Robust Optical-Flow Based Self-Motion Estimation for a Quadrotor UAV

Volker Grabe, Heinrich H. Bülthoff, and Paolo Robuffo Giordano

***Abstract*— Robotic vision has become an important field of research for micro aerial vehicles in the recent years. While many approaches for autonomous visual control of such vehicles rely on powerful ground stations, the increasing availability of small and light hardware allows for the design of more independent systems. In this context, we present a robust algorithm able to recover the UAV ego-motion using a monocular camera and on-board hardware. Our method exploits the continuous homography constraint so as to discriminate among the observed feature points in order to classify those belonging to the dominant plane in the scene. Extensive experiments on a real quadrotor UAV demonstrate that the estimation of the scaled linear velocity in a cluttered environment improved by a factor of 25% compared to previous approaches.**

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are becoming an increasingly popular robotic platform for a large variety of applications, ranging from typical 'spatial awareness' tasks such as mapping, patrolling, surveillance, inspection, to the more recent field of *Aerial Service Robotics* where a direct interaction with the environment is envisaged. The recently EU-funded projects sFly [1], AIRobots [2] and ARCAS [3] show the high level of interest within the scientific community in exploiting these systems. In this context, quadrotor UAVs are a particularly popular choice thanks to their low cost, mechanical simplicity and robustness, low weight and size, and high agility. This has led to an impressive number of results, both in terms of accurate motion control and sensor-based autonomy. In fact, contrarily to fixed wing aircrafts, quadrotors can easily navigate in small cluttered indoor scenes, and then represent a highly valuable choice for operations in unaccessible or generically hazardous environments.

Pivotal for the autonomy of any mobile robot and, in particular, of UAVs/quadrotors is the ability to obtain information on the surrounding environment to, e.g., detect the presence of obstacles or estimate the robot self-motion. Among the many existing sensors, cameras represent a valuable tool for obtaining a rich sensory feedback combined with the benefits of being small, light weight and affordable devices. Use of cameras for real-time robotic applications, i.e., robotic vision, lies at the intersection of robotics and computer vision and constitutes an active field of research in, e.g., providing sufficiently reliable spatial localization to mobile robots. This goal is reaching a good maturity thanks to the increasing availability of powerful small PCs which allow to process on-board and in real-time the high information content generated by cameras. Thus, cameras are of particular interest for use on flying vehicles and have been exploited in many indoor and outdoor scenarios, see, e.g., leader-follower and automatic landing maneuvers in [4], [5] as well as for full three dimensional mapping of the environment [6], [7].

In this work, we consider the problem of self-motion estimation for a quadrotor UAV equipped with a monocular onboard camera by exploiting measurements of the instantaneous optical flow w.r.t. a static environment. Use of optical flow for self-motion estimation has a long history in computer and robotic vision. It has been used in the context of UAVs in simulated environments [8], using on-board hardware [9] or in closed-loop control [10]. However, only recently some solutions presenting full closed-loop motion control based on optical flow and relying on sole on-board hardware have been presented [11]. One of the advantages of optical-flow based solutions is the low-computational power required for estimating self-motion. This is in contrast with more sophisticated algorithms based on 3D map reconstruction [12], [13], [14], which rely on ad-hoc ground stations for implementing the intensive computations needed to maintain a map of large environments.

In particular, this paper extends the results presented in [11] which were still based on several assumptions on the observed environment, namely the fact that *all* the detected feature points were assumed as belonging to a single ground plane. While this assumption is usually reasonable in many outdoor and even indoor environments, it can be clearly restrictive in other situations. We therefore present an approach to segment a dominant plane of any orientation within an unknown environment. Besides being able to still recover self motion, we also obtain, as a byproduct, the possibility of classifying features as belonging to potential obstacles. This then allows for a robust ego-motion estimation by means of a computationally light approach. Furthermore, as opposed to more sophisticated systems, our approach does not rely on the presence of a map or any other prior knowledge of the scene.

The paper is structured as follows: in Sect. II, we describe the theoretical foundations of our approach and line out its main features. In Sect. III we present an algorithm able to exclude outlier features from the dominant plane, and in Sect. IV we illustrate a practical method for coping with

V. Grabe, and P. Robuffo Giordano are with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany {volker.grabe, prg}@tuebingen.mpg.de.

H. H. Bülthoff is with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany, and with the Department of Brain and Cognitive Engineering, Korea University, Seoul, 136-713 Korea. E-mail: hhb@tuebingen.mpg.de.

high frame rates during the computation of optical flow. Section V then presents our platform and reports the results of the conducted experiments, which are then discussed in Sect. VI. Finally, Sect. VII draws some concluding remarks and addresses future projects.

II. SUMMARY OF SELF-MOTION ESTIMATION FROM OPTICAL FLOW

In this work, we exploit the 4-point algorithm for planar scenes based on the continuous homography constraint [15] in order to solve the self-motion estimation problem. This is complemented by taking into account the angular velocity measurements obtained from an on-board Inertial Measurement Unit (IMU) in order to improve the quality of the self-motion estimation. Full details of this approach can be found in [11]. For the reader's convenience, we will nevertheless briefly summarize the main features of the employed method.

In our setup, we consider a facing down camera at an angle of 45° w.r.t. the forward direction of motion. Features in the environment are tracked between consecutive frames to compute an optical flow field $\Phi = ((\boldsymbol{x}_1, \boldsymbol{u}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{u}_n))$ as function of n pairs $(\boldsymbol{x}_i, \boldsymbol{u}_i)$ of detected features on the image plane $\boldsymbol{x}_i \in \mathbb{R}^3$ and associated image velocities $\boldsymbol{u}_i \in \mathbb{R}^3$. In many scenarios, such as indoor hallways or areal coverage, one can safely assume presence of a dominant ground plane spanning most of the observed features. This assumption is exploited by the classical reconstruction algorithm based on the continuous homography constraint [15], which was extended in [11] to include measurements of the angular velocity from an onboard IMU.

A. Review of the continuous homography constraint

Seen from a moving camera, the apparent velocity of a point $\boldsymbol{X} \in \mathbb{R}^3$ still in space as a result of the camera motion is given by

$$\dot{\boldsymbol{X}} = \hat{\boldsymbol{\omega}}\boldsymbol{X} + \boldsymbol{v} \tag{1}$$

where $\boldsymbol{v} \in \mathbb{R}^3, \boldsymbol{\omega} \in \mathbb{R}^3$ are the camera linear/angular velocity (both expressed in the camera frame), and $\hat{\boldsymbol{\omega}} \in so(3)$ is the skew-symmetric matrix associated to the vector $\boldsymbol{\omega} \in \mathbb{R}^3$.

Consider a set of features located on a common plane of equation $\boldsymbol{N}^T\boldsymbol{X} = d$ where $\boldsymbol{N} \in \mathbb{S}^2$ is the unit normal vector to the plane, and $d \in \mathbb{R}$ the distance of the plane to the camera frame. The plane constraint can be rephrased as $\frac{1}{d}\boldsymbol{N}^T\boldsymbol{X} = 1$ so that eq. (1) becomes:

$$\dot{\boldsymbol{X}} = \hat{\boldsymbol{\omega}}\boldsymbol{X} + \boldsymbol{v}\frac{1}{d}\boldsymbol{N}^T\boldsymbol{X} = \left(\hat{\boldsymbol{\omega}} + \frac{1}{d}\boldsymbol{v}\boldsymbol{N}^T\right)\boldsymbol{X} = \boldsymbol{H}\boldsymbol{X}. \tag{2}$$

Matrix $\boldsymbol{H} \in \mathbb{R}^{3\times3}$ is commonly referred to as *continuous homography matrix*. $\boldsymbol{H}$ encodes the camera linear/angular velocity $(\boldsymbol{v}, \boldsymbol{\omega})$, as well as the scene structure $(\boldsymbol{N}, d)$.

Defining $\lambda\boldsymbol{x} = \boldsymbol{X}$ for a scalar depth factor λ as the image of a point $\boldsymbol{X}$, and exploiting the fact that $\dot{\boldsymbol{X}} = \dot{\lambda}\boldsymbol{x} + \lambda\boldsymbol{u}$ and $\dot{\boldsymbol{x}} = \boldsymbol{u}$, where $\boldsymbol{u}$ is the observed velocity of the point $\boldsymbol{x}$ on the image plane, we get:

$$\boldsymbol{u} = \boldsymbol{H}\boldsymbol{x} - \frac{\dot{\lambda}}{\lambda}\boldsymbol{x}. \tag{3}$$

The depth factor λ can be removed by multiplying both sides of (3) with $\hat{\boldsymbol{x}}$. We then obtain the so-called *continuous homography constraint*

$$\hat{\boldsymbol{x}}\boldsymbol{H}\boldsymbol{x} = \hat{\boldsymbol{x}}\boldsymbol{u} \tag{4}$$

since $\hat{\boldsymbol{v}}s\boldsymbol{v} = 0$ for any vector $\boldsymbol{v}$ and scalar s.

B. The extended 4-point algorithm

In order to retrieve $\boldsymbol{H}$, we stack the elements of $\boldsymbol{H}$ into the vector $\boldsymbol{H}^S = [H_{11}, H_{21}, \cdots, H_{33}] \in \mathbb{R}^9$ and rewrite (4) as

$$\boldsymbol{a}^T\boldsymbol{H}^S = \hat{\boldsymbol{x}}\boldsymbol{u} \tag{5}$$

where $\boldsymbol{a} \in \mathbb{R}^{9\times3}$ stands for the Kronecker product $\boldsymbol{x} \otimes \hat{\boldsymbol{x}}$. This allows to stack all the $\boldsymbol{a}_i$ obtained from n tracked features into one cumulative matrix $\boldsymbol{A} = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_n]^T \in \mathbb{R}^{3n\times9}$. Similarly, we stack all $\hat{\boldsymbol{x}}_i\boldsymbol{u}_i$ into a matrix $\boldsymbol{B} = [\hat{\boldsymbol{x}}_1\boldsymbol{u}_1, \cdots, \hat{\boldsymbol{x}}_n\boldsymbol{u}_n]^T \in \mathbb{R}^{3n}$ and generalize (5) to the case of n features:

$$\boldsymbol{A}\boldsymbol{H}^S = \boldsymbol{B}. \tag{6}$$

Using standard methods described in the literature [15], it is possible to recover $\boldsymbol{H}$ from eq. (6).

Since the gyroscopes of any typical onboard IMU directly provide a measurement of the angular velocity $\boldsymbol{\omega}_{IMU}$, we can consider $\boldsymbol{\omega} = \boldsymbol{\omega}_{IMU}$ as *known* from external (i.e., not vision-based) sources. Knowledge of $\boldsymbol{\omega}$ can then be used to derotate the perceived optical flow field. Therefore, we subtract the rotational components from the perceived flow using the interaction matrix which relates $\boldsymbol{u}$ to $(\boldsymbol{v}, \boldsymbol{\omega})$ [16]:

$$\begin{bmatrix} u'_x \\ u'_y \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} - \begin{bmatrix} -x_x x_y & 1+x_x^2 & -x_y \\ -(1+x_y)^2 & x_x x_y & x_x \end{bmatrix}\boldsymbol{\omega}. \tag{7}$$

After this manipulation, the remaining flow components $(\boldsymbol{x}_i, \boldsymbol{u}'_i)$ do not contain any angular velocity component, thus $\boldsymbol{H}$ reduces to

$$\boldsymbol{H} = \frac{1}{d}\boldsymbol{v}\boldsymbol{N}^T. \tag{8}$$

Since $\boldsymbol{N}$ spans $\boldsymbol{H}^T$ and $\|\boldsymbol{N}\| = 1$, we can obtain $\boldsymbol{N}$ from the singular value decomposition $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ as the first column of matrix $\boldsymbol{V}$. The inherent sign ambiguity can be resolved by enforcing $\boldsymbol{N}_z > 0$. Having retrieved $\boldsymbol{N}$, we then obtain $\frac{\boldsymbol{v}}{d}$ as $\frac{\boldsymbol{v}}{d} = \boldsymbol{H}\boldsymbol{N}$.

In order to find a unique solution for $\frac{\boldsymbol{v}}{d}$ and $\boldsymbol{N}$, the observation of at least three feature pairs $(\boldsymbol{x}, \boldsymbol{u})$ is required.

III. SEGMENTATION OF FEATURES

In order to develop a robust velocity estimation approach in presence of a dominant plane, we need a quantitative measure of how well a given plane fits a set of observed features. This also allows to classify new features as inliers or outliers w.r.t. a candidate plane. In the following, we will first propose a quantitative criterium in order to discriminate between features on the plane and outliers. This will be then used to obtain a real-time classification using on-board hardware as the quadrotor explores an unknown environment.

A. Planarity measures

In order to test whether a certain group of observed features belongs to a common plane, we considered two different quantitative measures. We start noting that a prerequisite for solving eq. (6) is that $\boldsymbol{B} \in \mathcal{R}(\boldsymbol{A})$ where $\mathcal{R}(\boldsymbol{A})$ denotes the range space of matrix $\boldsymbol{A}$. This can be restated as

$$\operatorname{rank}(\boldsymbol{A}) = \operatorname{rank}([\boldsymbol{A}\ \boldsymbol{B}]) = 8, \tag{9}$$

since $\operatorname{rank}(\boldsymbol{A}) = 8$ by construction. Let $\sigma_i \geq 0, i = 1 \dots 10$, be the singular values of the augmented matrix $[\boldsymbol{A}\ \boldsymbol{B}] \in \mathbb{R}^{3n \times 10}$ ordered from the largest to the smallest one. As a measure of how well condition (9) is satisfied by the given set of measured features/optical flow, one can monitor the value of σ_9: in fact it is straightforward to check that $\sigma_9 = 0$ if (9) holds, i.e., if all the observed points belong to a common plane, and $\sigma_9 > 0$ otherwise. Therefore, the value of $\sigma_9 \geq 0$ can be exploited as a measure of how well a certain set of features/optical flow meets the planarity constraint.

However, it is also possible to obtain an equivalent information by resorting to a different argument. Let $\boldsymbol{H}^S = \boldsymbol{A}^\dagger \boldsymbol{B}$ represent the least-square solution of the linear system (6): in order to obtain a measure of how well $\boldsymbol{H}^S$ is actually solving system (6), one can consider the 'reprojection' vector $\boldsymbol{R} \in \mathbb{R}^{3n}$

$$\boldsymbol{R} = \boldsymbol{B} - \boldsymbol{A}\boldsymbol{H}^S = \boldsymbol{B} - \boldsymbol{A}\boldsymbol{A}^\dagger \boldsymbol{B} = (\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^\dagger)\boldsymbol{B}$$

which should vanish if (6) admits an exact solution, i.e., if all the observed feature points belong to the same plane. Therefore, another equivalent measure of the planarity assumption besides the previous σ_9 can be taken as $r = \|\boldsymbol{R}\|/n$.

Although yielding similar information, we found in practice that r was much less sensitive to noise issues compared to σ_9. Additionally, the quantity r presents two advantages. First, it does not require intense additional computations since the pseudo-inverse of matrix $\boldsymbol{A}$, i.e., $\boldsymbol{A}^\dagger$, is already computed when solving (6) for recovering the continuous homography matrix. Secondly, this method can be used to test effectively whether a new feature could be added to an already known plane made of other feature points. For this purpose, one can obtain $\boldsymbol{H}^S = \boldsymbol{A}^\dagger \boldsymbol{B}$ using only the features which are already known to form a plane. Then, by constructing new $\boldsymbol{A'}$ and $\boldsymbol{B'}$ out of the new feature point as described in Sect. II-B, one can decide the membership to the given plane by testing $\|\boldsymbol{B'} - \boldsymbol{A'}\boldsymbol{A}^\dagger \boldsymbol{B}\|/n$ against a threshold.

B. Algorithm for robust velocity estimations

Having an effective measure to test both the plane hypothesis and the assignment of new features to existing planes, we can then design an algorithm able to dynamically decide whether an observed feature belongs to the dominant plane. Then, exploiting this clustering, we can obtain a better estimation of the sought scaled linear velocity and plane normal, while the angular velocity is, as explained before, assumed measured from the on-board IMU.

In the following, we will describe in detail the steps run as new images are captured:

- *Initialization.* At the beginning or after the dominant plane was lost, a RANSAC inspired approach is used to pick an initial set of features. The validity of the potential plane is evaluated using the method described above.
- *Update phase.* As a first step, all features already known to be part of the plane are updated to the current location as reported by the flow tracker. Additionally, all those features which have not been observed in the current frame are deleted from the plane.
- *Estimation step.* The scaled linear velocity is estimated together with the plane normal vector. Thus, the estimated linear velocity is independent of the particular plane orientation, allowing for a broader range of environments. This linear velocity is then taken as the best estimation of $\boldsymbol{v}/d$.
- *Integration of new features.* Afterwards, the algorithm tests all other observed features against the current model of the plane as built from known good features using the method explained above. A threshold is used to decide whether a new feature will be added to the feature set.
- *Validation of the feature set.* The performance of each feature is monitored over time. Thus, features which turned out to be outliers will be excluded from the set in this step.

IV. COMPUTATION OF OPTICAL FLOW

We made use of the established implementations provided with OpenCV[1] to process the incoming frames. In particular, we collected an initial set of Shi-Tomasi features [17] tracked over time using the pyramidal version [18] of the Lukas-Kanade tracker [19]. We limited the number of maintained features to 150 in order to limit the computational load of our algorithms. Whenever the size of the feature set dropped below a moving threshold, new Shi-Tomasi features were sampled and added to the set. Since both the floor and the walls of our flight arena are uniformly white, we placed structured carpets on the ground and posters on the walls to provide enough structure. No special lighting conditions were used apart from the default ceiling lights.

A. Interleaved Computation

Our previous study in [11] was carried out on less powerful hardware reaching a frame rate for image acquisition and processing of only 17 Hz. In that work, we were computing the optical flow by means of subtraction of feature locations on any two consecutive frames scaled by the elapsed time. The current hardware however, as described in Sect. V, supports considerably higher frame rates. Since efficient feature tracking is always carried out in pixel space only, a certain rounding error to the next pixel coordinate cannot be avoided. An increased frame rate, however, results in the

[1] opencv.willowgarage.com

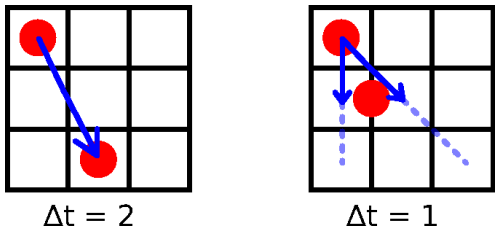


Fig. 1: The influence of high frame rates on the quality of the computed optic flow. An observation made in a short time frame will be subject to more noise than an observation which was made over an extended period of time. The center of a feature might be located on the edge of a pixel and thus will be assigned to one of the adjacent pixels. Clearly, the influence of this rounding error increases as the vector shortens due to higher sampling rates.

observation of even shorter optical flow vectors prone to an highly increased rounding error as explained in Fig. 1.

While an increased resolution of the sensor or sub-pixel refinement might reduce this problem at the cost of an increased computational load, an extended temporal baseline between the compared feature sets allows for the observation of longer optical flow vectors. This can be done in an interleaved manner in order to still make use of the benefits of an increased update rate. Thus, we developed a method to extract the optical flow over a predefined baseline of length d while the best possible frame rate is still maintained. Therefore, features are tracked from one frame to the next as they are acquired, but the resulting feature locations for each frame are stored in an array together with a time stamp. Afterwards, the frame with a time stamp closest to $t_{now} - d$ is used for the computation of the optical flow relative to the most recent image. The parameter d can be dynamically selected depending on the current vehicle speed or other practical considerations.

V. EXPERIMENTS

A. Experimental setup

For our experiments, we used a quadrotor purchased from MikroKopter.de[2] with a customized low level controller. The vehicle was equipped with a small computer holding an Intel Atom 1.8 GHz dual core CPU and running 11.10 Ubuntu Server. ROS (Robot Operating System) formed a middle layer to allow interprocess communication. The visual input was provided through a MatrixVision[3] mvBlueFOX MLC200w monochrome 752×480 camera. The installed $78°$ lens projected an effective field of view of $54° \times 38°$ onto the $1/3''$ sensor. In order to save computation time, the calibration of the visual system was done before the experiments using the Camera Calibration Toolbox for Matlab[4]. We numerically calculated a lookup table that allows the mapping of each pixel on the image plane to the corresponding 3D coordinate $\boldsymbol{x} = [x, y, 1]^T$ in camera frame.

[2] www.MikroKopter.de
[3] www.matrix-vision.com
[4] www.vision.caltech.edu/bouguetj/calib_doc/

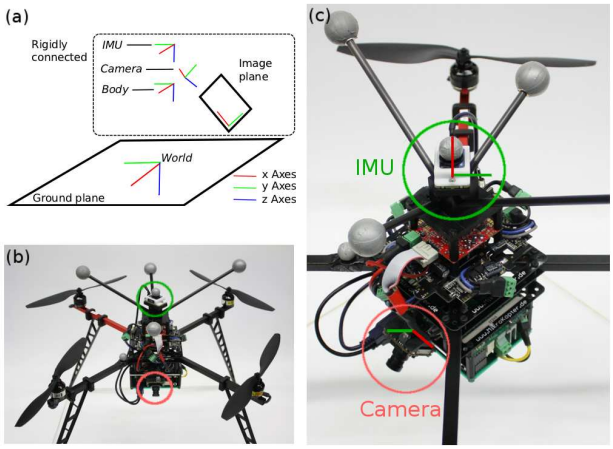


Fig. 2: Experimental setup: (a) Chart of the different frames in consideration for the proposed algorithm. (b, c) The quadrotor in its final configuration with IMU and camera being highlighted. The body frame of the quadrotor is fixed in the center of the two crossbars holding the motors. The x-axis equals the red arm. The IMU is aligned with the body frame while the camera is rotated around the z-axis and tilted downwards by $45°$ with respect to these frames. The image plane is fixed in $1\ m$ distance to the camera. All frames follow the NED convention commonly used in aviation.

As for the onboard IMU, we used a MicroStrain[5] 3DM-GX3 IMU at 200 Hz.

Figures 2(b) and 2(c) show the flight configuration used in the experiments, while Fig. 2(a) visualizes the locations of the sensor frames used in our setup. All measurements were always converted into the camera frame for the computations.

To obtain an accurate ground truth within our flight arena of $6 \times 8 \times 3$ m, a Vicon[6] setup consisting of six Bonita cameras was used. The linear velocity calculated from the tracking data was filtered using a low-pass filter with a cut-off frequency at 10 Hz. As common with quadrotors, we measured a strong rotational vibration around the z axis of the body. Thus, we applied a 10 Hz low-pass filter on the yaw readings of the gyroscopes as well.

B. Summary of the experiments

In order to evaluate the robustness of the proposed algorithm, we conducted experiments using a regular quadrotor UAV. The aim of these experiments was the comparison of our feature filtering approach in the presence of obstacles with our previously presented method which was based on the assumption of a planar environment [11]. Since the onboard hardware cannot run both algorithms in parallel for comparison on the same sensory input while controlling the robot, we recorded IMU, Vicon and video stream in a first session. To allow for an evaluation of both systems on the same trajectory and to obtain a ground truth, we controlled the vehicle using feedback from the Vicon tracking system during these recordings. Afterwards, the quadrotor was

[5] www.microstrain.com
[6] www.vicon.com

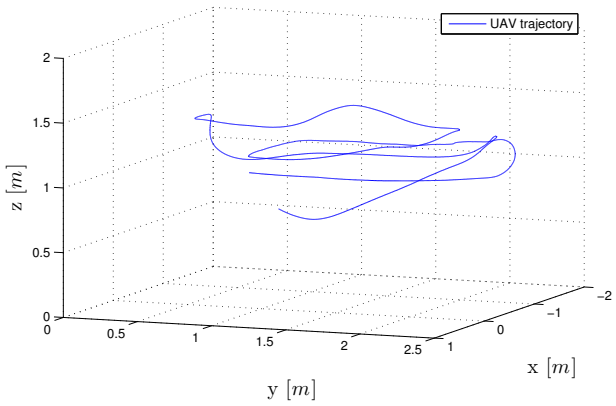


Fig. 3: A section of the trajectory which was used for the collection of the presented results. The section shown here was recorded during second 20 and second 50 of the experiment and extends 2.2 m in x, 2.1 m in y and 1.2 m in the z dimension.

commanded to track the same trajectory in space in order to estimate the scaled linear velocity $\boldsymbol{v}/d$ using the two algorithms. For the trajectory partly shown in Fig. 3, we took off from a flat horizontal floor and moved towards a table placed on the right side of the room. The UAV was then commanded to explore the room in which two more objects were placed: a small ladder and a stool. The three objects were covered with similar patters as the floor and a back wall to increase the chance of positive feature extraction.

Finally, we closed the loop and commanded the UAV while relying purely on on-board hardware.

VI. RESULTS AND DISCUSSION

For the evaluation of the experiment, we consider a total flight time of 120 s: Fig. 4(a-c) report the recovered scaled linear velocity for a duration of 30 s. Each of the three plots shows a comparison between: (i) the unmodified algorithm which, for each image frame, considers all observed flow vectors for the estimation of the linear velocities, (ii) the presented clustering approach, and (iii) the ground truth obtained from the Vicon tracking system. For the reader's convenience, the obtained estimations of the scaled velocity were multiplied by the current distance d from the dominant plane as shown in Fig. 5.

The error relative to the ground truth for both system during the 30 s section of the experiment is presented in Fig. 6. Computing the mean error over the entire length of the experiment, we find that our former approach reaches an accuracy of $0.111\ \frac{m}{s}$, while the proposed extended solution yields an error of $0.089\ \frac{m}{s}$. Thus, the proposed approach was able to improve the estimation by a factor of 25%. The standard deviation improved slightly from an initial $0.087\ \frac{m}{s}$ to $0.081\ \frac{m}{s}$ when using the modified algorithm.

Figures 7–8 present the clustering performance in two different scenes. In the case of Fig. 7, our algorithm is able to reject those features located on a table of $0.75\ m$ height and a board close by. Similarly, in the scene of Fig. 8, the system excludes features on the back wall and two irregular shaped

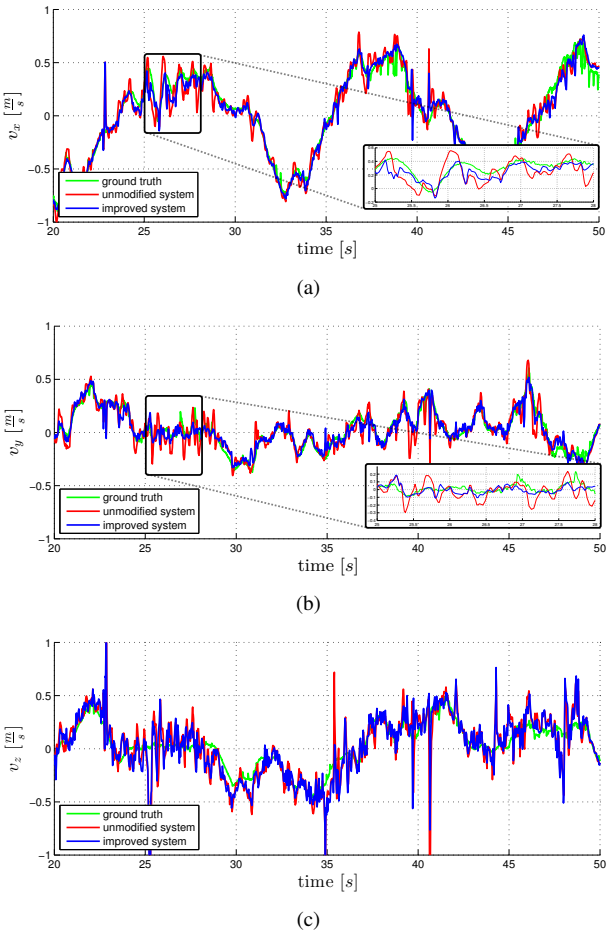


Fig. 4: Estimated linear velocities in camera frame along the (a) x, (b) y and (c) z axis for both the unmodified and the improved algorithm together with a ground truth as obtained from a Vicon system. The unmodified approach considers all available features for the computation of the velocity while the presented modification is able to remove features which are not on the dominant plane.

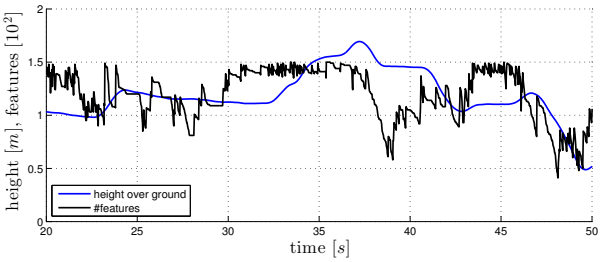


Fig. 5: Height of the UAV over ground with the number of features used.

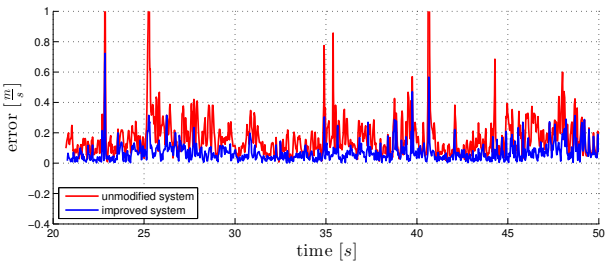
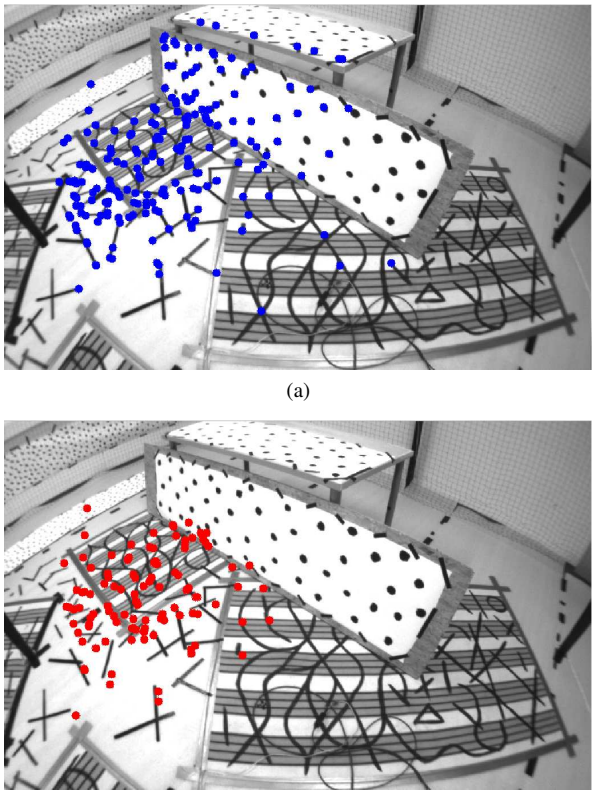


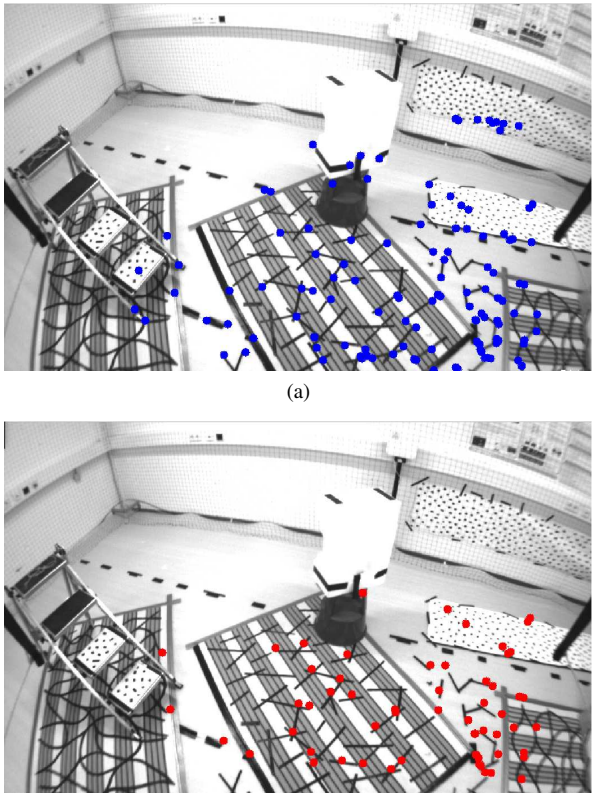
Fig. 6: Error of the original and improved approach as compared to the ground truth. The improvement is mainly due to an enhanced velocity estimation in the horizontal plane while the contribution of the z domain has a relatively low influence (compare Fig. 4).



(a)

(b)

Fig. 7: Direct comparison of (a) original and (b) improved algorithm. While the unmodified system considers all observed features, the presented approach is able to exclude the features on the table and the upper region of the plate which leans against the table. Note that the image was rotated by 180° for the readers convenience since the camera had to be mounted upside down for mechanical reasons (compare Fig. 2(a)). Furthermore, these images were taken using a 140° lens to improve the readers impression of the scene.



(a)

(b)

Fig. 8: Another direct comparison of (a) original and (b) improved algorithm. This example demonstrates the ability of our system to filter the features on the back wall and, apart from one false-positive classification, the two objects.

objects. The algorithm was mainly designed to avoid a false-positive classification of outliers and therefore the addition of bad features to the dominant plane. Consequently, a relatively high false-negative classification rate was accepted. In average, 20% of all features were outliers. Out of these, our system rejected 94% successfully.

During the final experiment in closed-loop control, an average frame rate of 31 Hz guaranteed a stable flight behavior.

VII. CONCLUSIONS AND FUTURE WORK

A. *Conclusions*

This work presented an algorithm able to estimate the self-motion of a quadrotor UAV using a monocular camera and relying on sole on-board hardware. For this, we developed an optical flow tracking system for high frame rates and extended a real-time velocity estimation algorithm based on the continuous homography constraint to detect outliers within the optical flow. The feature rejection system was shown to improve the estimation of the linear velocity by 25% and to reduce the standard deviation.

Our approach proved to be an effective solution as it does not rely on prior maps or on keeping visibility of specific features. Although requiring presence of a dominant plane, we were able to reject those features not belonging to this

plane with a very high accuracy, and consequently to improve the real-time velocity estimations for previously unknown environments.

B. Future work

Currently, we are addressing several limitations of our system which emerged during the validation phase. First, during the optical flow tracking, the features are not necessarily distributed evenly across the image plane. In fact, some areas attract features more than others and thus yield an unbalanced representation of the scene. This can become critical since the number of tracked features may be limited for coping with the available on-board hardware. Thus, we plan to split the image plane into tiles and try to keep the number of features on each tile balanced. To further speed up the feature sampling, the last velocity estimate could be used to understand in which direction the features moved. Consequently, the feature sampling process can be concentrated on the new areas of the image.

Additionally, using the same mathematical approach to test the validity of a single plane, it should be possible to decide on the validity of the entire assumption that all features are located on a plane as well. Thus, in cluttered indoor environments and urban outdoor scenes, the algorithm could automatically decide between the presented system and another solution not based on the assumption of piecewise planar scene as, e.g., the continuous eight-point algorithm [15].

VIII. ACKNOWLEDGMENTS

The authors like to thank Dr. Antonio Franchi and Martin Riedel for their valuable suggestions and contribution on the development of the underlying software framework.

This research was partly supported by WCU (World Class University) program funded by the Ministry of Education, Science and Technology through the National Research Foundation of Korea (R31-10008).

REFERENCES

[1] EU Collaborative Project ICT-231855, "sFly," www.sfly.org/.

[2] EU Collaborative Project ICT-248669, "AIRobots," www.airobots.eu.

[3] EU Collaborative Project ICT-287617, "ARCAS," www.arcas-project.eu.

[4] K. E. Wenzel, A. Masselli, and A. Zell, "Automatic Take Off , Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 221–238, 2010.

[5] W. Li, T. Zhang, and K. Kühnlenz, "A Vision-Guided Autonomous Quadrotor in An Air-Ground Multi-Robot System," in *Proceedings of the International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 2980–2985.

[6] S. Weiss, M. Achtelik, L. Kneip, D. Scaramuzza, and R. Siegwart, "Intuitive 3D Maps for MAV Terrain Exploration and Obstacle Avoidance," *Journal of Intelligent & Robotic Systems*, vol. 61, pp. 473–493, Nov. 2011.

[7] R. A. Newcombe and A. J. Davison, "Live Dense Reconstruction with a Single Moving Camera," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, San Francisco, CA , USA, 2010, pp. 1498–1505.

[8] R. Mahony, F. Schill, P. Corke, and Y. S. Oh, "A new framework for force feedback teleoperation of robotic vehicles based on optical flow," in *Proceedings of the International Conference on Robotics and Automation*, Kobe, Japan, May 2009, pp. 1079–1085.

[9] L. R. García Carrillo, A. Dzul, R. Lozano, and C. Pégard, "Combining Stereo Vision and Inertial Navigation System for a Quad-Rotor UAV," *Journal of Intelligent & Robotic Systems*, 2011.

[10] F. Kendoul, I. Fantoni, and K. Nonami, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," *Robotics and Autonomous Systems*, vol. 57, no. 6-7, pp. 591–602, June 2009.

[11] V. Grabe, Heinrich H. Bülthoff, and P. R. Giordano, "On-board Velocity Estimation and Closed-loop Control of a Quadrotor UAV based on Optical Flow," in *Proceedings of the International Conference on Robotics and Automation*, St. Paul, MN, USA.

[12] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *Proceedings of the International Symposium on Mixed and Augmented Reality*, Nara, Japan, Nov. 2007, pp. 225–234.

[13] R. O. Castle, G. Klein, and D. W. Murray, "Video-rate Localization in Multiple Maps for Wearable Augmented Reality," in *Proceedings of the International Symposium on Wearable Computers*, Pittsburgh, PA, USA, 2008, pp. 15–22.

[14] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision Based MAV Navigation in Unknown and Unstructured Environments," in *Proceedings of the International Conference on Robotics and Automation*, Anchorage, AK, USA, 2010, pp. 21–28.

[15] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision*. Springer, 2004.

[16] F. Chaumette and S. Hutchinson, "Visual Servo Control, Part I. Basic Approaches," *Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[17] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, no. June, Seattle, WA, USA, 1994, pp. 593–600.

[18] J.-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm," *In Practice*, vol. 1, no. 2, pp. 1–9, 1999.

[19] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Vancouver, BC, Canada, 1981, pp. 674–679.