points vision based approach for mav localization in gps denied environments, in: European Conference on Mobile Robotics (ECMR), 2013.

[8] C. Troiani, A. Martinelli, C. Laugier, D. Scaramuzza, 1-point-based monocular motion estimation for computationally-limited micro aerial vehicles, in: European Conference on Mobile Robotics (ECMR), 2013.

[9] E. Kruppa, Zur ermittlung eines objektes aus zwei perspektiven mit inner

orientierung, in: Sitz. –Ber. Akad. Wiss, Wien, Math. Naturw. Kl., Abt. IIa., Vol. 122, 1913, pp. 1939–1948.

[10] O. D. Faugeras, S. Maybank, Motion from point matches: multiplicity of solutions, International Journal of Computer Vision 4 (3) (1990) 225–246.

[11] J. Philip, A non-iterative algorithm for determining all essential matrices corresponding to five point pairs, The Photogrammetric Record 15 (88) (1996) 589–599.

[12] B. Triggs, Routines for relative pose of two calibrated cameras from 5 points.

[13] D. Nistér, An efficient solution to the five-point relative pose problem, Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 (6) (2004) 756–770.

[14] H. Stewénius, C. Engels, D. Nistér, Recent developments on direct relative orientation, ISPRS Journal of Photogrammetry and Remote Sensing 60 (4) (2006) 284–294.

[15] O. Pizarro, R. Eustice, H. Singh, Relative pose estimation for instrumented, calibrated imaging platforms, in: DICTA, Citeseer, 2003, pp. 601–612.

[16] T. P. Fraundorfer F., M. Pollefeys, A minimal case solution to the calibrated relative pose problem for the case of two unknown orientation angles, in: European Conf. Computer Vision, 2010, pp. 269–282.

[17] O. Naroditsky, X. S. Zhou, J. Gallier, S. I. Roumeliotis, K. Daniilidis, Two efficient solutions for visual odometry using directional correspondence, Pattern Analysis and Machine Intelligence, IEEE Transactions on 34 (4) (2012) 818–824.

[18] L. Kneip, M. Chli, R. Siegwart, Robust real-time visual odometry with a single camera and an imu, in: Proc. of The British Machine Vision Conference (BMVC), Dundee, Scotland, 2011.

[19] L. Heng, G. H. Lee, F. Fraundorfer, M. Pollefeys, Real-time photo-realistic 3d mapping for micro aerial vehicles, in: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, 2011, pp. 4012–4019.

[20] D. Ortin, J. Montiel, Indoor robot motion based on monocular images, Robotica 19 (3) (2001) 331–342.

[21] D. Scaramuzza, F. Fraundorfer, R. Siegwart, Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC, in: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, 2009, pp. 4293–4299.

[22] D. Scaramuzza, 1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints, International journal of computer vision 95 (1) (2011) 74–85.

[23] D. Scaramuzza, Performance evaluation of 1-point-RANSAC visual odometry, Journal of Field Robotics 28 (5) (2011) 792–811.

[24] L. Armesto, J. Tornero, M. Vincze, Fast ego-motion estimation with multi-rate fusion of inertial and vision, Int. J. Rob. Res. 26 (6) (2007) 577–589.

[25] P. Gemeiner, P. Einramhof, M. Vincze, Simultaneous motion and structure estimation by fusion of inertial and vision data, Int. J. Rob. Res. 26 (6).

[26] M. Veth, J. Raquet, Fusion of low-cost imaging and inertial sensors for navigation, in: Journal of the Institute of Navigation, Vol. 54, 2007.

[27] J. Kim, S. Sukkarieh, Real-time implementation of airborne inertial-slam, Robot. Auton. Syst. 55 (1) 62–71.

[28] M. Bloesch, S. Weiss, D. Scaramuzza, R. Siegwart, Vision based mav navigation in unknown and unstructured environments, in: Proc. of The IEEE International Conference on Robotics and Automation (ICRA), 2010.

[29] E. Jones, A. Vedaldi, S. Soatto, Inertial structure from motion with autocalibration, in: Proceedings of the Internetional Conference on Computer Vision - Workshop on Dynamical Vision, 2007.

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

[31] J. Kelly, G. S. Sukhatme, Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration, in: CIRA'09, 2009, pp. 360–368.

[32] A. Martinelli, Closed-form solution for attitude and speed determination by fusing monocular vision and inertial sensor measurements., in: ICRA, IEEE, 2011.

[33] A. Martinelli, Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale and bias determination, Transaction on Robotics 28 (2012) 44–60.

[34] A. Martinelli, Observability properties and deterministic algorithms in visual-inertial structure from motion, Foundations and Trends in Robotics 3 (3) (2013) 139–209.

[35] A. I. Mourikis, S. I. Roumeliotis, A multi-state constraint Kalman filter for vision-aided inertial navigation, in: Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy, 2007, pp. 3565–3572.

[36] T. Zhang, Y. Kang, M. Achtelik, K. Kiihnlenz, M. Buss, Autonomous hovering of a vision/imu guided quadrotor, in: Proc. of International Conference on Mechatronics and Automation, Changchun, China, 2009.

[37] D. Eberli, D. Scaramuzza, S. Weiss, R. Siegwart, Vision based position control for mavs using one single artificial landmark, in: Proc. of International Conference and Exhibition on Unmanned Aerial Vehicles, Dubai, 2010.

[38] T. Cheviron, T. Hamel, R. Mahony, G. Baldwin, Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of uav, in: Proc. of International Conference on Robotics and Automation, Rome, Italy, 2007.

[39] M. Hwangbo, T. Kanade, Visual-inertial uav attitude estimation using urban scene regularities, in: Proc. of International Conference on Robotics and Automation, Shanghai, 2011.

[40] S. M. Weiss, Vision based navigation for micro helicopters (2012).

[41] S. Weiss, M. Achtelik, S. Lynen, M. Chli, R. Siegwart, Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments, in: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2012.

[42] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, in: Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 2007.

[43] S. Weiss, D. Scaramuzza, R. Siegwart, Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments, J. Field Robot. 28 (6) (2011) 854–874.

[44] A. Natraj, C. Demonceaux, P. Vasseur, P. F. Sturm, Vision based attitude and altitude estimation for uavs in dark environments., in: IROS, IEEE, 2011.

[45] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, Vol. 2, Cambridge Univ Press, 2000.

[46] J. Lobo, J. Dias, Relative pose calibration between visual and inertial sensors, The International Journal of Robotics Research 26 (6) (2007) 561–575.

[47] P. I. Corke, Robotics, Vision & Control: Fundamental Algorithms in Matlab, Springer, 2011.

[48] B. M. Kitt, J. Rehder, A. D. Chambers, M. Schonbein, H. Lategahn, S. Singh, Monocular visual odometry using a planar road model to solve scale ambiguity, in: Proc. of the European Conference on Mobile Robots, Orebro, Sweden, 2011.

[49] A. Martinelli, Closed-form solution of visual-inertial structure from motion, International Journal of Computer Vision 2 (106) (2014) 138–152.

[50] J. Farrell, Aided navigation: GPS with high rate sensors, McGraw-Hill New York, 2008.

[51] H. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, MA Fischler and O. Firschein, eds (1987) 61–62.

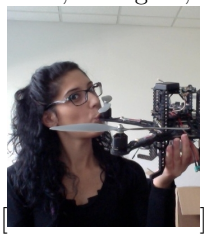
[52] J.-Y. Bouguet, Camera calibration toolbox for matlab.

[53] V. Grabe, M. Riedel, H. Bulthoff, P. R. Giordano, A. Franchi, The telekyb framework for a modular and extendible ros-based quadrotor control, in: submitted to ECMR, IEEE, 2013.

[54] P. Castillo, R. Lozano, A. E. Dzul, Modelling and Control of Mini-Flying Machines, Springer, 2005.

[55] Ascending technologies gmbh.
URL `http://www.asctec.de`

[56] M. W. Achtelik, M. Achtelik, S. Weiss, R. Siegwart, Onboard imu and monocular vision based control for mavs in unknown in- and outdoor environments, in: Proc. of International Conference on Robotics and Automation, Shanghai, 2011.



Chiara Troiani received the M.S. degree in automatic

and informatic engineering from University of LÁquila, Italy in 2009. She received the Ph.D. degree in Computer Science and Mathematics from University of Grenoble, and Inria (French institute for computer science), France in 2014. During her Ph.D. she spent eight months at the Robotics and Perception Group, University of Zurich, Switzerland.

[]Agostino Martinelli Agostino Martinelli received the M.Sc. degree in theoretical physics from the University of Rome Tor Vergata, Rome, Italy, in 1994 and the Ph.D. degree in astrophysics from the University of Rome La Sapienza, in 1999. While working toward the Ph.D. degree, he spent one year at the University of Wales, Cardiff, U.K., and one year with the Scuola Internazionale Superiore di Studi Avanzati, Trieste, Italy. His research focused on chemical and dynamical evolution in elliptical galaxies, in quasars, and in the intergalactic medium. He also introduced models based on general relativity to investigate the time evolution of the anisotropies of cosmic background radiation. After receiving the Ph.D. degree, his interests moved to the problem of autonomous navigation. He was with the University of Rome Tor Vergata for two years, and in 2002, he moved to the Autonomous Systems Laboratory, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, as a Senior Researcher,where he lead several projects on multisensor fusion for robot localization, simultaneous localization and odometry error learning, and simultaneous localization and mapping. Since September 2006, he has been a Researcher with the Institut National de Recherche en Informatique et en Automatique (INRIA), Rhone Alpes, Grenoble, France. His current research focuses on three main topics: Visual-Inertial Structure from Motion; Unknown Input non-linear observability; Overdamped Brownian motion and the Fokker-Plank equation without detailed balance. He is the author of more than 100 journal and conference papers.

]Christian Laugier is Research Director at Inria and Scientific Leader of the e-Motion Team. He is also responsible at INRIA for scientific relations with Asia & Oceania. He received the PhD degree in computer science from Grenoble University, France in 1976. His current research interests mainly lie in the areas of motion autonomy, intelligent vehicles and probabilistic robotics. He has co-edited several books in the field of robotics and several special issues of scientific journals such as IJRR, Advanced Robotics, JFR, or IEEE Transactions on ITS. In 1997, he was awarded the Nakamura Prize for his contributions to "Intelligent Robots and Systems". Dr. Laugier is a member of several scientific committees including the Steering/Advisory Committees of the IEEE/RSJ IROS, FSR, and ICARCV conferences. He is also co-Chair of the IEEE RAS Technical Committee on AGV & ITS. He has been General Chair or Program Chair of such conferences as: IEEE/RSJ IROS'97, IROS'02, IROS'08, or FSR'07. Additionally to his research and teaching activities, he co-founded four start-up companies in the fields of robotics, computer vision, computer graphics, and Bayesian programming tools. He has served as scientific consultant for the ITMI, Aleph Technologies, and ProBayes companies.

]Davide Scaramuzza Davide Scaramuzza (1980, Italian) is Assistant Professor of Robotics at the University of Zurich. He is founder and director of the Robotics and Perception Group, where he develops cutting-edge research on low-latency vision and visually-guided micro aerial vehicles. He received his PhD (2008) in Robotics and Computer Vision at ETH Zurich. He

was Postdoc at both ETH Zurich and the University of Pennsylvania. From 2009 to 2012, he led the European project sFly, which introduced the worlds first autonomous navigation of micro quadrotors in GPS-denied environments using vision as the main sensor modality. For his research contributions, he was awarded the IEEE Robotics and Automation Early Career Award (2014), a Google Research Award (2014), the European Young Researcher Award (2012), and the Robotdalen Scientific Award (2009). He is coauthor of the 2nd edition of the book Introduction to Autonomous Mobile Robots (MIT Press). Finally, he is author of several top-ranked robotics and computer vision journals. His research interests are field and service robotics, intelligent vehicles, and computer vision. Specifically, he investigates the use of cameras as the main sensors for robot navigation, mapping, exploration, reasoning, and interpretation. His interests encompass both ground and flying vehicles.