HAL
archives-ouvertes.fr

1-Point-based Monocular Motion Estimation for Computationally-Limited Micro Aerial Vehicles

Chiara Troiani, Agostino Martinelli, Christian Laugier, Davide Scaramuzza

1-Point-based Monocular Motion Estimation for Computationally-Limited Micro Aerial Vehicles

Chiara Troiani[1], Agostino Martinelli[1], Christian Laugier[1] and Davide Scaramuzza[2]

TABLE I: Number of iterations of RANSAC

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

Abstract— **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.**

I. INTRODUCTION

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], 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. The resulting matching points are usually contaminated by outliers, i.e., wrong data associations. To have a robust estimation of the camera motion, it is necessary to remove the outliers since they can affect negatively the accuracy of the estimated motion. Outliers rejection consists of exploiting the geometric constraints induced by the motion model and is computationally very expensive.

The standard method for model estimation from a set of data affected by outliers is RANSAC (RANdom SAmple Consensus) [5]. 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 [4]:

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

[1]C. Troiani, A. Martinelli, and C. Laugier are with INRIA Rhone Alpes, Montbonnot, France (`chiara.troiani, agostino.martinelli, christian.laugier) @inria.fr`
[2]D. Scaramuzza is with the Artificial Intelligence Lab—Robotics and Perception Group, Department of Informatics, University of Zurich, Switzerland `davide.scaramuzza@ieee.org`

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 I 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.

Fig. 1: Our nano quadrotor from KMelRobotics: a 150g and 18cm sized platform equipped with an integrated Gumstix Overo board and MatrixVision VGA camera.

In this paper, we show that for the specific case of a camera equipped with an IMU, under the assumption that the camera motion lies in a plane perpendicular to the direction of gravity, we only need a single scene point observation to determine the relative motion. Using this 1-point motion parametrization, we provide two very efficient algorithms to remove the outliers of the feature matching process, which outperform standard RANSAC-based implementations by up to two orders of magnitude in speed.

Due to the extremely-low computational complexity, the proposed algorithm is very suited for ultra-lightweight micro flying platforms, such as our nano quadrotor equipped with a computationally-limited embedded computer (an integrated Gumstix Over board, see Figure 1). Note that the proposed algorithm is not considered to be a replacement of the 5-point algorithm (that represents the state of the art in terms of real time applications), but mostly a way to save time and CPU resources whenever the motion of the vehicle allows a

suitable 1-point parametrization.

The paper is organized as follows. Section II reviews the related works. Section III deals with the parametrization of the camera motion. In Section IV, we describe the proposed methods for outlier detection. An evaluation of the performance is presented in Section V and the conclusions in Section VI.

II. RELATED WORKS

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 [6]. Several five-point minimal solvers were proposed later in [7],[8],[9], but an efficient implementation, based on [8], was found only in 2003 by Nister [10] and later revised in [11]. Before that, the six- [12], seven- or eight- solvers were commonly used. However, the five-point solver has the advantage that it works also for planar scenes.

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 [13], 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 [14] improved on [13] 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 VO was proposed in [15], which uses the full rotation matrix from an IMU rigidly attached to the camera. 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 [16]. For wheeled vehicles, the work in [17], [18] 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. A performance evaluation of five-, two-, and one-point RANSAC algorithms for VO was finally presented in [19].

III. PARAMETRIZATION OF THE CAMERA MOTION

We consider a micro aerial vehicle equipped with a monocular camera and an IMU. The transformation between the camera reference frame $\{C\}$ and the vehicle's body frame $\{B\}$ (that for aerial vehicles is coincident with the IMU frame) can be computed using [20]. Without loss of generality, we can assume that these two frames are coincident.

According to aerospace conventions [21], the X_B-axis of an aerial vehicle commonly defines the forward direction, the Z_B-axis is downward, and the Y_B-axis follows the right-hand rule. We assume the same convention for our vehicle (Figure 2). We use the $Z - Y - X$ Euler angles to model the rotation of the vehicle in the World frame. To go from the World frame to the Body frame, we first rotate about z_W axis by the angle Yaw, then rotate about the intermediate y-axis by the angle $Pitch$, and finally rotate about the X_B-axis by the angle $Roll$.
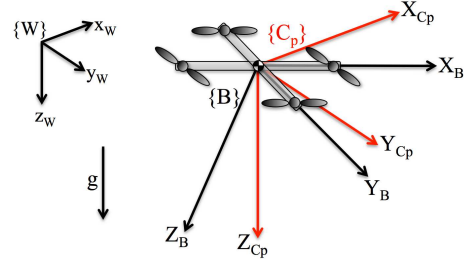

Fig. 2: Notation.

We define as well a coordinate frame $\{C_p\}$ attached to the vehicle, with the same origin as the vehicle's Body Frame but with its z-axis aligned to the gravity vector (**g**). The $Roll$ and $Pitch$ angles and the relative rotation about Z_{C_p}-axis ($dYaw$) of the vehicle are provided by the IMU fusing the integration of the high frequency gyroscopic measurements with the gravity direction obtained by the accelerometers. If the system is in motion, the resulting estimation allows us to safely recover the short term relative orientation of the vehicle, that is only affected by a slowly changing drift term.

A. Computation of the Essential Matrix

Assuming that the camera mounted on the vehicle is calibrated, we can use spherical image coordinates and 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 scene point seen from two camera positions and back projected onto a unit sphere (i.e., $\| \mathbf{p_1} \| = \| \mathbf{p_2} \| = 1$) (Figure 3). According to computer vision fundamentals [22], the image coordinates and the two relative unknown camera positions must satisfy the *epipolar constraint* (Figure 3).
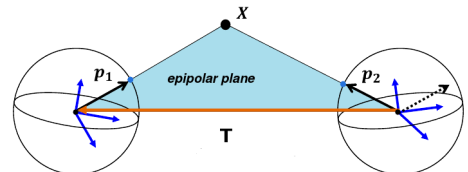

Fig. 3: Epipolar constraint. $\mathbf{p_1}$, $\mathbf{p_2}$, T and X lies on the same plane (the *epipolar plane*).

$$\mathbf{p_2}^T \mathbf{E} \mathbf{p_1} = 0 \qquad (2)$$

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:

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

Equation (2) allows us to compute the essential matrix $\mathbf{E}$ given a set of image coordinate points. $\mathbf{E}$ can then be decomposed into $\mathbf{R}$ and $\mathbf{T}$ [22]. In the case of a 6-DOF motion, the minimum number of image points necessary to compute the essential matrix is 5; indeed $\mathbf{E}$ is function of the rotation (3 parameters) *plus* the translation (3 parameters) *minus* the scale factor (1 parameter).

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

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

allow us to virtually rotate the two camera frames into two new frames $\{C_{p_1}\}$ and $\{C_{p_2}\}$ (Figure 4). $Pitch_i$ 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 $(z_{C_{p_1}} \parallel z_{C_{p_2}} \parallel g)$.
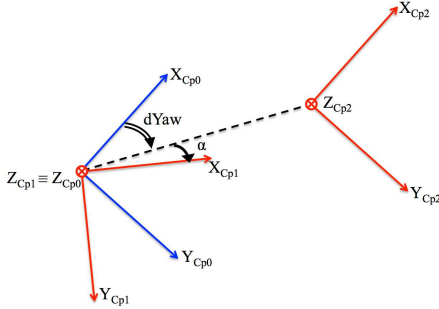

Fig. 4: Write description.

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 4)). The matrix that describes this rotation is the following:

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

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

$$\begin{aligned} \mathbf{p_{C_{P_0}}} &= {}^{Cp0}R_{Cp1} \cdot {}^{Cp1}R_{B_1} \cdot \mathbf{p_1} \\ \mathbf{p_{C_{P_2}}} &= {}^{Cp2}R_{B_2} \cdot \mathbf{p_2} \end{aligned} \qquad (6)$$

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

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

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

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

At this point, being $\mathbf{p_{C_{P_0}}} = [x_0 \ \ y_0 \ \ z_0]^T$ and $\mathbf{p_{C_{P_2}}} = [x_2 \ \ y_2 \ \ z_2]^T$, we impose the epipolar constraint according to (2) and we obtain the homogeneous equation that must be satisfied by all the point correspondences.

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

where $\mathbf{p_0} = [x_0 \ \ y_0 \ \ z_0]^T$ and $\mathbf{p_2} = [x_2 \ \ y_2 \ \ z_2]^T$ 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:

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

IV. OUTLIER DETECTION

In order to address the outlier detection task by taking into account the above-mentioned minimal motion parametrization, we propose two methods. The former is based on RANSAC (1-point RANSAC) and the latter is based on computing α^* as the median of the distribution obtained from all the feature correspondences and selecting the inliers by using the reprojection error. We call this second method "MEdian + Reprojection Error" (Me-RE). Once that the outliers are removed, the motion of the vehicle can be estimated from the remaining inliers using standard methods [11], [22].

A. 1-point RANSAC

One feature correspondence is randomly selected from the set of all the matched features. The motion hypothesis is computed according to (7). 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)).

B. 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 aforementioned 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.

V. PERFORMANCE EVALUATION

We evaluated the performance of the proposed approaches on both synthetic and real data. We compare our 1-point RANSAC and Me-RE methods with the 5-point RANSAC [10] in simulations, and with the 5-point RANSAC [10] and the 8-point RANSAC [23] in experiments on real data.

A. Experiments on synthetic data

We simulated different trajectories of a quadrotor moving in indoor scenarios (Figure 5). The simulations have been performed using the *Robotics and Machine Vision Toolbox* for Matlab [21].

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 5). 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. The vehicle was flying at the fix height of $2m$ above the ground. We generated a circular trajectory (easily repeatable in our flying arena) with a diameter of $1.5m$. 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 6 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 7 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 8

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 9 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 6), the Me-RE algorithm finds more inliers than the 5-point RANSAC. The latter algorithm requires at least 145 iterations according to Table I to insure a good performance.

When the variables $Roll$, $Pitch$ and $dYaw$ are affected by errors(Figures 7 and 8), 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 9), 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.
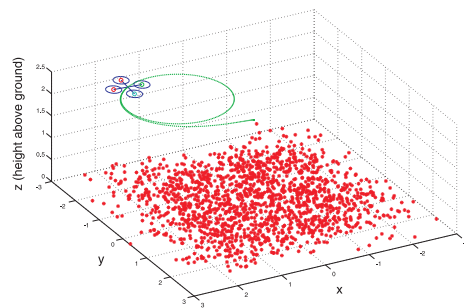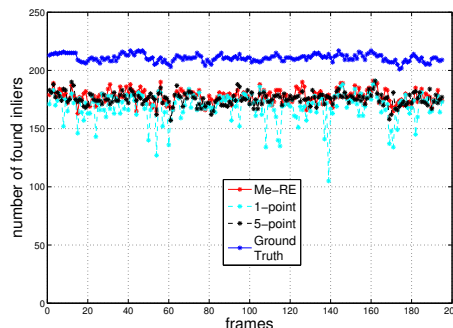

Fig. 5: Synthetic scenario.

Fig. 6: 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.

B. Experiments on real data

We tested our method on a nano quadrotor (Figure 1)[1] 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* [24]. The extrinsic calibration between the IMU and the camera has been
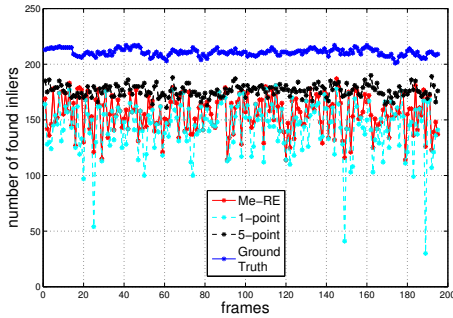
---

[1]http://KMelRobotics.com

Fig. 7: 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.
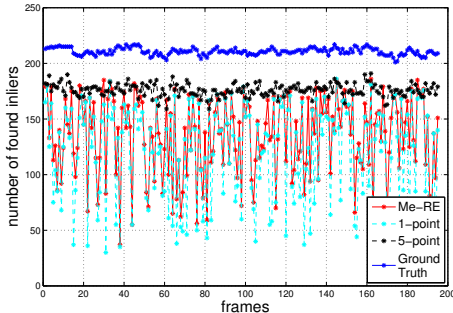

Fig. 8: Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) in presence of perturbations on the *dYaw* angle.

performed using the *Inertial Measurement Unit and Camera Calibration Toolbox* [20]. 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 trajectory has been generated using the TeleKyb Framework [25] (Figure 10). The vehicle followed a circular trajectory ($1.5m$ of diameter, period 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 has been performed using the *Machine Vision Toolbox* from [21].

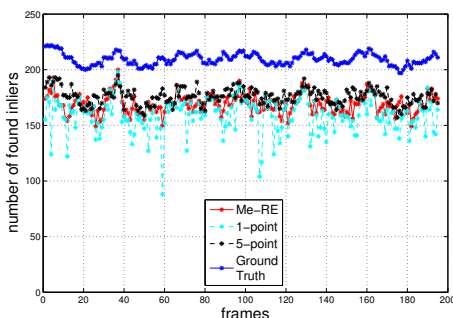To evaluate the performance of our methods, we compared

Fig. 9: 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 ($s_1 = 0.02 * sin(8 * w_c \cdot t)$).

TABLE II: Computation time

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

the number of inliers found by the 1-point RANSAC and Me-RE methods with the number of inliers found by the 5-point RANSAC and the 8-point RANSAC methods. Figure 11 presents the result of this comparison.

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 12 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 II 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 II), 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.
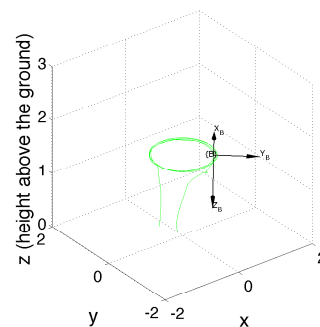

Fig. 10: Plot of the real trajectory. The vehicle's body frame is depicted in black and the green line is the trajectory followed.

VI. CONCLUSIONS

In this paper we presented two algorithms (1-point RANSAC and *Median + Reprojection Error*) to perform outlier detection on computationally constrained micro aerial vehicles. The algorithms operate with the aid of an onboard IMU and assume that the vehicle's motion is locally planar. Both the algorithms rely on the reprojection error to look for
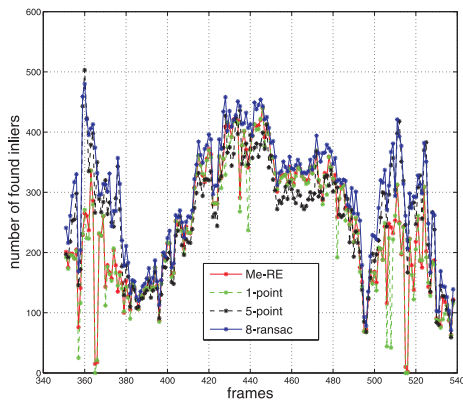
Fig. 11: Number of found inliers by Me-RE (red), 1-point RANSAC (green), 5-point RANSAC (black), 8-point RANSAC (blue) along the trajectory depicted in Figure 10.
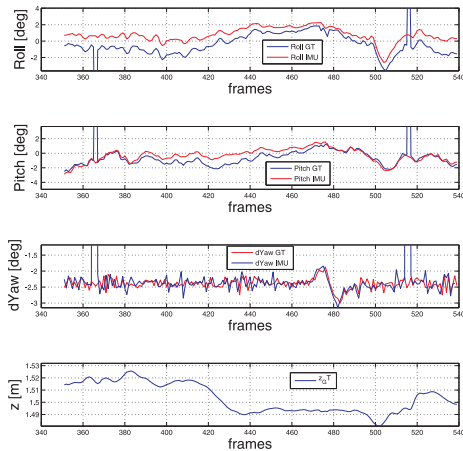

Fig. 12: From the top to the bottom: $Roll$, $Pitch$ and $dYaw$ angles [deg] estimated with the IMU (red) versus $Roll$, $Pitch$ and $dYaw$ angles [deg] estimated with the Optitrack system (blue). The last plot shows the height of the vehicle above the ground (non perfect planarity of motion).

inliers once that the essential matrix has been estimated, but the 1-point RANSAC needs at least 7 iterations to provide a satisfying solution, whereas the Me-RE's computational complexity is linear in the number of features and the performance is better. The Me-RE algorithm can therefore be a good replacement of the 5-point RANSAC when the motion of the vehicle is smooth and the camera framerate is high. The motion can then be refined applying standard methods [11], [22] to the remaining inliers. Considering that $\alpha*$ 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 index of reliability of the Me-RE algorithm.

ACKNOWLEDGMENT

REFERENCES

[1] S. Weiss., D. Scaramuzza, and R. Siegwart, "Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, 2011.

[2] M. Achtelik, S. Lynen, S. Weiss, L. Kneip, M. Chli, and 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, and et al., "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments, under review for the ieee robotics and automation magazine. pdf available on the author webpage." 2013.

[4] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011.

[5] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[6] 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.

[7] O. D. Faugeras and S. Maybank, "Motion from point matches: multiplicity of solutions," *International Journal of Computer Vision*, vol. 4, no. 3, pp. 225–246, 1990.

[8] J. Philip, "A non-iterative algorithm for determining all essential matrices corresponding to five point pairs," *The Photogrammetric Record*, vol. 15, no. 88, pp. 589–599, 1996.

[9] B. Triggs, "Routines for relative pose of two calibrated cameras from 5 points," 2000.

[10] D. Nistér, "An efficient solution to the five-point relative pose problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 756–770, 2004.

[11] H. Stewénius, C. Engels, and D. Nistér, "Recent developments on direct relative orientation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 4, pp. 284–294, 2006.

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

[13] T. P. Fraundorfer F. and 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.

[14] O. Naroditsky, X. S. Zhou, J. Gallier, S. I. Roumeliotis, and K. Daniilidis, "Two efficient solutions for visual odometry using directional correspondence," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 4, pp. 818–824, 2012.

[15] L. Kneip, M. Chli, and 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, August 2011.

[16] D. Ortin and J. Montiel, "Indoor robot motion based on monocular images," *Robotica*, vol. 19, no. 3, pp. 331–342, 2001.

[17] D. Scaramuzza, F. Fraundorfer, and 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.

[18] D. Scaramuzza, "1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints," *International journal of computer vision*, vol. 95, no. 1, pp. 74–85, 2011.

[19] ——, "Performance evaluation of 1-point-ransac visual odometry," *Journal of Field Robotics*, vol. 28, no. 5, pp. 792–811, 2011.

[20] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 561–575, 2007.

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

[22] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge Univ Press, 2000, vol. 2.

[23] 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*, pp. 61–62, 1987.

[24] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2004.

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