

Comparison of optical flow-based linear angular rate estimation methods considering real flight data

Gabor Jevuczo and Peter Bauer

Abstract—The angular velocities of an observer moving through an arbitrary scene can be obtained solely from mono camera image sequences, utilizing the optical flow. A major obstacle in this approach is that, the optical flow is non linearly related to the 3-D motion parameters. In this work three possible linear methods are presented to solve this nonlinear equation and then these methods are evaluated and compared using real mono camera images and real sensor measurements collected during two test flights.

I. INTRODUCTION

Angular velocity estimated from mono camera images can be utilized for sensor error detection on aerial vehicles as shown in [1], which is a vital task according to [2] and [3]. Although it is not the only application in which angular rate estimation is advantageous, it can also be important in many robot vision applications such as augmenting traditional inertial measurement units. There are many approaches to obtain velocity and angular rate from optical flow [4], [5], [6], [7]. In [8] a nonlinear optimization-based solution is presented, [4] uses the differential epipolar constraint to recover translational and rotational velocity vectors which leads to a nonlinear optimization problem. Many other methods require the solution of a nonlinear system of equations which relates the optical flow field with the 3-D motion parameters.

To obtain the angular rates from this system of equations there are multiple approaches in the literature. [9] introduces a linear method by which this equation can be solved linearly in two steps, when at least 8 points are detected. Applying these results [10] proposed a method called Motion Detection using Randomized Hough Transform which utilizes the same linear equation but then applies the Randomized Hough Transform to reduce the effect of noise in the computation of the 3-D motion parameters. [8] presents methods to estimate the Focus of Expansion (FOE) and angular rate of an aircraft based on the optical flow. [11] proposed a novel multiple step linear solution for the estimation of aircraft angular rates. [12] presents a linear solution which requires the knowledge

of the direction of the translation vector because if it is known then a unit vector can be constructed which will eliminate the feature point depth from the equation resulting a linear solution for the estimation of angular rate.

This work solely examines linear methods and also proposes a novel linear method. To the best of our knowledge, in the literature these methods are not tested on real camera image sequences and flight data. [9] presented their method on simulated data, similarly in [10] they also tested their method on simulated data in the general motion case. They also applied real image sequences but only for the case of pure translation. [12] showed their methods capability on simulated data and also on a computer-graphic generated image sequence flying through the Yosemite valley.

Therefore the main goal of the current work is to compare the existing linear solutions for the estimation of angular rate components using real mono camera image sequences and IMU measurements collected during test flights with a multicopter. The comparison is based on precision and runtime of the presented algorithms, thus helping the reader to find the proper algorithm for their specific application.

The structure of the paper is as follows. Section II introduces the equations which relate the optical flow with the 3-D motion. Each one of Section III, IV and V presents a method for the angular rate estimation utilizing the optical flow. Section VI presents the test flight trajectories and the workflow of video processing. Section VII compares the presented methods. Finally, Section VIII concludes the paper.

II. OPTICAL FLOW EQUATIONS

The (1) system of equations relates the optical flow field and the 3-D motion parameters, a detailed derivation of these equations are presented in [12] and [10] where the coordinate systems of the moving observer and the camera coincide. Derivations for a camera placed away from the center of gravity are presented in [13] and [1]. In this work these are derived in the camera coordinate system which can be seen in Fig. 1 where the following notations are used. F_i is the i -th feature point. (η_i^C) is the position vector of F_i in the camera system. (μ_i, ν_i) are image coordinates of F_i on the image plane at $Z_c = f$ focal length in pixel (px) dimension. The camera is mounted rigidly to the drone and it is tilted down by 28.38° relative to the horizontal axis of the drone.

The research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems. (RRF-2.3.1-21-2022-00002) Project no. TKP2021-NVA-01 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme.

P. Bauer is with Systems and Control Laboratory, HUN-REN Institute for Computer Science and Control (HUN-REN SZTAKI), Budapest, Hungary and Széchenyi Egyetem, University of Győr, Győr, Hungary. bauer.peter@sztaki.hun-ren.hu

G. Jevuczo is with Systems and Control Laboratory, HUN-REN Institute for Computer Science and Control (HUN-REN SZTAKI), Budapest, Hungary jevuczo.gabor@sztaki.hun-ren.hu

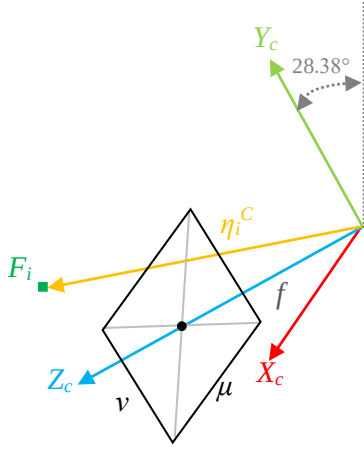


Fig. 1. The camera coordinate system.

The equations in camera system are as follows.

$$\begin{aligned} \begin{bmatrix} \dot{\mu}_i \\ \dot{\nu}_i \end{bmatrix} &= \begin{bmatrix} \frac{\mu_i \nu_i}{f} & -f - \frac{\mu_i^2}{f} & \nu_i \\ \frac{\nu_i^2}{f} + f & -\frac{\mu_i \nu_i}{f} & -\mu_i \end{bmatrix} \begin{bmatrix} p^C \\ q^C \\ r^C \end{bmatrix} \\ &+ \begin{bmatrix} -f & 0 & \nu_i \\ 0 & -f & \mu_i \end{bmatrix} \begin{bmatrix} u^C \\ v^C \\ w^C \end{bmatrix} \frac{1}{\eta_{3i}} \\ &= A_{\omega_i} \omega^C + B_{v_i} V^C \frac{1}{\eta_{3i}} \end{aligned} \quad (1)$$

where $\omega^C = [p^C \ q^C \ r^C]^T$ and $V^C = [u^C \ v^C \ w^C]^T$ are the drone's angular rate and velocity components respectively, transformed to the camera coordinate system and $(\dot{\mu}_i \ \dot{\nu}_i)$ are the velocity of F_i (optical flow) on the image plane.

The difficulty to solve this system of equations originates from that the optical flow field is non linearly related to the 3-D motion and feature depth parameters. In the next three sections, three linear methods are presented, two of which are from the literature, and the third one is proposed by the authors.

III. METHOD 1

The first method was presented in [9], where they derived a linear solution for (1), this method was also presented in [10] where Randomized Hough Transform (RHT) was applied to reduce the effect of noise in the computation of the 3-D motion parameters but the RHT is not considered here because we solely focus on the linear solutions. The first step is to divide both sides of (1) by w^C assuming $w^C \neq 0$.

$$\begin{aligned} \frac{\dot{\mu}_i}{w^C} &= \frac{\mu_i}{\eta_{3i}} - \frac{f q^C}{w^C} + \frac{\nu_i r^C}{w^C} - \frac{\mu_i^2 q^C}{f w^C} - \frac{f u^C}{\eta_{3i} w^C} + \frac{\mu_i \nu_i p^C}{f w^C} \\ \frac{\dot{\nu}_i}{w^C} &= \frac{\nu_i}{\eta_{3i}} + \frac{f p^C}{w^C} - \frac{\mu_i r^C}{w^C} + \frac{\nu_i^2 p^C}{f w^C} - \frac{f v^C}{\eta_{3i} w^C} - \frac{\mu_i \nu_i q^C}{f w^C} \end{aligned} \quad (2)$$

The following step is to substitute the image coordinates of the focus of expansion (3) (FOE) (or focus of contraction (FOC)) into (2).

$$\mu_f = \frac{u^C}{w^C} \quad \nu_f = \frac{v^C}{w^C} \quad (3)$$

The μ_f, ν_f FOE and FOC indicates where the camera is heading (or how an object is moving towards the camera) when translational motion is present. For example when the camera is moving forward, the image velocity vectors diverge from this point (FOE) and when the camera moves backward then the image velocity vectors will converge towards that point (FOC).

After the substitution $\frac{1}{\eta_{3i}}$ is expressed from both equations. By equating them and rearranging the terms the following form is obtained (4).

$$\begin{aligned} \dot{\mu}_i \nu_i - \dot{\nu}_i \mu_i &= -f \nu_i (q^C + r^C \nu_f) + (\nu_i^2 + \mu_i^2) r^C \\ &\quad - \mu_i \nu_i (q^C \mu_f + p^C \nu_f) + f \dot{\mu}_i \nu_f \\ &\quad + (f^2 + \mu_i^2) q^C \nu_f - f \mu_i (p^C + r^C \mu_f) \\ &\quad - f \dot{\nu}_i \mu_f + (f^2 + \nu_i^2) p^C \mu_f \end{aligned} \quad (4)$$

(4) can be rewritten in matrix form with the following eight unknowns [9]:

$$\begin{bmatrix} \dot{\mu}_i \nu_i - \dot{\nu}_i \mu_i \end{bmatrix} = \begin{bmatrix} -f \dot{\nu}_i \\ f \dot{\mu}_i \\ -f \mu_i \\ -f \nu_i \\ -\mu_i \nu_i \\ \mu_i^2 + \nu_i^2 \\ f^2 + \nu_i^2 \\ f^2 + \mu_i^2 \end{bmatrix}^T \begin{bmatrix} \mu_f \\ \nu_f \\ p^C + r^C \mu_f \\ q^C + r^C \nu_f \\ q^C \mu_f + p^C \nu_f \\ r^C \\ p^C \mu_f \\ q^C \nu_f \end{bmatrix} \quad (5)$$

Note that here, one feature point and its corresponding optical flow will result in one equation. Therefore, this linear equation can be solved with eight or more feature points by calculating the pseudoinverse.

The objective is to calculate the angular rates of the drone therefore a second step is also necessary, by solving (6) for ω^C from the results of (5).

$$\begin{bmatrix} p^C + r^C \mu_f \\ q^C + r^C \nu_f \\ q^C \mu_f + p^C \nu_f \\ r^C \\ p^C \mu_f \\ q^C \nu_f \end{bmatrix} = \begin{bmatrix} 1 & 0 & \mu_f \\ 0 & 1 & \nu_f \\ \nu_f & \mu_f & 0 \\ 0 & 0 & 1 \\ x_f & 0 & 0 \\ 0 & y_f & 0 \end{bmatrix} \begin{bmatrix} p^C \\ q^C \\ r^C \end{bmatrix} \quad (6)$$

The drawback of this method is that it cannot be applied for cases when the $w^C = 0$, so for example when flying with a multicopter equipped with downward looking gimbal stabilized camera, the $w^C \neq 0$ assumption will not hold unless there is a continuous change in the flight altitude.

IV. METHOD 2

The second method was introduced in [12]. They divided the solution of (1) into three sets of equations. The first relates the translational motion with the optical flow field. After they obtained the translation direction vector from the first set of equations, they use it to solve solely the angular rates from (1). After both the translation and rotation are calculated then a relative depth map is computed with the third sets of equations. In this work only the angular rate estimation is considered, with the assumption that the direction vector is already known (see Section VI).

The method proposed by [12] states that if the direction of the translation vector t is known, then a unit vector d_i^T perpendicular to the translational component of the flow field can be constructed. Multiplying both sides of (1) with d_i^T will eliminate the dependence on $\frac{1}{\eta_{3i}}$ i.e. projects the dependent term to zero, and thereby results in a linear equation.

Note that here, substitution of the $t = \frac{V^C}{\|V^C\|}$ translation direction vector gives $b_i = \|V^C\|/\eta_{3i}$ as the relative inverse depth.

$$d_i^T O F_i = d_i^T A_{\omega_i} \omega^C + d_i^T B_{v_i} t b_i \quad (7)$$

The d_i^T unit vector can be constructed as follows. Taking into account that the second term of (7) must be zero ($d_i^T B_{v_i} t b_i = 0$), the d_i^T results as:

$$d_i^T = \frac{R_{\pi/2} B_{v_i} t}{\|B_{v_i} t\|} \quad (8)$$

where $R_{\pi/2}$ is a 90° rotation matrix in an arbitrary direction to obtain a perpendicular vector.

It is important to note that this d_i^T vector is different for each feature point (and thereby for the each corresponding optical flow). With this projection to zero, the dependence on the depth of the feature point $\frac{1}{\eta_{3i}}$ from equation (1) is eliminated resulting in a linear solution.

V. METHOD 3

Similarly to method2, in method3 first the translation direction vector $t = [t_x \ t_y \ t_z]^T$ is substituted to (1) but without utilizing the projection with d_i^T unit vector and thereby preserving b_i as unknown in the linear system of equations.

In method2 after the projection with d_i^T vector, only the first term of (7) will remain therefore the system of equations has to be solved only for 3 unknowns. Contrary to this in method3, the relative feature point depth b_i is not eliminated i.e. the second term of (7) is not projected to zero. Thereby the number of the unknowns will depend on the number of feature points so it will have $3 + N$ unknowns. As a result, the coefficient matrix of the system of equations grow from the $N \times 3$ in method2, to $2N \times (3 + N)$ in method3.

After substituting the translation direction into (1) the equation (9) is obtained which can be solved by calculating the pseudoinverse of the coefficient matrix.

$$\begin{bmatrix} \dot{\mu}_i \\ \dot{\nu}_i \end{bmatrix} = \begin{bmatrix} \frac{\mu_i \nu_i}{f} & -f - \frac{\mu_i^2}{f} & \nu_i & t_z \mu_i - t_x f \\ f + \frac{\nu_i^2}{f} & -\frac{\mu_i \nu_i}{f} & -\mu_i & t_z \nu_i - t_y f \end{bmatrix} \begin{bmatrix} p^C \\ q^C \\ r^C \\ b_i \end{bmatrix} \quad (9)$$

VI. IMAGE PROCESSING

Two flights were considered during this work, both flown above ZalaZONE Smart City proving ground [14] with a DJI M600 Pro drone [15] equipped with a rigidly mounted, oblique Basler acA2040-35gc [16] camera. The two flight trajectories are shown in Fig. 2. Both of the flights contain a take-off to reach the constant flight altitude at 35 meters and after some hovering at the starting position, predefined aircraft-like fixed trajectories are flown in which the longitudinal velocity is 5m/s in both flights and the roll angles of the banked turns are 10° in flight 539 and 15° in flight 540.

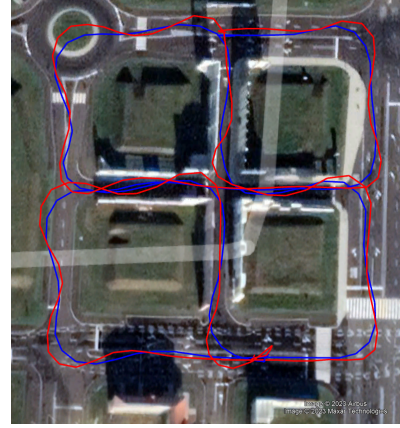


Fig. 2. The trajectories of the considered flights, flight 539 with blue and the flight 540 with red.

The obtained camera recordings were processed offline with the following workflow implemented in Python language.

To calculate the optical flow, feature point correspondences are needed between the consecutive images. It requires the images to be processed one by one from the video.

First the *goodFeaturesToTrack* (Shi-Tomasi Corner Detector with the following parameters 200 maximum number of feature points and 0.05 qualityLevel; 40 minDistance; 5 block size) OpenCV function was applied on greyscale converted images to find the feature points on the first frame (indicated with index k). Then from these points the optical flow is calculated by the *OpticalFlowPyrLK* (sparse optical flow with iterative Lucas-Kanade method applying pyramids with maximum 7 level, 9×9 window size, 10 count and 0.03 epsilon) OpenCV function, as a forward optical flow from the k frame to the $k + 1$ frame.

To filter out the outliers, a backward optical flow calculation is applied utilizing the *OpticalFlowPyrLK* function from the frame $k + 1$ to the frame k . The feature points which are also found by the backward optical flow and also by the *goodFeaturesToTrack* within a certain range on frame k , are then kept for later calculations.

After finding the image correspondences, distortion compensation was applied.

The translation direction of the camera between two consecutive image frames can be obtained from the essential matrix, therefore the *findEssentialMat* OpenCV function was applied with RANSAC method (with 0.999 confidence; 0.5 threshold parameters for the RANSAC method). Then this essential matrix is decomposed according to [17] into the translation direction and the rotation matrix between the two image frames.

By this the μ_i , ν_i , $\dot{\mu}_i$, $\dot{\nu}_i$ and t terms are determined allowing us to solve the previously presented methods which are also implemented in the same Python workflow. To see how these algorithms behave for different number of feature points, after calculating the optical flow and filtering out the outliers, we select 10, 20, 30, 40, 50 or 60 from the remaining feature points and the corresponding optical flows, to calculate the essential matrix and the three different methods.

Both the image processing and equation solving was carried out on a desktop PC equipped with a 12 core AMD Ryzen™ 9 7900 processor and with a kit of 6000MT/s, CL30, 32GB DDR5 RAM memory and it was developed using Python language in Microsoft Visual Studio Code development environment.

VII. RESULTS AND COMPARISON

The evaluation of the estimation results (from the Python workflow) was done using Matlab. The three estimation methods are compared using the angular rates measured by DJI's onboard IMU sensor, collected during the test flights presented in Section VI. The IMU measurements were recorded with 50Hz and then resampled to 10Hz to match the 10FPS camera recordings. Since the IMU is fixed to the body system, and the angular rate components are estimated in the camera system, the estimations must be transformed to the body system for comparison.

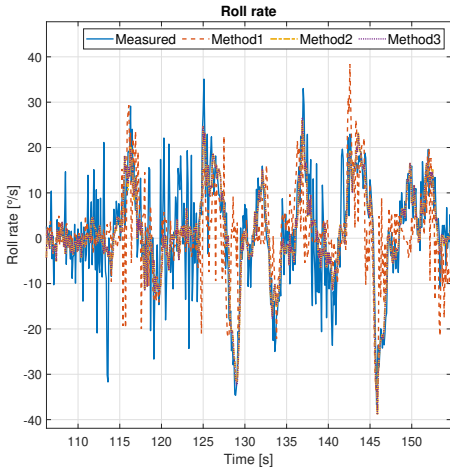


Fig. 3. Roll rate estimation result for flight 540.

Fig. 3, 4 and 5 show the plots of estimated angular rate components (calculated with 60 feature points) with the measured values. Due to the limited space and for better visibility, only the results from characteristic parts of flight 540 are presented in these plots. From the plots, we can infer that method1 has the worst estimation from all and method 2 and 3 are successfully able to estimate all the angular velocity components while containing even less noise than the measured data.

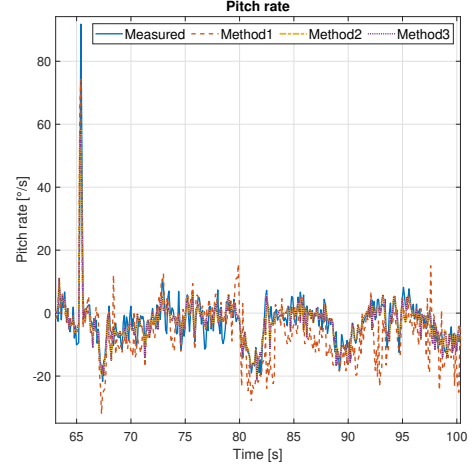


Fig. 4. Pitch rate estimation result for flight 540.

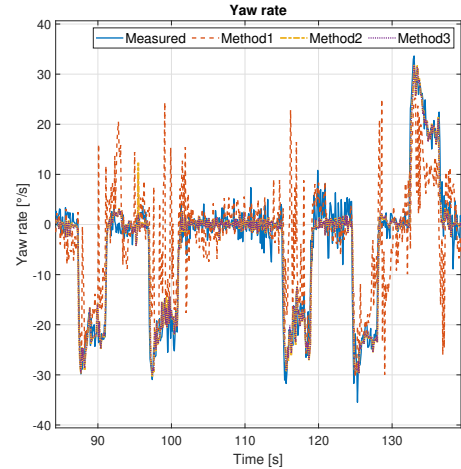


Fig. 5. Yaw rate estimation result for flight 540.

This can be validated by calculating the Mean Absolute Error (MAE), the Maximum Absolute Error (MaxAE) and the Minimum Absolute Error (MinAE) between the estimated and measured components for the whole flight. These values are shown in Fig. 6, 7 and 8 where each one individually illustrates the MAE, MaxAE and MinAE for one component.

Regarding the MAE we can see that increasing the number of feature points will decrease the estimation error until around 30 or 40 feature points, then it takes on an approximately constant value. This shows that after a certain number of feature points, we reach the precision which can be achieved by these methods and there is no more benefit in applying more points on the contrary, it will slow down the algorithms.

The MAE values for the method2 and method3 are about the same in all cases. The MAE values for the roll rate estimations are around $\sim 4.3^\circ/s$ for method2 and method3 and is around $\sim 6.5^\circ/s$ for method1. For the pitch estimations these values are around $\sim 2.8^\circ/s$ for method2 and 3 and around $\sim 5.4^\circ/s$ for method1. And for the yaw rate estimations the MAE values are around $\sim 2^\circ/s$ for method 2 and 3 and $\sim 5.7^\circ/s$ for method1.

The MaxAE values are quite high, the largest errors are caused when the multicopter starts the predefined aircraft like trajectories, it starts with a sudden, large pitch movement (large nod) and as a result the optical flow based estimation methods were underestimating the angular rate and this causes the large maximal absolute error. This can be seen in Fig. 4 at 65s. Additionally the noisy measurements also take role in this. In the yaw rate estimation of method2 also a larger spike occurred, this is shown in Fig. 5 at 95.5s. Such a spike like differences would not cause any problem during sensor error detection because their effects can be suppressed by using a well selected up-down counter (see in [1]). From all the three method3 has the smallest MaxAE value for the roll and yaw angular rate components.

Regarding the minimum absolute errors they are all below $1.5 \cdot 10^{-3} \text{deg/s}$.

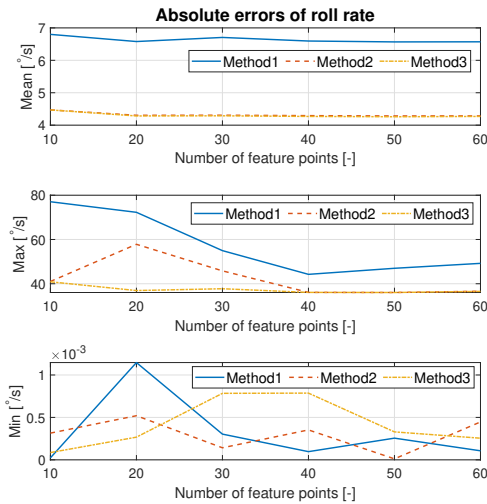


Fig. 6. The absolute errors of roll rate estimation results for flight 540.

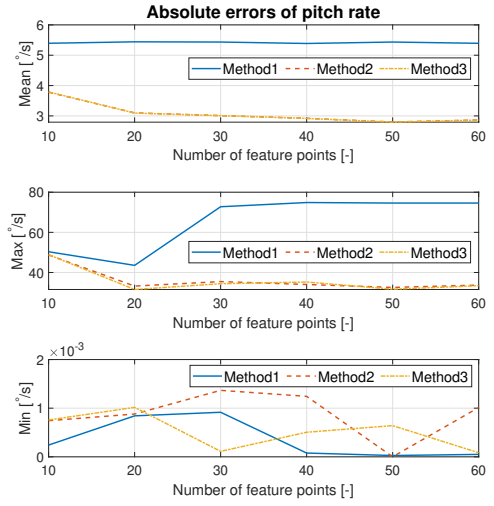


Fig. 7. The absolute errors of pitch rate estimation results for flight 540.

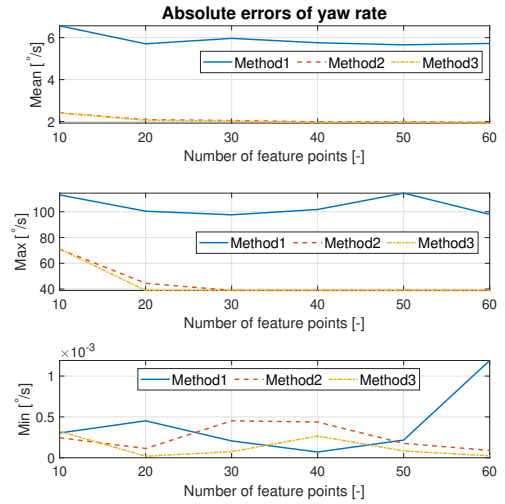


Fig. 8. The absolute errors of yaw rate estimation results for flight 540.

Fig. 9 shows the processing time of each method calculated with different number of feature points i.e. the time required by the solver in each time step (to solve the system equations) summed together for the whole flight duration.

This time measurement includes in case of method1 the construction of the matrices and the vectors in (5) and (6) and the solving of these equations with a pseudoinverse calculation. For method2 this contains the construction of the d_i^T vector in (8) and the A_{ω_i} matrix and OF vector in (7) and the solving of this equation (of course with excluding the second term of equation (7)) with the use of a pseudoinverse. For method3 this time includes the construction of the known terms in (9) and the solving of (9) with a pseudoinverse calculation.

One can see that by increasing the number of feature points the processing time increases. The fastest from all is method1, but it is the least accurate. Method2 and method3 has a very similar run time however the benefit of method 3 is that relative depth is calculated in one step together with the angular rate while in method2 a separate step is needed to calculate it as written in [12].

Fig. 9 also shows that while the processing time of method2 is linearly increases with the number of feature points, the processing time of method3 progressively increases with the number of feature points, this is due to the difference in number of unknowns and equations (see Section V).

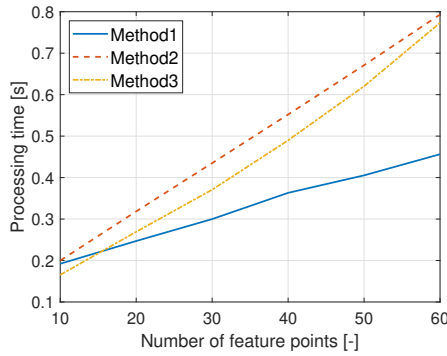


Fig. 9. Processing time of the methods with different number of feature points.

VIII. CONCLUSIONS AND FUTURE WORKS

In this work, three different linear methods are compared for estimating the angular rate of a multicopter using solely mono camera images obtained in real test flights.

Method1 is a direct elimination which constructs one equation from two optical flow equations by substituting in the focus of expansion thus eliminating the $\frac{1}{\eta_{3i}}$ inverse feature point depth. Method2 is an elimination with a projection, this also eliminates the $\frac{1}{\eta_{3i}}$ but by a projection with a vector which can be obtained, if the translation direction is already known. Method3 which is proposed by the authors also uses the translation direction vector but without applying a the projection.

Comparing the estimations with the measured values, it can be seen that although method1 is the fastest, it also gives the worst results. Method2 and method3 successfully estimates the angular rate components and their execution time is very similar, however the advantage of proposed novel method is that the relative inverse depth is also determined in one step with the angular rate estimation. We need to emphasize that this is a relative inverse depth due to the fact that the absolute value of the velocity is not known. However if one can obtain the absolute value of the velocity vector for example from a GPS measurement, then the feature point distances can be determined.

For future work, we aim to use the results presented in this work for angular rate sensor error detection and then

later extending it to IMU magnetic and acceleration sensor error detection.

REFERENCES

- [1] Gabor Jevuczo and Peter Bauer. "Optical flow-based vertical angular rate fault detection on UAVs". In: *IFAC-PapersOnLine* 58.4 (2024). 12th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2024, pp. 109–114. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2024.07.202>.
- [2] G. Heredia et al. "Detection of Sensor Faults in Autonomous Helicopters". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. 2005, pp. 2229–2234. DOI: 10.1109/ROBOT.2005.1570444.
- [3] Philippe Goupil et al. "An overview of the FP7 RECONFIGURE project: industrial, scientific and technological objectives". In: *International Federation of Automatic Control (IFAC)*. 2015, pp. 976–981.
- [4] Moshe Hamaoui. *Optical flow-based state estimation for guided projectiles*. Tech. rep. ARL-TR-7321. US Army Research Laboratory, 2015.
- [5] Li Xiang et al. "Three-dimensional attitude determination using vector observations and optical flow sensors". In: *Journal of Physics: Conference Series* 2005.1 (Aug. 2021), p. 012098. DOI: 10.1088/1742-6596/2005/1/012098.
- [6] Giancarmine Fasano et al. "Satellite Angular Velocity Estimation Based on Star Images and Optical Flow Techniques". In: *Sensors* 13.10 (2013), pp. 12771–12793. ISSN: 1424-8220. DOI: 10.3390/s131012771. URL: <https://www.mdpi.com/1424-8220/13/10/12771>.
- [7] G. Gebert et al. "Optical flow angular rate determination". In: *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*. Vol. 1. 2003, pp. I–949. DOI: 10.1109/ICIP.2003.1247121.
- [8] J.J. Kehoe et al. "Partial aircraft state estimation from optical flow using non-model-based optimization". In: *2006 American Control Conference*. 2006, 6 pp.–. DOI: 10.1109/ACC.2006.1657154.
- [9] Mitiche Amar, Zhuang Xinhua, and Haralick Robert. "Three-dimensional interpretation of optical flow". In: *Computer Vision and Image Processing (UNICOM Applied Information Technology Reports)*. Chapman Hall, 1990, pp. 182–208. ISBN: 9780412377303.
- [10] Jukka Heikkonen. "Recovering 3-D motion parameters from optical flow field using randomized Hough transform". In: *Pattern Recognition Letters* 16.9 (1995), pp. 971–978. ISSN: 0167-8655. DOI: [https://doi.org/10.1016/0167-8655\(95\)00025-C](https://doi.org/10.1016/0167-8655(95)00025-C).
- [11] Szabolcs Kun and Peter Bauer. "Improvements in Optical Flow-based Aircraft Partial State Estimation". In: *2021 17th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*. 2021, pp. 1–4. DOI: 10.1109/CNNA49188.2021.9610822.
- [12] David J. Heeger and Allan D. Jepson. "Subspace methods for recovering rigid motion I: algorithm and implementation". In: *Int. J. Comput. Vision* 7.2 (Jan. 1992), pp. 95–117. ISSN: 0920-5691. DOI: 10.1007/BF00128130.
- [13] Peter Bauer and Szabolcs Kun. "Optical flow-based angular rate sensor fault detection on UAVs". In: *IFAC-PapersOnLine* 55.14 (2022). 11th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2022, pp. 46–51. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2022.07.581>.
- [14] AVL. *AVL Zolazone Smart City module*. AVL Zolazone, 2022.
- [15] DJI. *DJI M600 Pro hexacopter*. <https://www.dji.com/hu/matrice600-pro>. 2017.
- [16] Basler. *acA2040-35gc camera*. 2022.
- [17] Qi Cai et al. "Equivalent Constraints for Two-View Geometry: Pose Solution/Pure Rotation Identification and 3D Reconstruction". In: *International Journal of Computer Vision* 127.2 (2019), pp. 163–180. ISSN: 1573-1405. DOI: 10.1007/s11263-018-1136-9.