TU Delft

Delft University of Technology

# Self-supervised monocular distance learning on a lightweight micro air vehicle

Lamers, K.; Tijmons, Sjoerd; de Wagter, Christophe; de Croon, Guido

**DOI**
10.1109/IROS.2016.7759284

**Publication date**
2016

**Document Version**
Accepted author manuscript

**Published in**
IROS 2016

**Citation (APA)**
Lamers, K., Tijmons, S., de Wagter, C., & de Croon, G. (2016). Self-supervised monocular distance learning on a lightweight micro air vehicle. In *IROS 2016: 2016 IEEE/RSJ International Conference*
https://doi.org/10.1109/IROS.2016.7759284

**Important note**
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

**Copyright**
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

*This work is downloaded from Delft University of Technology.*
*For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.*

# Self-Supervised Monocular Distance Learning on a Lightweight Micro Air Vehicle

Kevin Lamers, Sjoerd Tijmons, Christophe De Wagter, Guido de Croon, *Member, IEEE*

***Abstract*— Obstacle detection by monocular vision is challenging because a single camera does not provide a direct measure for absolute distances to objects. A self-supervised learning approach is proposed that combines a camera and a very small short-range proximity sensor to find the relation between the appearance of objects in camera images and their corresponding distances. The method is efficient enough to run real time on a small camera system that can be carried onboard a lightweight MAV of $19\ g$. The effectiveness of the method is demonstrated by computer simulations and by experiments with the real platform in flight.**

## I. Introduction

Very small and lightweight Micro Air Vehicles (MAVs) can play an important role in many useful applications where size is important. Example applications are the inspection of difficult to reach areas, agriculture, monitoring, disaster management and tasks where interaction with humans is likely, since lightweight MAVs can be inherently safe. Most applications need fully autonomous systems, which requires onboard localization and navigation capabilities. Because sensor weight is a crucial factor for these small MAVs, the most promising solution involves the use of a single camera.

Many recent studies focus on odometry and mapping tasks using monocular approaches [1], [2], [3]. These methods provide accurate information about platform motion and structure of the environment, but are very demanding in terms of computing power. Furthermore these methods do not provide the scale of the estimated motions and distances.

For obtaining the scale from single images, learning techniques can be applied. Supervised learning has been applied by training based on a dataset that contains ground truth [4],[5]. But also self-supervised learning methods have been demonstrated that make use of terrain classification and specific system characteristics to combine information from different types of sensors [6], [7], [8]. Downsides of these approaches are either the use of heavy sensors for metric measurements, or assumptions such as ground plane visibility that are not generic for the application of MAVs.

In this study we propose a self-supervised learning approach for monocular distance estimation that makes use of a very small short-range infrared sensor which serves as a near collision detector. By extracting efficient visual features from the camera image sequence before each near collision detection, the system learns the appearance of the object/environment at different distances through regression.

All authors are with the Control and Simulation Section, Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands (e-mail: s.tijmons@tudelft.nl, g.c.h.e.decroon@tudelft.nl)

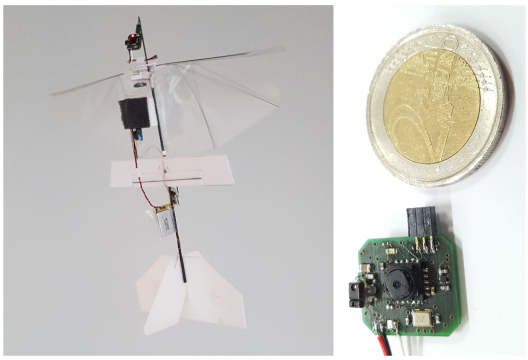

Fig. 1. The DelFly platform, a flapping wing MAV featuring a monocular camera system attached on the nose. A closeup of the camera system is shown at the bottom. The camera board also contains a tiny infrared proximity sensor. This sensor serves as a near collision detector in a self-supervised learning scheme in which the DelFly learns to estimate distances based on obstacle appearance in the camera images.

This approach enables a $19\ g$ MAV to perform collision avoidance based on individual camera images without the need for additional continuous metric information and also enables it to adapt to its environment during flight.

The contribution of this paper is: a self-supervised learning approach that relies on individual camera images and an efficient additional sensor for near collision detection. It is shown that this method can be implemented on a $2\ g$ camera system to provide real time onboard distance estimates to a lightweight MAV.

Section II describes related studies. Section III explains the proposed self-supervised learning method. Section IV discusses implementation details of the method and the setup of the experiments, which are performed in computer simulations (Section V) and on the real platform (Section VI).

## II. Related Work

Monocular vision is a commonly applied method for autonomous navigation of MAVs weighing less than $50\ g$. Since vision sensors are passive and provide a relatively high information density, various studies aim at relying solely on single cameras to perform various control and navigation tasks. Optical flow is an effective method that enables onboard processing on these small platforms, making them fully autonomous. So far, ego-motion estimation [9] and reactive obstacle avoidance [10][11] have been demonstrated using optical-flow. In another study a monocular Simultaneous Localization And Mapping (SLAM) method was demonstrated for hovering and waypoint navigation with such a

small platform [12]. Processing was performed off-board in this study because of the high computational demand of this vision method. On extremely light weight systems even the processing of optical flow is still too demanding. Optical flow based height control was demonstrated on a $101\ mg$ platform by relying on off-board processing [13].

Monocular approaches can provide motion estimation and obstacle detection but these estimations lack direct scale measurements. The most common approach to compute scale is to use stereo vision. This is a computationally demanding method but has been demonstrated to run in real time onboard several platforms [14],[15], even on a light weight platforms [16]. However, this approach requires a second camera, which increases the weight of the sensor payload significantly. The scale ambiguity can also be solved by relying on other secondary sensors. Several studies show monocular-SLAM approaches that use additionally an ultrasound and optical flow sensor facing downwards for measuring absolute horizontal speed [17]. A more elegant approach is to rely on the Inertial Measurement Unit (IMU), as this sensor is already present on many platforms for attitude control, which saves the weight of an additional sensor. By tightly-coupled fusion of IMU and monocular feature tracking measurements the scale problem can be solved [18]. Using this approach is more difficult on a light weight platform because of platform vibrations, and also when the platform has nonholonomic constraints. Another elegant solution is to purely use divergence estimates from monocular optical flow measurements by exploiting the self-induced oscillations that result from the fundamental imperfection of fixed-gain optical flow-based control [19]. This approach does not require additional sensors and has been demonstrated for vertical control of a quadrotor. It is theoretically plausible that also horizontal control can be integrated. The applicability to the platform in our study is less likely due to nonholonomic constraints and because the method requires fast platform dynamics.

Learning techniques have been applied in several vision based applications, also for MAV control. Imitation learning is a form of supervised learning that has been used to map monocular optical flow and visual features to control inputs given by a human pilot [20]. This method allows a quadrotor to avoid trees while flying in a forest. Another (nonlinear) supervised learning method has been used to perform stereo vision based distance estimation without the need to perform camera calibration [21]. Supervised learning has been used to solve the scale problem in monocular vision [4],[5]. Based on image databases with corresponding depth information (from laser or RGB-D measurements) this method learns how to select and use image features to obtain a dense depth map, but only for the trained environment. In self-supervised learning, the system generates its own reference data online. For example, this has been demonstrated on autonomous cars in two different ways. The first approach is to detect which part of the camera view corresponds to drivable road, based on short-range laser data [6]. After learning what the road looks like, the system determines from the camera images how the road continues at larger distances. The second approach is to assume that drivable road is visible right in front of the car and using the fact that the distance between the road and the mounted camera is fixed and known [8]. However, these assumptions cannot be used on flying vehicles. Self-supervised learning has also been demonstrated for the landing task of an MAV [22]. Optical flow information is obtained to detect surface discontinuities while the MAV is moving around. Objects and potential landing locations are then classified and their appearance is learned. When the drone has to land, it can choose landing locations from still images. Combining optical flow and appearance has also been shown in an application where a wheeled robot learns from near collisions with trees [7]. When its infrared sensors detect a near tree, optical flow from a history of images is then used to track it over time and to learn the appearance of the tree for a range of distances. Experiments show that the average number of tree encounters and the time to travel a certain path both decrease significantly using this learning approach. Again the assumption of a ground plane is used to estimate distances.

## III. Self-Supervised Learning for Distance Estimation

In this study a Self-Supervised Learning (SSL) method is proposed that learns to estimate distances from still images. SSL differs from classical supervised learning in that labels are not generated by a human but by the robot itself. Essentially, SSL allows different sensors to work together: measurements on a certain parameter obtained from one sensor are used to label data from a second sensor which does not directly have knowledge about this parameter. When enough training data is collected, this method enables the robot to use only the second sensor to measure this parameter using regression. In this study self-supervised learning is applied to combine a short-range sensor that provides binary distance information with a camera that provides continuous data at all distances. This combination of sensors will provide distance estimates also for longer ranges.

### A. Distance estimation methods

Distance measurements are performed in two ways: using camera images in combination with information from the learning process or using a proximity sensor for detecting near collisions. The first method, using the camera, requires that the system can rely on training data that was learned in the past. The second method, using the proximity sensor, only indicates whether the distance to a nearby obstacle is too small to continue flying in the current direction. In this case, two simultaneous actions are performed: the vehicle changes its heading, and recently recorded images are used to perform an iteration in the learning process.

*1) Camera:* A TCM8230 color camera provides RGB images throughout each flight. To enable distance estimation on the limited processor ($168\ MHz$, $192\ kB$), data reduction of the images is realised using an efficient image descriptor which has the form of a histogram. The histogram indicates the frequencies of a predefined set of textons [23]. Textons

are fundamental micro-structures in images. In this study a set of $R$ small representative RGB image patches form a dictionary of textons. From each camera image, $N$ evenly spread patches with the same size as the textons are extracted and matched with the texton dictionary, based on minimum Euclidean distance. The indices of all best matches with the dictionary form the histogram of texton occurrences for each image. Note that the histograms contain information about the overall appearance of images, not on local image patterns. The histograms are used for two purposes. First, to obtain a distance estimate for the current image based on what has been learned (Section III-B explains how this is done). Second, to serve as temporary training data in case a near collision occurs shortly after the image was recorded.

*2) Proximity sensor:* A TMG399x infrared proximity sensor ($2 \times 4\ mm$) is used as a short-range binary detector for near collisions; it indicates whether an object is detected within a range of $\approx 50\ cm$ which allows the MAV to perform an evasive manoeuvre. In case a near collision is detected, the manoeuvre is executed and recently stored histograms are assigned a distance label. The distance assigned to each histogram is based on retrograde extrapolation assuming constant heading, constant flight speed and constant frame rate. These distances are regarded as ground truth and are used to perform an iteration in the learning process.

### B. Learning Algorithms

The effectiveness of different learning algorithms is tested in this study. These algorithms have two functions. First, to provide a distance estimate based on a histogram input. Second, to learn from near collisions by importing histograms with assigned distance labels as training data.

*1) Perceptron Network:* The simplest approach that was tested is an ADALINE network, which is a single-layer perceptron without hard limits. A perceptron is a simple form of a neural network in which the output **a** (distance estimate) is the weighted sum of all inputs **p** (the $R \times 1$ histogram) and a bias term:

$$\mathbf{a} = \mathbf{W}\mathbf{p} + \mathbf{b} \tag{1}$$

When a near collision occurs, the weights $\mathbf{W}$ $(1 \times R)$ and bias **b** are updated with a Widrow-Hoff learning rule [24]:

$$\begin{aligned} \mathbf{W}(k+1) &= \mathbf{W}(k) + 2\alpha\mathbf{e}(k)\mathbf{p}^{\mathbf{T}}(k) \\ \mathbf{b}(k+1) &= \mathbf{b}(k) + 2\alpha\mathbf{e}(k) \end{aligned} \tag{2}$$

In this equation, $\alpha$ is the learn rate and **e** is the error between **a** and the corresponding ground truth label.

*2) k-NN:* k-Nearest Neighbours (k-NN) is an algorithm that can be used for both classification and regression problems. In k-NN regression an input feature vector is compared with the full set of trained feature vectors and the $k$ nearest neighbours (based on smallest Euclidean distances) are used to calculate the output using the labels of the training samples. In this study, the feature vectors are formed by the image histograms that are labeled with distance values. In case $k > 1$, distances are estimated by taking the average of the corresponding distance labels. k-NN is a type of lazy learning; new training data (histograms with distance labels) is simply added to the training set. This makes the training phase fast, but leads to large amounts of training data that needs to be stored and a slow (distance) evaluation process.

*3) k-NN with clustering:* To solve the mentioned limitations of the k-NN algorithm, a clustering method is proposed that reduces the amount of stored training data. Similar methods, such as condensation [25] or instance selection [26], have been proposed to remove either noisy samples from the training set to improve accuracy, or to eliminate redundant samples to optimally reduce the size of the training set. The currently proposed clustering method is based on the assumption that similar histograms correspond to similar parts of an environment, and that merging their feature vectors and labels is therefore legitimate. This allows for storage of a fixed amount of training data while maintaining diversity. Clustering is done by looking for pairs of similar histograms (based on Euclidean distance) and by only storing the averages of their histogram values and labeled distances.

## IV. Implementation and Test Setup

The proposed distance estimation method is used to enable obstacle avoidance on a lightweight flapping wing MAV, the DelFly [27], [16]. In this study the vehicle has a wing span of $28\ cm$ and a weight of $17\ g$. This includes a $1\ g$ autopilot with an IMU (MPU9150) and a barometer. Its payload is a $2\ g$ camera system featuring a TCM8230 color camera, a TMG399x infrared proximity sensor and an STM32F405 ARM processor. The system has a total weight of $19\ g$ and is able to run the learning algorithm on board.

Its flight characteristics make the DelFly a suitable platform to use the proposed distance estimation method because it flies passively stable with a constant low forward speed. The vertical speed and distance to the ground can be restricted using feedback from the barometer. At low speed the vehicle can perform avoidance maneuvers within the space covered by the short-range proximity sensor, such that no real crashes occur. Using gyroscope feedback a fixed heading can be maintained when no control action is required.

The camera system is mounted in a specific way to the DelFly; the proximity sensor is aligned with the forward velocity vector and thus looks straight ahead. The camera is rotated with an offset of $15\ deg$ to the left. The following sections on simulation and flight tests explain how this setup is exploited for control purposes. Tests haven been performed in a small room of $4 \times 4\ m$ which has walls with different types of textures, as shown in Fig. 2. The top image in this figure shows the test room as used for simulations. For the $6 \times 6\ m$ simulated room photos from the real room are used.

## V. Simulations

Computer simulations were initially performed to test and analyse the performance of the different learning-algorithms and to explore effective reactive avoidance strategies.

Fig. 2. **Top**: screenshot showing the simulated environment. **Bottom**: photo of the DelFly flying in the test room. The simulated room is an imitation of the real test room.

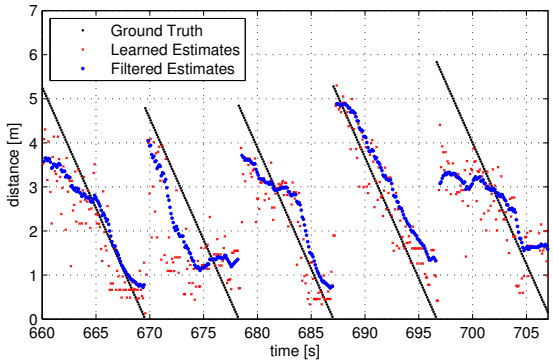


Fig. 3. Estimated distance with k-NN using 500 clustered points after 660 $s$ of training.

### A. Distance estimation performance

To compare the different learning algorithms, the vehicle is simulated as flying at a constant height, in straight lines and with a speed of 0.55 $m/s$. Each time the vehicle hits a wall, its heading is changed instantly with a random offset such that it continues flying within the test room. This way a data set with recorded image histograms (10 $Hz$) and flight tracks is obtained to test the learning algorithms.

Fig. 3 shows an example of the distance estimation performance of k-NN learning with clustering after 660 $s$ of training.Individual distance estimates are presented, as well as the result after low-pass filtering. It is observed that the filtered estimates show an obvious correspondence with the ground truth data, even though the estimates contain significant noise. For this reason the performance of the learning algorithms is expressed by the correlation coefficient, which is a measure of the linear dependence between the ground truth data points and the estimated data points.

Fig. 4 shows the correlation coefficients for the different algorithms over time. The most interesting observation is that the clustered k-NN method outperforms the Widrow-Hoff method and has almost similar performance as the standard k-NN method. The correlation coefficient increases

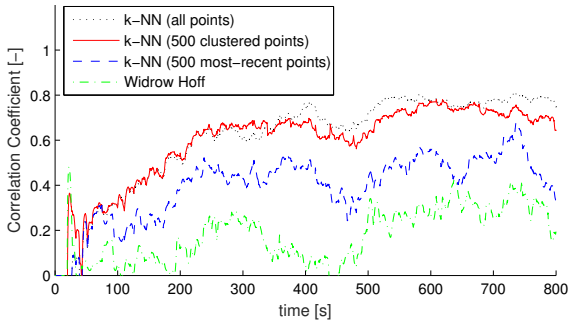


Fig. 4. Correlation between distance estimate and ground truth using different algorithms. For k-NN with all points, all histograms are stored which after 600 seconds adds up to 6000 data points. Both other k-NN methods keep only 500 data points in memory. The clustered approach is also compared to an approach that simply remembers the most recent points. All k-NN methods use $k = 5$. For the Widrow-Hoff approach, $\alpha = 0.05$. All methods use histograms with $R = 24$ bins constructed from matching $N = 70$ patches. The patch size is $5 \times 5$ pixels.

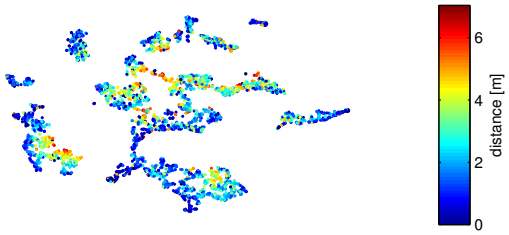


Fig. 5. t-SNE: 3000 histograms with 30 texton bins.

significantly during the first few minutes, and reaches a more steady performance afterwards. The effectiveness of the clustering method is clearly visible from Fig. 4. For comparison the performance is shown if a fixed number of histograms would be maintained by simply dumping the oldest histograms. After reaching the maximum number of histograms (50 $s$) the difference in performance becomes visible. The effectiveness of the clustering method can be explained by analysing the histogram data using the t-SNE [28] algorithm. It allows to visualise the total set of high dimensional histograms as a two-dimensional image, as shown in Fig. 5. Each histogram is a point in this image, and relative distances between the points are based on similarity. The result shows that clusters of similar points are formed, and that within the groups color gradients are visible. This confirms the hypothesis that similar histograms correspond to the same part of the test room and that the histograms change gradually with distance. The proposed clustering method is thereby justified and used in further experiments.

### B. Control

The camera is mounted with an offset on the vehicle, also in simulation. This allows for obtaining individual distance estimates (k-NN with 500 clustered points) for the two halves of the camera images: one halve that looks straight ahead and one halve that looks to the left side. Control is based on

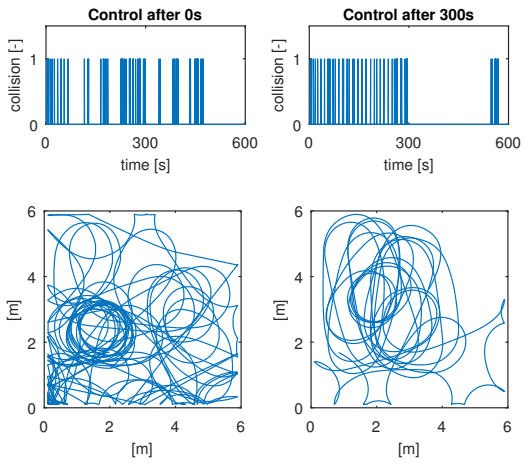


Fig. 6. Near collision events and flight trajectories of two simulated flights where vision-based control is applied. The left plot shows results when control is active from the start. The right plot shows results when control is activated after 300 seconds. The bottom plots show trajectories from the moment vision-based control was turned on.

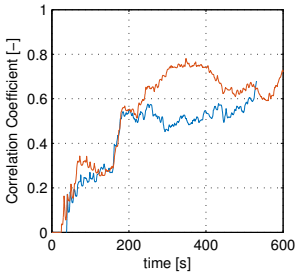


Fig. 7. Correlation coefficient of learned distance estimates over time for two different runs while walking with the DelFly through the test room.

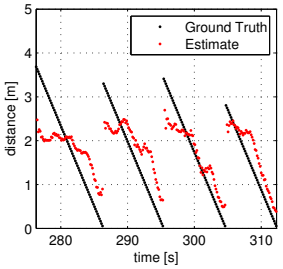


Fig. 8. Partial results showing estimated distance versus ground truth data from a test while walking with the DelFly.

thresholds on the two distance estimates; if either of the two estimates indicates a small distance, the vehicle turns left with constant input. Otherwise the vehicle flies straight. The threshold used for the left side is more conservative to ensure a free space on the side to perform an avoidance manoeuvre.

Fig. 6 shows results of two tests where the vehicle uses its distance estimates for avoidance control. In the first test avoidance control is active from the start, in the second test the switch to active avoidance is made after $300\ s$. In both tests training is performed throughout the flight. For the second test only the flight trajectory after the control switch is shown. From the results it is clear that in the second test the near collision rate after $300\ s$ is much lower than in the first test. This can be explained by the total number of near collision events which is much higher at this point. Furthermore the flight track of the first test indicates that the vehicle flies a lot of small circles which slows down the learning process. The moment of switching apparently influences the total number of near collision events that occurs within a certain amount of flight time.

## VI. Experimental Results

The k-NN learning method using $500$ clustered points has been implemented on the $2\ g$ stereo vision system of the DelFly. First tests show the potential performance of this system by manually walking around in the test room. In these tests the operator walks in straight lines, and chooses a new random direction when the proximity sensor indicates a near collision. Results of these performance tests are shown in Fig. 7. The correlation coefficient reached after $600\ s$ is comparable to what was observed in the simulation results. Fig. 8 shows distance estimates during one of the tests after a correlation coefficient of more than $0.5$ was reached. These results clearly visualise that wall approaches can be detected.

Autonomous flight tests with the DelFly have been performed using the proposed learning approach. In these tests the altitude is regulated using barometer feedback and the heading is controlled using gyroscope feedback (for enabling straight flight paths during wall approaches). Furthermore, a visual tracking system is used for logging the position of the vehicle and to assist in deciding to turn left or right in case of a near collision detection. Especially in case the wall is approached non-perpendicular it is critical to turn in the right direction. This assistance can be made superfluous by increasing the heading control authority of the vehicle at low speeds. Fig. 9 shows distance estimation performance results of three different flights. In these tests the trained data in the camera system is cleared prior to the flight. Note that lower correlation coefficients are reached as in previous experiments ($0.5 - 0.6$ instead of $0.6 - 0.7$), and that the learning rate is lower. Fig. 10 shows distance estimates during the final part of one of these test flights. These results also show a worse performance compared with previous tests, especially for the small ranges. This can be explained by variations in the altitude, heading changes due to disturbances, and platform vibrations. These influences result in variations in camera observations, especially at distances close to the walls.

To show the potential of the proposed learning method the autonomous tests were split in two parts. First a training flight was performed to train the camera system in the test room. In this test the vehicle only performs collision avoidance manoeuvres based on inputs from the proximity sensor. A flight trajectory of such a test is shown in Fig. 11. After the training flight the trained data is retained for the next flight. During this subsequent flight the distances estimated by the learning algorithm are used as control input. A threshold value of $1.2\ m$ is used to decide whether to fly straight or to turn left. The results are shown in Fig. 12.

## VII. Conclusions

A self-supervised learning method is proposed that enables a lightweight MAV to estimate distances based on monocular images. The method combines distance information from a small proximity sensor during near collisions with an efficient image description algorithm to enable online distance

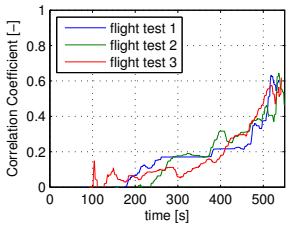


Fig. 9. Correlation coefficient of learned distance estimates over time for three flight tests with the DelFly using closed loop heading control.

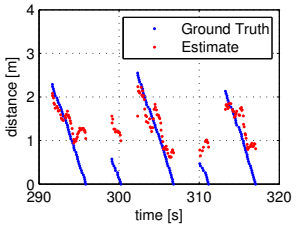


Fig. 10. Partial results showing estimated distance versus ground truth data from a real flight test with the DelFly using closed loop heading control.

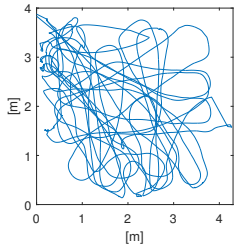


Fig. 11. Flight trajectory of training test flight of the DelFly equipped with the monocular camera system.

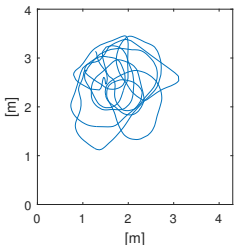


Fig. 12. Flight trajectory of the DelFly with monocular distance estimates in the loop for active wall avoidance.

estimation on a 2 $g$ camera system. The k-NN based learning method uses a clustering step to limit the amount of stored training data that has a marginal effect on performance. Computer simulations show that the proposed method allows the MAV in this study to significantly reduce the number of near collisions over time. Real world tests indicate that similar performance can be reached on the real system. Real test flights indicate a lower learning rate, but show that collision avoidance is possible using the proposed method.

## REFERENCES


[1] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.

[2] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 834–849.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *Robotics, IEEE Transactions on*, vol. 31, no. 5, pp. 1147–1163, 2015.

[4] C. Plagemann, F. Endres, J. Hess, C. Stachniss, and W. Burgard, "Monocular range sensing: A non-parametric learning approach," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 929–934.

[5] K. Bipin, V. Duggal, and K. Madhava Krishna, "Autonomous navigation of generic monocular quadcopter in natural environment," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1063–1070.

[6] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, "Self-supervised monocular road detection in desert terrain." in *Robotics: science and systems*. Philadelphia, 2006.

[7] A. Lookingbill, D. Lieb, and S. Thrun, "Optical flow approaches for self-supervised learning in autonomous mobile robot navigation," in *Autonomous Navigation in Dynamic Environments*. Springer, 2007, pp. 29–44.

[8] B. Lee, K. Daniilidis, and D. D. Lee, "Online self-supervised monocular visual odometry for ground vehicles," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5232–5238.

[9] A. Briod, J.-C. Zufferey, and D. Floreano, "Optic-flow based control of a 46g quadrotor," in *Workshop on Vision-based Closed-Loop Control and Navigation of Micro Helicopters in GPS-denied Environments, IROS 2013*, no. EPFL-CONF-189879, 2013.

[10] R. J. D. Moore, K. Dantu, G. L. Barrows, and R. Nagpal, "Autonomous mav guidance with a lightweight omnidirectional vision sensor," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3856–3861.

[11] J.-C. Zufferey, A. Klaptocz, A. Beyeler, J.-D. Nicoud, and D. Floreano, "A 10-gram vision-based flying robot," *Advanced Robotics*, vol. 21, no. 14, pp. 1671–1684, 2007.

[12] O. Dunkley, J. Engel, J. Sturm, and D. Cremers, "Visual-inertial navigation for a camera-equipped 25g nano-quadrotor," in *IROS2014 aerial open source robotics workshop*, 2014.

[13] P.-E. J. Duhamel, N. O. Pérez-Arancibia, G. L. Barrows, and R. J. Wood, "Altitude feedback control of a flapping-wing microrobot using an on-board biologically inspired optical flow sensor," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4228–4235.

[14] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss, "Stereo vision-based obstacle avoidance for micro air vehicles using disparity space," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3242–3249.

[15] A. J. Barry and R. Tedrake, "Pushbroom stereo for high-speed navigation in cluttered environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3046–3052.

[16] C. De Wagter, S. Tijmons, B. D. W. Remes, and G. C. H. E. de Croon, "Autonomous flight of a 20-gram flapping wing mav with a 4-gram onboard stereo vision system," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4982–4987.

[17] H. Alvarez, L. M. Paz, J. Sturm, and D. Cremers, "Collision avoidance for quadrotors with a monocular camera," in *Experimental Robotics*. Springer, 2016, pp. 195–209.

[18] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5303–5310.

[19] G. C. H. E. de Croon, "Monocular distance estimation with optical flow maneuvers and efference copies: a stability-based strategy," *Bioinspiration & biomimetics*, vol. 11, no. 1, p. 016004, 2016.

[20] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, "Learning monocular reactive uav control in cluttered natural environments," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1765–1772.

[21] F. Sinz, G. Quinonero-Candela, G. Bakir, C. Rassmussen, and M. Franz, "Learning depth from stereo," in *Pattern Recognition Proc 26th DAGM Symposium LNCS 3175*, 2004, pp. 245–252.

[22] H. W. Ho, C. De Wagter, B. D. W. Remes, and G. C. H. E. de Croon, "Optical-flow based self-supervised learning of obstacle appearance applied to mav landing," *arXiv preprint arXiv:1509.01423*, 2015.

[23] M. Varma and A. Zisserman, "Texture classification: Are filter banks necessary?" in *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*, vol. 2. IEEE, 2003, pp. II–691.

[24] B. Widrow, M. E. Hoff *et al.*, "Adaptive switching circuits," in *IRE WESCON convention record*, vol. 4, no. 1. New York, 1960, pp. 96–104.

[25] F. Angiulli, "Fast condensed nearest neighbor rule," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 25–32.

[26] A. Arnaiz-González, M. Blachnik, M. Kordos, and C. García-Osorio, "Fusion of instance selection methods in regression tasks," *Information Fusion*, vol. 30, pp. 69–79, 2016.

[27] G. de Croon, M. Groen, C. D. Wagter, B. Remes, R. Ruijsink, and B. van Oudheusden, "Design, aerodynamics, and autonomy of the delfly," *Bioinspiration and Biomimetics*, vol. 7, no. 2, 2012.

[28] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.