2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Localization and stabilization of micro aerial vehicles based on visual features tracking

Jan Chudoba, Martin Saska, Tomáš Báča, Libor Přeučil

Abstract— **This article presents a method for long-term autonomous micro-aerial vehicle (MAV) localization and position stabilization. The proposed method extends MAV proprietary stabilization based on inertial sensor or optical flow processing, without use of an external positioning system. The method extracts visual features from the images captured by a down-looking camera mounted under the MAV and matching these to previously observed features. Due to its precision and reliability, the method is well suited for stabilization of MAVs acting in closely cooperating compact teams with small mutual distances between team members. Performance of the proposed method is demonstrated by experiments on a quad-copter equipped with all necessary sensors and computers for the autonomous operation.**

I. Introduction

A control and stabilization system designed for Micro Aerial Vehicles (MAV) is presented in this paper. The proposed approach is based on localization by matching visual features in subsequent images captured by on-board down-looking camera. The localization method is used for MAV position stabilization and for navigation in metric coordinate frame, providing long-term robustness. The designed method is independent on external localization methods or sensors (e.g. GPS or Vicon), which predestinates its utilization in environment, where these systems are not available. In addition, the open access systems (GPS) offer about 10 meter accuracy with commonly available receivers, which could be even worse near buildings or other obstacles blocking view to satellites. This accuracy is not sufficient for stabilization of multi-MAV systems closely cooperating together in the same operational space, where a localization with precision in units to tenths of centimeters is required.

MAV inertial measurement unit provides sufficiently precise sensory input for the low-level stabilization of the system, however they are not able to hold MAV on desired position or trajectory for longer time due to sensor drift and external disturbances. The proposed approach combines an exteroceptive sensor measurement of distances or displacement relative to ground frame. It stabilizes the MAV with positional drift bound in an allowed tolerance, which is not dependent on flight duration. Besides, there is no need for any prior environment knowledge or mapping.

The proposed navigation system estimates MAV position changes by comparison of subsequent camera images of ground and by estimating the transformation between these images as the vehicle moves. The results of this transformation (the estimated MAV drift) is then used in feedback to control the vehicle position (to compensate the drift if the MAV is hovering). For robust and effective transformation estimation, the matching of image features, which are invariant to scale and rotation, is used. From the available image features, the speeded-up robust features (SURF, [2]) are used in the experiments described in this article. SURF technique provides good matching robustness compared to computational efficiency, but most of the image features that are used nowadays may be applied in this method with sufficient result.

There are two possible approaches when camera image is used for MAV position estimation: visual odometry and visual SLAM (simultaneous localization and mapping). Classical visual odometry estimates current position by integrating position differences between two consecutive images, resulting in permanent error cumulation. Such approaches were used in e.g. in [1] or [6]. In the [5] the PX4Flow optical flow sensor is used to estimate MAV trajectory.

SLAM based approach builds a map in defined frame of reference and estimates positional difference between camera image and the map. Although this approach has principal SLAM problem with error accumulation during the movement, a position error in each place is bounded independently on time as long as the map remains consistent. When MAV moves from an origin place to some destination and then back to the origin, the position error of reaching origin location is bounded, assuming the vehicle follows previously learned trajectory. When the trajectory forms a loop through a previously unknown environment, the loop closing algorithm is necessary to assure map consistence.

In [8], Krajník at al. use SURF features to navigate a MAV along the previously learned trajectory using a single forward-looking camera. In [4], the KLT algorithm is used to track corner features in down-looking camera image to estimate helicopter displacement. Additionally, they apply image registration of camera images to global geo-referenced visual map to achieve localization in global coordinates. Our method uses a SLAM approach to localize MAV without a necessity of external equipment or an a-priory area map availability.

II. Algorithm description

The basic idea of the proposed MAV visual localization lies in an estimation of parameters of geometric transformation between subsequent images captured from the onboard down-looking camera. In the theoretical description of the

[1]The authors are with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague. chudoba@labe.felk.cvut.cz, saskam1@fel.cvut.cz, bacatoma@fel.cvut.cz, preucil@labe.felk.cvut.cz

978-1-4799-2376-2/14/$31.00 ©2014 IEEE

method, we assume a flat and horizontal surface under the MAV, but the method is robust enough to deal with terrain undulations. The stabilization of MAV flying above a rough terrain is achieved as shown in the presented outdoor experiments, but its localization (the visual odometry) may be affected by an additional cumulative localization error.

In addition to the transformation between subsequent images, the transformation between current image and a visual map, which is built during the flight from previous observations, is estimated for achievement of long-term stability and robustness. In the mapping part of the algorithm, the observed scale-invariant features are stored in a map. The map of features called $\mathbf{F_m}$ defines the localization frame of reference. Each feature has a 2-dimensional position in a map, which is obtained as the projection of its transformed image coordinates. Additionally, two numbers are attached to each feature for discarding useless features from map: 1) number of repeated observations and 2) time of the first observation.

In the proposed approach, each single camera image is processed as described in this section. The output of the algorithm is the actual map $\mathbf{F_m}$ and the transformation matrix $\mathbf{T}$ in form

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}, \tag{1}$$

where $\mathbf{R}$ is a matrix representing rotation and scale and $\mathbf{t}$ is a translation vector. When new image from camera is captured, the set of features $\mathbf{F_i}$ is detected in the image and feature descriptors are computed. These features are matched to features stored in the map $\mathbf{F_m}$. We expect that position change between two consecutive images is bounded and therefore all matches can be validated using the last known transformation. In the validity test, the matches with the projection of image feature farther from the matched feature in the map than a adaptive threshold α_1 are discarded. The α_1 threshold is calculated as average projection error over all matches multiplied by constant 3. This mechanism effectively helps to eliminate many accidental wrong matches and so improves method robustness.

The transformation of displacement between images (and consequently displacement in MAV position) is computed from the matching set, which contains two sets of image features $\mathbf{f_i}$ and map features $\mathbf{f_m}$. The proposed method is based on corresponding point set registration algorithm described in [3]. First, centroids of each set ($\mathbf{c_i}$ and $\mathbf{c_m}$) is computed as

$$\mathbf{c_i} = \frac{1}{N}\sum_{n}^{N} {\mathbf{F_i}}^n, \quad \mathbf{c_m} = \frac{1}{N}\sum_{n}^{N} {\mathbf{F_m}}^n, \tag{2}$$

where N is number of matches.

For estimation of the rotation between two feature sets, the feature image coordinates $\mathbf{F_i}$ and $\mathbf{F_m}$ are translated by values $\mathbf{c_i}$ and $\mathbf{c_m}$, respectively (to have centroids of both sets in origin). The rotation matrix is then calculated from a singular value decomposition of matrix

$$\mathbf{H} = \sum_{n}^{N} ({\mathbf{F_i}}^n - \mathbf{c_i}) \cdot ({\mathbf{F_m}}^n - \mathbf{c_m})^T, \tag{3}$$

where ${\mathbf{F_i}}^n$ and ${\mathbf{F_m}}^n$ are column vectors of image coordinates of feature in image or map respectively, as

$$[\mathbf{u}, \mathbf{s}, \mathbf{v}] = \mathrm{svd}(\mathbf{H}), \tag{4}$$

$$\mathbf{R} = \mathbf{v}^T \cdot \mathbf{u}. \tag{5}$$

The first row of R is multiplied by -1 if $\det(\mathbf{R}) < 0$. The scale of the transformation is estimated as a ratio between average distances of features from origin in each set:

$$scale = \frac{\sum_n^N {\mathbf{F_m}}^n}{\sum_n^N {\mathbf{F_i}}^n}. \tag{6}$$

The translation component $\mathbf{t}$ is computed as a difference between position of the centroid $\mathbf{c_m}$ of the set of features $\mathbf{F_m}$ and the centroid $\mathbf{c_i}$ of the set of image features $\mathbf{F_m}$, which is transformed by the obtained rotation matrix:

$$\mathbf{t} = \mathbf{c_m} - scale \cdot \mathbf{R} \cdot \mathbf{c_i}. \tag{7}$$

The resulting transformation matrix is a compound of the calculated rotation, scale and translation components.

The obtained transformation is used for projection of the detected image feature points into the map. Distance from matching map features is used for detection of outlier matches and removing them from the set of stored matches. The outlier matches are removed from the set of found matches and the transformation is re-calculated. At least 3 matches are required for sufficient transformation estimation. If the transformation is considered as valid, the remaining features from $\mathbf{F_i}$, which were not matched with any feature stored in previous iterations of the algorithm, are projected into map coordinates and added to the map for possible matching in future iterations. The validity test depends on the number of valid matches and re-projection error. Accordingly, older map features, which were not matched in few consecutive images, are removed from the map to prevent unnecessary growth of map size.

The estimated transformation describes MAV position in a map in camera image coordinates relative to its initial position and height above surface. The $scale$ has a meaning of ratio between the actual height above the ground and the initial height. Therefore, MAV position in metric units relative to initial position may be computed based on the known camera image calibration parameters and MAV initial height.

The functionality of the above described algorithm is conditioned by a requirement on zero MAV inclination, which means that the camera optical axis has to be perpendicular to the surface below the MAV. This strong requirement can not be satisfied if using moving MAVs and therefore the effect of inclination has to be compensated. The inclination by angle

α in one axis of MAV hovering in height h causes estimated position error

$$\Delta x = h \cdot \tan\alpha. \quad (8)$$

This distance error can be easily subtracted from the estimated position if we suppose the inclination values are known from the MAV inertial sensors.

The algorithm employed for updating the transformation matrix by the processing of the captured image, which is described in this section, is summarized in listing 1.

Listing 1: Image processing algorithm

```
1  find a complete set of image features F_i
2  calculate feature descriptors for all features
       in F_i
3  find the best match f_m ∈ F_m for each f_i ∈ F_i,
       and add (f_i, f_m) to set M
4  calculate error between f_m and projection f_i
       using the last known transformation T for
       each match (f_i, f_m) from M
5  discard matches from M with error greater than
       adaptive threshold α_1
6  compute new transformation T from sets of
       matches from M
7  re-project features from M using T and
       discard matches with error greater than
       fixed threshold α_2 from M
8  if some outlier matches were discarded, re-
       calculate transformation T
9  project all remaining features from F_i, which
       were not matched with any feature from F_m,
       using T, and store them in the map F_m
10 discard features from map F_m having no match
       in consecutive images
```

III. System description

A. Visual localization module

The MAV localization based on image features detection is computed on Intel Atom CPU 330 at 1.6GHz with integrated Nvidia ION VGA and CUDA platform. For the experiments, CMOS VGA camera with 1/3" chip MT9V032 was used, field of view $61° \times 41°$, configured for 640×480 pixels resolution.

The GPU implementation of SURF algorithm [2] was used to detect features in camera images. The feature detection algorithm hessian threshold is adaptively adjusted in order to obtain about 100 features in each image depending on the surface under the MAV. The image processing delay is approximately 250 ms. The position update rate is about 2 updates per second, in case that a transformation is found.

B. Experimental micro-aerial vehicle

An experimental quad-copter Mikrokopter L4-ME (see figure 1) was used for the verification of the proposed stabilization/localization method. The MAV equipped with above mentioned sensors and computers is able to fly about 5–6 minutes, until battery is discharged.

This quad-copter, equipped with FlightCtrl flight control board provides basic MAV stabilization using its inertial sensors. However, this own stabilization naturally suffers by a drift and consequently an additional stabilization system is necessary. We equipped the Mikrokopter MAV with the

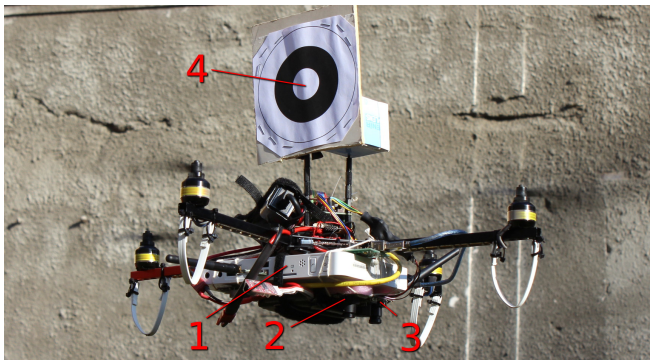


Fig. 1: Mikrokopter equipped with Intel Atom PC (1), localization camera (2), PX4Flow sensor (3) and visual pattern (4) for reference position estimation.

PX4Flow optical flow sensor [5], pointing its camera downwards.

PX4Flow sensor provides optical flow information (proportional to horizontal velocity) in rates over 100 Hz (max. 250 Hz in good light conditions), which makes it sufficient for the MAV control. Additionally, the PX4Flow sensor is able to measure height above the ground using integrated sonar, which allows us to control MAV altitude.

Feedback control using this sensor enhanced the stabilization significantly, allowing steady hovering for durations of units to tens of seconds, depending on light conditions and ground surface properties. Since the PX4Flow sensor estimates actual velocity of the MAV, the position control is not able to guarantee robust position stabilization in the long term.

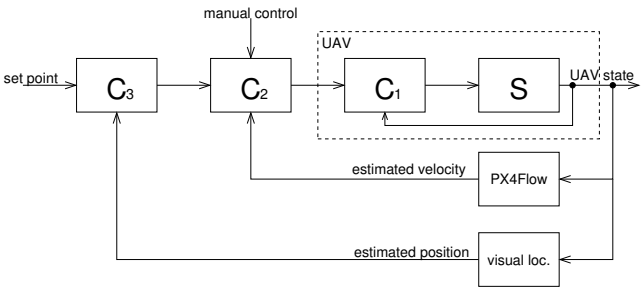


Fig. 2: MAV controller scheme

The overall control scheme is depicted in figure 2. The block (S) stands for quad-copter, where inputs are requested propeller velocities and output is full state of the MAV (3D position, velocity and acceleration). There are basically three nested control loops. Most inner control loop is provided by a proprietary FlightCtrl board (C_1), providing mainly MAV tilt control. Input to the C_1 controller are desired tilt angles (proportional to desired horizontal velocities) and desired vertical velocity. Second control loop provides velocity control by use of PX4Flow (C_2) and proportional-derivative controller. Third control loop includes designed visual feature stabilization method to robustly stabilize MAV position in long term by a controller (C_3). C_3 is realized as a

PID controller. The possibility to measure real displacement of MAV in ground-attached coordinate frame also allows to navigate MAV to requested position with final error dependent only on distance from the starting position.

IV. Experiments

A. *Long-term stability*

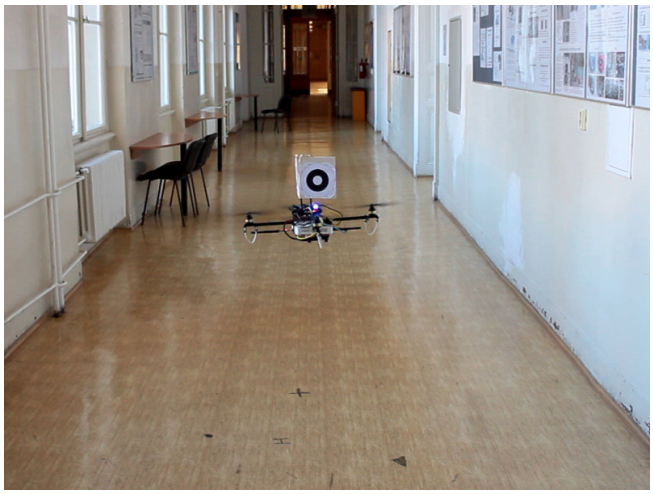


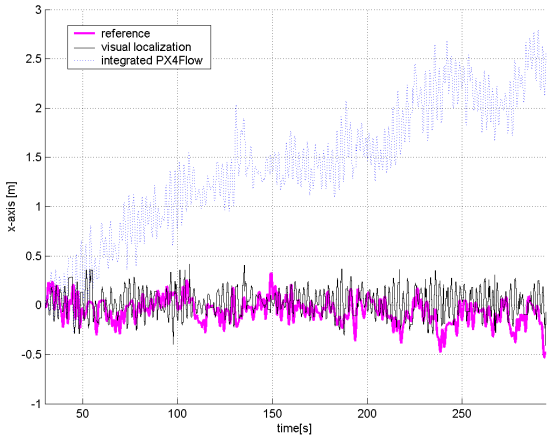
Fig. 3: MAV hovering stabilization experiment.

The proposed method is employed to control and stabilize position of MAV while hovering in the experiment (see image 3). External fixed camera was used to obtain a reference position. Beside the reference position and position estimated by the visual feature localization method, we further show integrated data from the PX4Flow optical flow sensor to prove localization robustness improvement. Visual localization data are used to stabilize MAV position by loop-back control during the experiment.

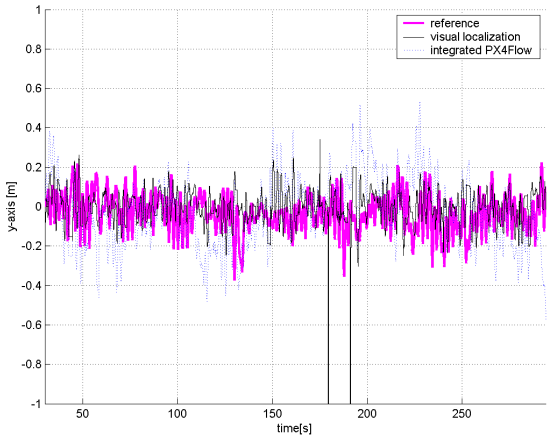
An information about MAV inclination angles is not used in localization during this experiments, which causes notable oscillations of estimated position and consequently oscillations in real MAV position as a result of loop-back control. This may cause small differences in reference and estimated positions (visible mainly in time domain), however the result of the experiment is not affected as we want to prove that average position error is bounded.

For the reference position measurement the visual method of tracking ring pattern in image from external fixed camera was used. The description of the tracking method can be found in [7].

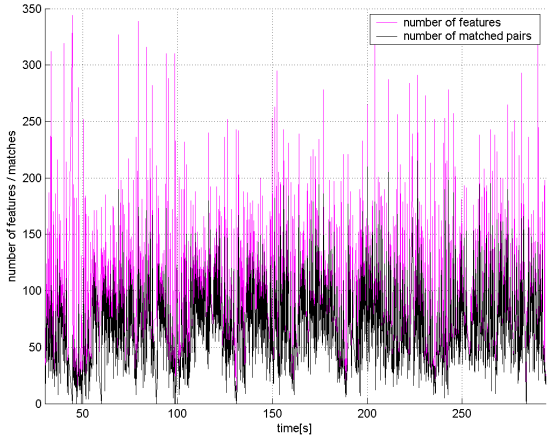
Figure 4 show reference position, position integrated from PX4Flow measured velocities and visual feature method position. The duration of experiment is about 270 seconds, which limited by MAV battery capacity with the actual payload. It is obvious that after few minutes the PX4Flow integrated position estimate drifts from the real position as expected (observed mainly in x-axis in this experiment). Average value of visual feature position estimator remains bounded. Figure 4c shows number of detected and matched features during the experiment.



(a)



(b)



(c)

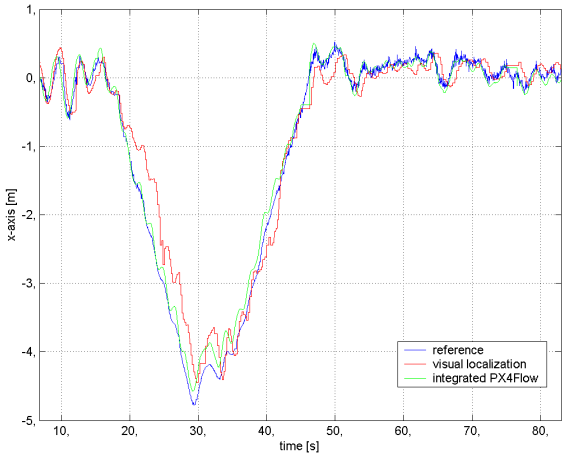
Fig. 4: MAV hovering stabilization in x (a) and y (b) axes. Blue line – reference, green – integrated PX4Flow, red – visual localization. (c) Number of detected features and matched pairs.

B. Navigation experiment

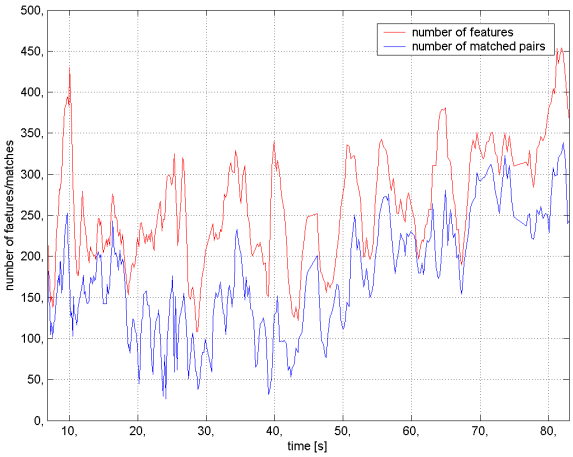
(a)

(b)

Fig. 5: MAV navigation by visual feature localization experiment.



(a)



(b)

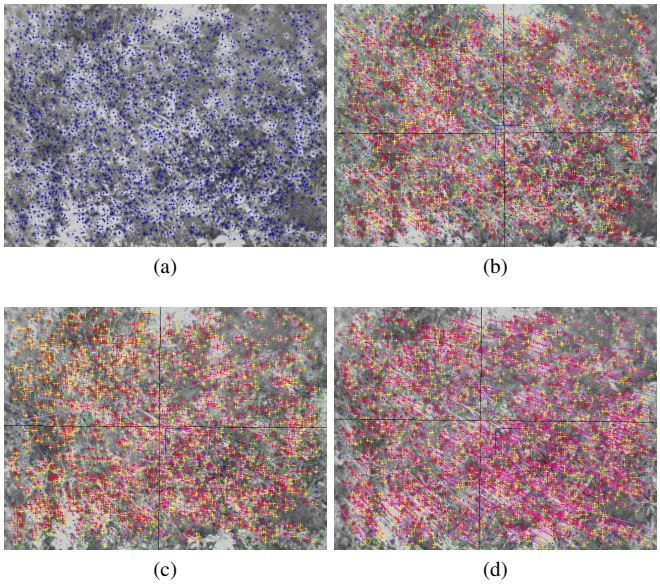
Fig. 6: MAV navigation by visual feature localization – x-axis. Blue line – reference, green – integrated PX4Flow, red – visual localization.

During this experiment a desired position set point is changed to prove ability of localization method to estimate position while an MAV travels through the previously unknown environment. When the distant position is reached (image 5b), the set point is changed back to the original location (image 5a), so the MAV performs navigation along the previously learned path. Position control is based on the estimated position from visual localization.

Figure 6a shows estimated position compared to externally measured position in one axis. Second axis is not plotted, because the reference visual localization does not provide sufficient precision in this direction. There is a notable position error in far set point (about 4 meters from origin position), compared to error near trajectory origin. This error is an consequence of SLAM nature of the localization method. However, an estimated position of MAV after return to the original location varies about ± 0.2m, which lays in the expected method precision tolerance (including achievable precision of position control). It is evident, that method is able to navigate MAV to previously known location with error bounded by this method precision, when navigating to location near previously traveled trajectory. Figure 6b shows number of detected features and number of matched features during the experiment. Figures 7 show feature matching during initialization and first steps of the method. In the figure 8 a situation of return to the origin is shown. Lines in the images connect matching feature points in current and first (reference) image.

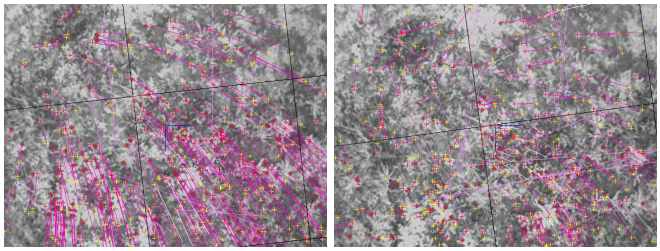
V. Conclusion

In this article the method for MAV position localization and stabilization, based on visual feature matching, is presented. The method is compared to a method based on optical flow integration using data from the PX4Flow sensor. Performed experiments proved that the method is able to stabilize the MAV on fixed place in long-term operation, in comparison to the inertial or optical flow stabilization methods that suffer from accumulative positioning error.



(a) (b) (c) (d)

Fig. 7: Image processing – method initialization.



(a) (b)

Fig. 8: Image processing – return to original position.

The localization method has natural limitations caused by necessity to detect distinguishable visual features on a ground under the MAV. The method fails if the visual properties of the ground are uniform, which means that it does not offer sufficient number of features. Another possible problem is caused by similar repeating patterns on the ground, which may result in incorrect feature matching. However, such situation occurs usually only if the MAV is operating in structured urban environments, specifically indoors. In outdoor scenarios the operational conditions of this method are in most cases satisfied, mainly if the MAV is flying in a higher altitude.

Another problem, which requires especial investigation and improvement of the basic method, were observed in MAV movement from "feature-rich" to "feature-poor" environments. The most features are detected in richer part of the environment, since the adaptive feature threshold is used to limit the number of detected features with the aim of decreasing the computational requirements. Consequently, insufficient number of matches can be found in the "feature-poor" parts, where the MAV is flying to. In the worst case, it may cause a higher position estimation error.

There is a computational limitation caused by available hardware for the visual feature detection and tracking, which limits the image processing to about 4-5 frames per second. Due to this limitation, another faster control loop needs to be employed to locally stabilize the MAV. In the presented experiments, we used underlying controller with the PX4Flow sensor to provide this sufficient stabilization. Nevertheless, utilization of a specialized hardware (e.g. FPGA) for computation of the computational expensive features extraction speeds up the image processing, and the method becomes sufficient also for the low-level stabilization of MAV with only common inertial sensors.

In the current implementation of the proposed method, navigation of the MAV into a previously known place along the unknown path is not supposed. This situation may cause a position mismatch and localization map corruption, which is well known problem in SLAM. In the future development, the loop-closing algorithm needs to be integrated to the method to deal with this issue.

References

[1] Omead Amidi. *An Autonomous Vision-Guided Helicopter*. PhD thesis, Carnegie Mellon University, Department of Electrical and Computer Engineering, Pittsburgh, PA 15213, August 2006.

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[3] P.J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 14:239–256, Feb 1992.

[4] Gianpaolo Conte and Patrick Doherty. Vision-based unmanned aerial vehicle navigation using geo-referenced information. *EURASIP Journal on Advances in Signal Processing, Special section p1*, 2009, 2009.

[5] Dominik Honegger, Lorenz Meier, Petri Tanskanen, and Pollefeys Marc. An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications. In *IEEE International Conference on Robotics and Automation*, Karlsruhe, 2013.

[6] Jonathan Kelly, Srikanth Saripalli, and GauravS. Sukhatme. Combined visual and inertial navigation for an unmanned aerial vehicle. In Christian Laugier and Roland Siegwart, editors, *Field and Service Robotics*, volume 42 of *Springer Tracts in Advanced Robotics*, pages 255–264. Springer Berlin Heidelberg, 2008.

[7] T. Krajník, M. Nitsche, J. Faigl, T. Duckett, M. Mejail, and L. Přeučil. External Localization System for Mobile Robotics. In *Proceedings of the International Conference on Advanced Robotics*, Montevideo, 2013. IEEE.

[8] T. Krajník, M. Nitsche, S. Pedre, L. Přeučil, and M. Mejail. A Simple Visual Navigation System for an UAV. In *International Multi-Conference on Systems, Signals and Devices*, page 34, Piscataway, 2012. IEEE.