The Appearance Variation Cue for Obstacle Avoidance

G.C.H.E. de Croon[1], E. de Weerdt[2], C. De Wagter[2], B.D.W. Remes[2], and R. Ruijsink[2*†]

Abstract

The *appearance variation cue* captures the variation in texture in a single image. Its use for obstacle avoidance is based on the assumption that there is less such variation when the camera is close to an obstacle. For videos of approaching frontal obstacles, it is demonstrated that combining the cue with optic flow leads to better performance than using either cue alone. In addition, the cue is successfully used to control the 16-gram flapping wing MAV DelFly II.

1 Introduction

Autonomous flight of biologically inspired flapping wing Micro Air Vehicles (MAVs) is a largely uncharted area of research. The design and construction of flapping wing MAVs is still a subject of study, cf. [3], and only a few flapping wing MAVs exist that can carry a payload such as a camera onboard [1, 13, 5]. Therefore, studies on controlling flapping wing MAVs mostly focus on the use of external cameras [10].

[*] (1). Advanced Concepts Team, European Space Agency, (2). Micro Air Vehicle lab, Control and Simulation, Delft University of Technology. Contact: guido.de.croon@gmail.com

In fact, the light weight and limited energy supply of flapping wing MAVs prevent the imitation of successes in autonomous flight with quad rotors using Simultaneous Localization And Mapping (SLAM) [4, 17, 26]. These successes rely on miniaturized laser range finders with a weight and energy consumption largely exceeding the payload capabilities of a flapping wing MAV such as the DelFly II (see Fig. 1).

In this article we focus on using onboard images of a flapping wing MAV for the avoidance of large frontal obstacles in indoor environments. At the moment, there are two main approaches for achieving indoor flight with monocular vision. The first approach is visual SLAM, in which the state of the MAV (3D position and attitude) is estimated by 'matching' camera images to known locations in a 3D-model of the environment [12, 2, 9, 7]. This approach attempts to tackle both obstacle avoidance and navigation in the same time. However, the algorithms still have problems with drift [7] and are computationally expensive.

The second approach is a bio-inspired approach that is computationally more efficient. Typically a state estimate is abandoned altogether and the MAV directly responds to the incoming visual inputs [6, 22]. Generally, optic flow is used [19, 24, 18], since it is known to play an important role in insect flight [16, 11]. The flow of image points away from the Focus of Expansion (FoE) can be used for estimating the time-to-impact to an obstacle (cf. [27]). The limitations of this strategy include its dependence on texture in the environment and the reliance on accurate optic flow measurements (especially close to the FoE).

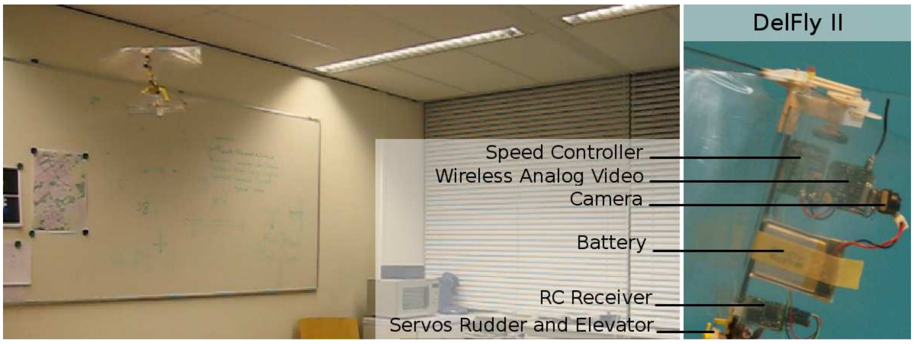

Figure 1: The 16-gram, 28 cm wing span flapping wing MAV DelFly II during an obstacle avoidance experiment with monocular vision. The inset shows the top of the DelFly's body with its electronic components.

Unfortunately, the images made with current state-of-the-art flapping wing MAVs are rather degraded, rendering optic flow measurements quite noisy. In particular, the flapping movements in combination with the line-by-line recording of the video images can lead to considerable image deformations. Fig. 2 shows two subsequent images made with the camera onboard the DelFly II. These images are particularly affected by the flapping movements: the (straight) edge of the closet in the center of the image is curved to the right in the first image and curved to the left in the second image. As a consequence of these unpredictable deformations, images often deviate from the linear camera model assumed by optic flow algorithms. The poor quality of the images and the reliance of optic flow on texture lead to a necessity for complementing optic flow with other visual cues.

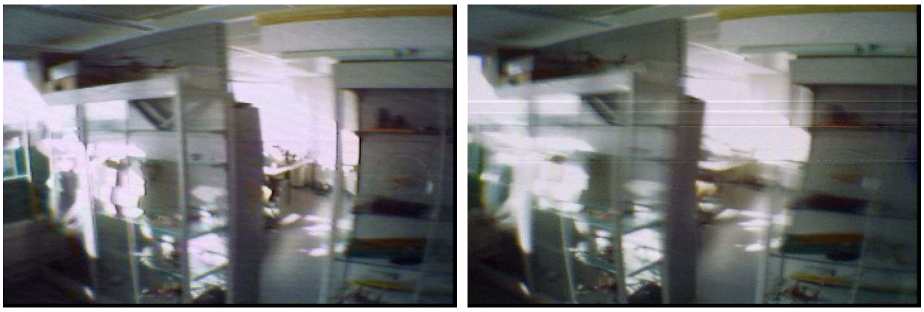

Figure 2: Two subsequent images in a video made onboard the DelFly II with deformations caused by the combination of the flapping frequency of the DelFly and the line-by-line recording of the camera.

The main contribution of this article is the investigation of a novel visual cue for obstacle detection. The cue captures the variation in texture in a single image, and is based on the assumption that there is less such variation when the camera is close to an obstacle. The *appearance variation cue* is a valuable complement to optic flow for autonomous obstacle avoidance in indoor environments.

The remainder of the article is organized as follows. In Section 2, we investigate the appearance variation cue. In Section 3 it is shown that the novel cue is complementary to the time-to-impact determined by optic flow. In Section 4, the DelFly II autonomously avoids obstacles using the appearance cue and optic flow. Finally, we draw conclusions in Section 5.

2 Appearance Variation as a Cue for Obstacle Proximity

Robotics research on monocular vision has mainly focused on apparent motion and perspective cues for automatically determining distances to objects in a robot's environment. However, it is well-known (cf. [20]) that humans use various other, complementary monocular vision cues to this end, including: occlusion, image size of familiar objects, aerial perspective, lens accomodation, blur, and texture gradient. The novel cue for estimating obstacle proximity introduced here is termed the *appearance variation cue.* When an observer approaches an object, there are two concurrent effects:

1. The image size of the object increases in the image, while other objects go out of view.

2. The detailed texture of the object in view becomes more and more visible.

The *main assumption underlying the novel cue* is that the variation in appearance of many different objects in view is larger than that of the detailed texture of one object alone. In other words, it is assumed that in general the first effect decreases the appearance variation more than the second effect increases it.

The appearance variation cue depends on the distance and on the textures of the objects in view. We expect the cue to be complementary to optic flow, since (i) it is fit for dealing with the absence of texture, a fail-case for optic flow, (ii) it directly depends on the distance and does not require motion of the observer, and (iii) it does not require accurate, sub-pixel measurements in the image, which can make it more robust to noise. The complementarity is expected to be mutual: if the cue's main assumption is invalid, this implies that there are different textures in view, which is beneficial for determining optic flow. The verification of the expected complementarity between the two cues is performed in the Section 3.2. In the remainder of this section, we verify whether the appearance variation indeed decreases towards impact.

2.1 Measuring appearance variation

For measuring the appearance variation, in this study the term *appearance* is interpreted as textures. One could also take colors into consideration, but this is left to future work. The approach to estimating the *variation* of texture is to first estimate a probability distribution p of different textures in the image. Subsequently, the Shannon entropy $\mathcal{H}$ [25] of the estimated probability distribution is calculated.

For automatically determining the texture distribution, the *texton* method [28] is used. This method evaluates texture on the basis of small local image samples and was shown to outperform computationally more intensive filtering methods (e.g., Gabor filters) on a texture classification task. Below, we describe the implementation of the texton method in our experiments.

The texton method starts with the formation of a *dictionary* of n textons[1]. To this end, small image samples of size $w \times h$ pixels are extracted from a set of images of size $W \times H$. The samples are clustered by means of a Kohonen network [21].

After learning the dictionary, the texton method evaluates texture by estimating the probability distribution of textons in the image. s image samples are extracted from the image to build a histogram g with n bins. For each sample, the closest texton i in the dictionary is determined (Euclidian distance), and the corresponding bin in the histogram g_i is incremented. Normalizing g results in a maximum likelihood estimate $\hat{p}$ of the texton distribution in the image, with $\hat{p}_i = g_i/s$. This estimate is then inserted into the formula of entropy to determine the texture variation: $\mathcal{H}(\hat{p}) = -\sum_{i=1}^{n} \hat{p}_i \log_2(\hat{p}_i)$. A high entropy corresponds to a high variation in appearance, while a low entropy corresponds to the contrary.

One aspect of the method is worth mentioning. Computer vision applications typically extract all possible local samples from the image, making $\hat{p}$ equal to p. We first follow this convention, but for the classification and robotic experiments (Section 3 and 4) a number of samples $s \ll WH$ is extracted for computational efficiency. The higher efficiency comes at the cost of a slightly lower accuracy (cf. [14]).

2.2 Experimental setup

Three types of experiments are performed, all with a focus on obstacle distances typical for flying ranges of MAVs in indoor environments (3m - 0m). First, we determine the appearance variation for a small set of 10 obstacle approach sequences made by holding an analog wireless camera in hand.

[1] All parameter settings will be mentioned in Subsection 2.2.

Each approach starts from a distance of 3 meters from a wall or other large obstacle, which is approached with an approximately constant pace. All images are resized to 160 × 120 pixels. Second, a larger set of 65 approach sequences is studied, captured in the same manner but with the camera of a mobile phone. The top row of Fig. 3 shows three shots from one of the sequences with the mobile phone. Third, to study an even larger number of approach sequences, we also simulate approach sequences by zooming in on digital photographs. Of course, the disadvantages of simulated approaches are the absence of (a) three-dimensional visual effects and (b) realistic noise conditions.

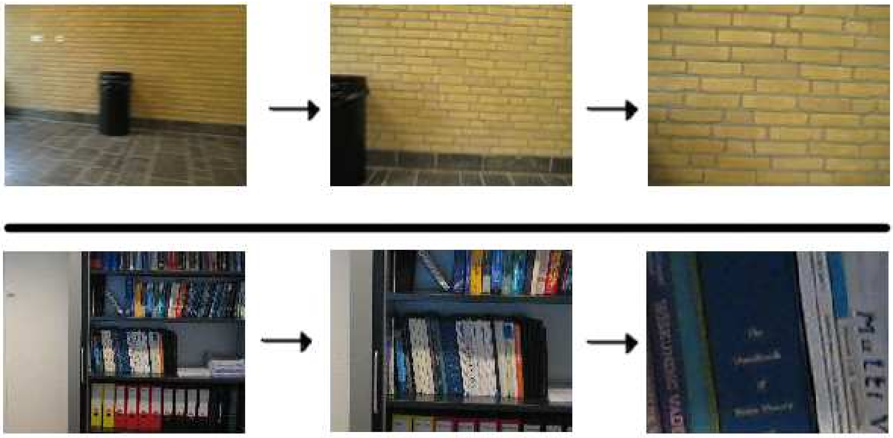

Figure 3: Example images from an actual approach sequence captured with a mobile phone (top) and a simulated approach sequence (bottom).

For the simulated approaches, a set of 62 photographs was taken in inside environments at a distance of 3 meters from a wall or other large obstacle. All photographs have a dimension of 3072 × 2304 pixels. The simulated camera images have a dimension of 160 × 120 pixels. At the start of the approach sequence the camera captures an area of 2000 × 1500 pixels in the original image, which is resized to the virtual camera dimension with bicubic sampling. At the end of the approach, the captured area in the original image equals 160 × 120 pixels. Digitally zooming in even further would always result in a lower appearance variation, since no new details can come into sight. Each sequence comprises 90 frames during which the width and height of the captured image area decreased linearly. The bottom row of Fig. 3 shows three images from one of the simulated sequences.

The settings of the Texton Method Gray (*TMG*) are as follows. The size of the image patches is $w \times h = 5 \times 5$, and it extracts all possible samples in the gray-scale image, $s = 155 \times 115 = 17825$. The dictionary has $n = 30$ textons and is learned on a separate set of images not belonging to the approach sequences.

2.3 Results

Fig. 4 shows the TMG's entropy values over time (thin gray lines) for the 10 wireless video sequences. Time is represented on the x-axis, the entropy on the y-axis. It also shows linear fits (thick red lines) minimizing the mean square error to the data points.

As can be seen in Fig. 4, the slopes of the entropy over time are all negative for the wireless video sequences. In order to get an idea of how often the entropy decreases towards impact, the proportions of negative slopes are determined for the other video sets as well. For the larger video set of 65 sequences made with the mobile phone 80% of the slopes is negative. For the 310 simulated sequences 90% of the slopes is negative. The large proportions of negative slopes imply that the appearance variation generally decreases towards impact.

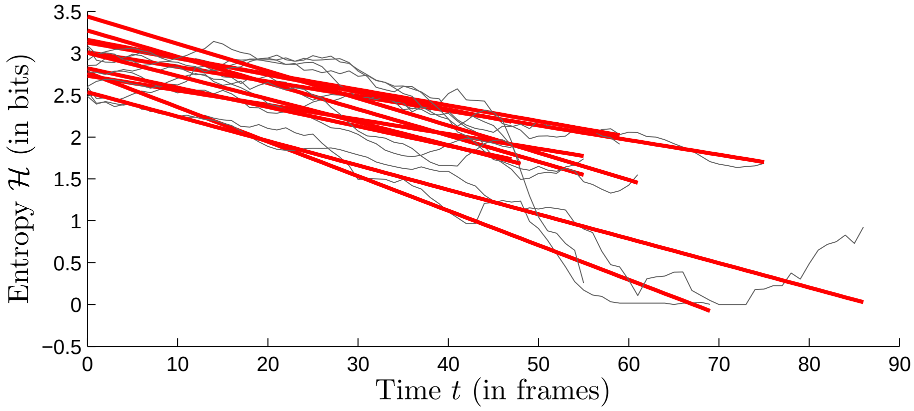

Figure 4: The entropy of TMG over time (light gray lines) and corresponding linear fits (thick red lines). The entropy decreases towards impact.

In some sequences the entropy increases over time. Investigation of these sequences showed that they resemble the one in the bottom row of Fig. 3.

The simulated camera approaches a book shelf that has objects in it with different, highly detailed textures. The detailed texture leads to an increasing entropy as it comes into sight. The existence of sequences in which the entropy increases and the different offsets of the entropy (see Fig. 4) suggest that the sole use of the appearance variation cue will not lead to perfect obstacle avoidance.

3 Complementarity with Optic Flow

In this section, we investigate a task in which the appearance variation cue and / or the optic flow are used to classify whether the time-to-impact τ is larger or smaller than a desired detection time τ^*. In Subsection 3.1 the optic flow method is explained. In Subsection 3.2 the classification performances of the methods are evaluated.

3.1 Optic Flow for Calculating Time-To-Impact

The optic flow algorithm's implementation consists of two parts: (1) finding and tracking feature points to determine several optic flow vectors between two images, and (2) determining τ on the basis of the vectors.

For the first part, the method of Lucas-Kanade [23, 8] from the openCV library is used (`http://www.opencv.org/`). The second part is performed as follows. It is assumed that the camera is moving straight towards a flat surface, while the camera possibly undergoes pitch and yaw changes. The Focus of Expansion (FoE) is estimated with the least-squares method described in [27]. Subsequently, the optic flow vectors are used to determine the distance from the old location (x_t, y_t) and the new location (x_{t+1}, y_{t+1}) to the FoE (x_e, y_e); $d_{e,t}$ and $d_{e,t+1}$ respectively. The difference in distance to the FoE is $\Delta d_{e,t}$. Each optic flow vector leads to one estimate $\hat{\tau}$:

$$\hat{\tau} = d_{e,t}/\Delta d_{e,t} \tag{1}$$

Since it is assumed that there is one flat surface and the optic flow vectors are noisy, the final estimate τ_{OF} is taken to be the median of the resulting $\hat{\tau}$-distribution. The uncertainty of τ_{OF} can be captured with the standard deviation of the $\hat{\tau}$-distribution (σ_{OF}). Despite the strong assumptions, this straightforward method works rather well in practice.

3.2 Classification Performance

After determining TMG's entropy, τ_{OF}, and σ_{OF} on the set of videos, we investigate the classification performances of different combinations of methods. The task is to classify a time step t as positive when $\tau \leq \tau^*$. The following logical expressions are used for classification: '$\tau_{OF} < \vartheta_1$', 'TMG $< \vartheta_1$', '$\tau_{OF} < \vartheta_1$ and $\sigma_{OF} < \vartheta_2$', '$\tau_{OF} < \vartheta_1$ or TMG $< \vartheta_2$', and '($\tau_{OF} < \vartheta_1$ and $\sigma_{OF} < \vartheta_2$) or TMG $< \vartheta_3$'. The rationale behind this last expression is that the optic flow estimate should only be trusted when it is accurate enough, with the entropy of TMG to ensure detection if this is not the case. For brevity, the thresholds ϑ_i will be henceforth omitted.

By varying the thresholds in the above expressions, a Receiver Operator Characteristic (ROC) curve [15] can be made that represents the trade-off between True Positives (TPs, detections when τ is indeed smaller than τ^*) and False Positives (FPs, detections when τ is actually larger than τ^*).

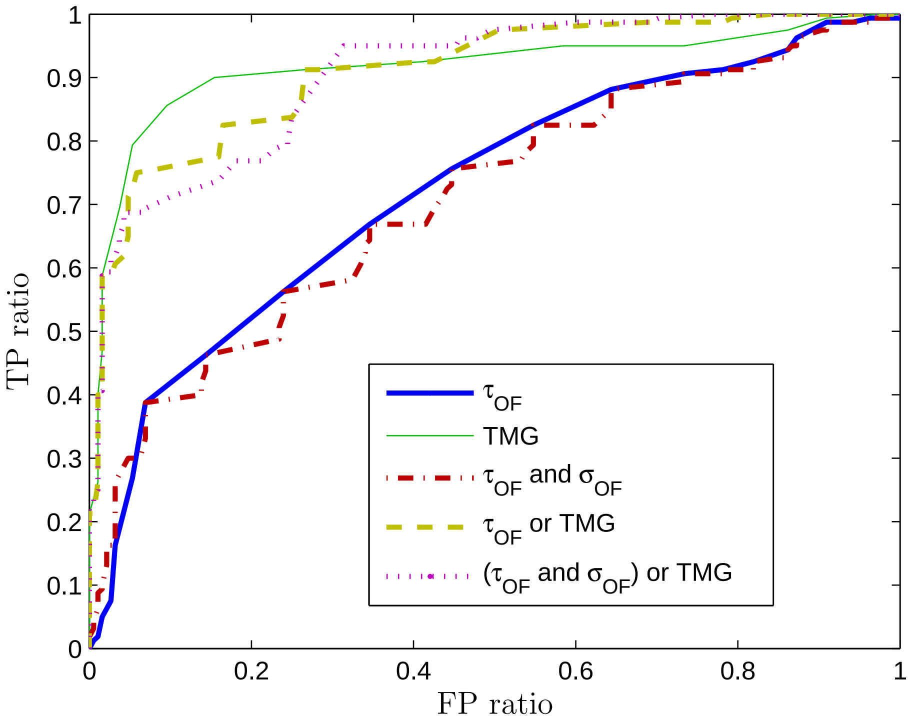

Figure 5: ROC-curves of different combinations of variables, $\tau^* = 1\frac{1}{3}$ s., for the video sequences made with the flying DelFly II.

Table 1: AUC for combinations of τ_{OF}, σ_{OF}, and TMG. Bold indicates the highest AUC. DelFly II sequences.

τ^*	τ_{OF}	TMG	τ_{OF} and σ_{OF}	τ_{OF} or TMG	(τ_{OF} and σ_{OF}) or TMG
$\frac{1}{3}$	0.787	0.844	0.722	**0.846**	0.824
$\frac{2}{3}$	0.727	**0.848**	0.713	0.831	0.810
1	0.775	**0.934**	0.753	0.922	0.905
$1\frac{1}{3}$	0.726	**0.916**	0.709	0.910	0.906
$1\frac{2}{3}$	0.682	0.853	0.672	**0.860**	0.850

Fig. 5 shows the ROC-curves for 19 video sequences recorded with the camera onboard the flapping wing MAV DelFly II. It includes the methods 'τ_{OF}' (blue solid), 'TMG' (green thin solid), 'τ_{OF} and σ_{OF}' (red dash-dotted), 'τ_{OF} or TMG' (yellow dashed), and '(τ_{OF} and σ_{OF}) or TMG' (purple dotted) for $\tau^* = 1\frac{1}{3}$ s. The x- and y-axis represent the FP ratio and TP ratio, respectively. The higher the curve, the better.

The results show that complementing optic flow measures with the appearance variation cue leads to improved collision detections. TMG has the highest TP ratios for FP ratios smaller than ~ 0.3. For higher FP ratios, the combination of TMG with σ_{OF} and τ_{OF} gives the best results. Optic flow measures alone do not lead to acceptable performances, because the DelFly images are rather degraded and distorted.

The findings above are further confirmed by investigating the quality of the classifier by means of the Area Under the Curve (AUC). Table 1 shows the AUC-values for the different methods for different τ^*. A bold setting indicates the best method. The optic flow estimate τ_{OF} is constantly outperformed by TMG on the DelFly images. The best performances are either obtained by the combination of τ_{OF} and TMG, or by TMG alone. Importantly, these results have been obtained while TMG only extracted $s = 100$ samples from each image - a number low enough to ensure real-time operation.

On the other (less degraded) image sets such as the hand-held video sequences, optic flow performs on a par with TMG (see Table 2). On those sequences, the best results are almost always obtained by a combination of the two methods.

Table 2: AUC for combinations of τ_{OF}, σ_{OF}, and TMG. Bold indicates the highest AUC. Hand-made sequences.

τ^*	τ_{OF}	TMG	τ_{OF} and σ_{OF}	τ_{OF} or TMG	(τ_{OF} and σ_{OF}) or TMG
$\frac{1}{3}$	0.836	0.881	0.769	0.876	**0.897**
$\frac{2}{3}$	0.893	**0.976**	0.848	0.963	0.877
1	0.926	0.927	0.899	**0.942**	0.936
$1\frac{1}{3}$	0.908	0.819	0.896	0.867	**0.944**
$1\frac{2}{3}$	0.887	0.733	0.880	0.769	**0.974**

4 Obstacle Avoidance Experiment

The final experiment involves the use of both the appearance variation cue and optic flow to control the flapping wing MAV DelFly II [13] in order to verify whether the detection performance is good enough for successful obstacle avoidance.

The DelFly II cannot yet carry its own processing onboard. The images are made with a forward-looking, onboard MO-S588 1/4" CMOS NTSC color camera ($f = 3.1$mm, with FOV$_h$ = 61.3°, FOV$_v$ = 46°). They are sent to a ground station for processing. While the vision algorithms are currently running offboard, their efficiency remains of uttermost importance in order to control small MAVs with fast dynamics and to have a chance of being implemented onboard in the near future. The video frame rate is 30Hz. The frames are down-sized to 160×120 images and processed by both the optic flow algorithm running at 30Hz and the texton method running at 13Hz ($n = 30$ and $s = 100$). The execution frequencies have been measured on a dual-core 2.26GHz laptop with all ground station software running at the same time.

While the height of the DelFly is controlled by a human pilot, the rudder is entirely controlled by the ground station. The algorithm for the rudder control is illustrated with a flow-chart in Fig. 6. In case of a collision detection, the control of the rudder is straightforward: it performs a sequence of open loop commands for ~ 1.5 seconds. The sequence is always executed fully before new commands are allowed. When not executing a turn, a collision is detected if ($\tau_{OF} >= 0$ and $\tau_{OF} < 2$ and $\sigma_{OF} < 2$) or TMG < 2.2. In order to prevent the DelFly from getting into a downward spiral, collision detections only result in turning behavior if the absolute yaw rate ($|\dot{\psi}|$) is lower than 12°/sec. The yaw rate is estimated by means of the median horizontal flow f_x as follows: $\dot{\psi} = (f_x/W)\mathrm{FOV}_h$.

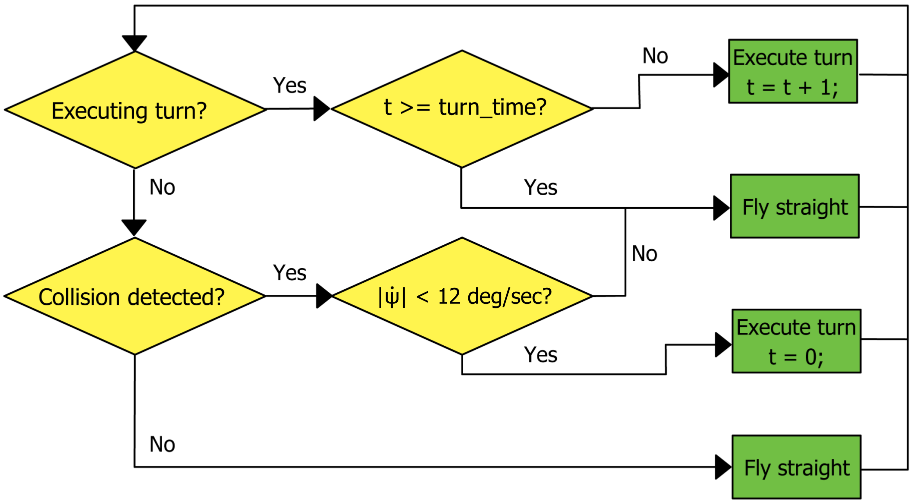

Figure 6: Flow chart of the obstacle avoidance algorithm controlling the DelFly II during the avoidance experiments.

The DelFly successfully avoided obstacles in two different office spaces. Fig. 7 shows the trajectory of an experiment in the office space shown in Fig. 1, from its launch (X) to when it is caught (O). The top part of the figure shows the DelFly's x, y-position during the flight, obtained from two external cameras. The borders indicate the large obstacles such as walls in the room, and delimit the flyable area. The trajectory shows that the DelFly detects all obstacles on time, while not performing an excessive number of avoidance maneuvers. Not every open loop turn has an equal effect. One factor playing a role is that subsequent open loop turns tend to result in sharper turns, but external factors such as drafts also have a big influence on the dynamics of the 16-gram DelFly.

The middle plot shows the variables computed on the basis of the onboard images during the entire experiment. The x-axis shows the time in seconds. The y-axis shows the values of the individual variables relevant to obstacle avoidance: τ_{OF} (dark blue solid), σ_{OF} (red dash dotted), TMG (green solid), and the absoluate yaw rate $|\dot{\psi}|$ scaled by a factor 0.1 for visualization purposes (light blue dashed). The corresponding thresholds are shown with dotted lines in the same colors. Finally, the plot also shows the command c sent to the rudder (purple solid), for flying straight ($c = 0$) or making a turn ($c = -0.8$). Colored crosses in the figure show when the variables satisfy their conditions for an obstacle detection. The black crosses at $y = 0$ shown when a collision is detected *and* $|\dot{\psi}|$ allows for a new open loop turn.

The bottom plot focuses on a time interval in which the DelFly performs two subsequent open loop turns. The first turn is the consequence of a false positive detection by the optic flow measurements. The turn directs the DelFly towards a wall, necessitating collision avoidance. The wall has little texture and is first detected on the basis of the appearance variation cue, leading to a timely turn away from the wall. The onboard image at the moment of detection is shown in the figure with a yellow border. Videos and additional material concerning the experiments are available at http://www.bene-guido.eu/guido/.

5 Conclusions

Our investigation shows that for indoor environments the appearance variation generally decreases when approaching an obstacle. Still, it can increase when the obstacle has a detailed texture, implying that the cue cannot be used alone to detect all obstacles. Our main conclusion is that the appearance variation cue is a useful complement to optic flow: the AUC of a classifier using both cues is higher than that of either cue alone. The performance gain is larger for more degraded images, making the cue a valuable asset for autonomous obstacle avoidance of light-weight flapping wing MAVs. Future work includes onboard processing and autonomous altitude control.

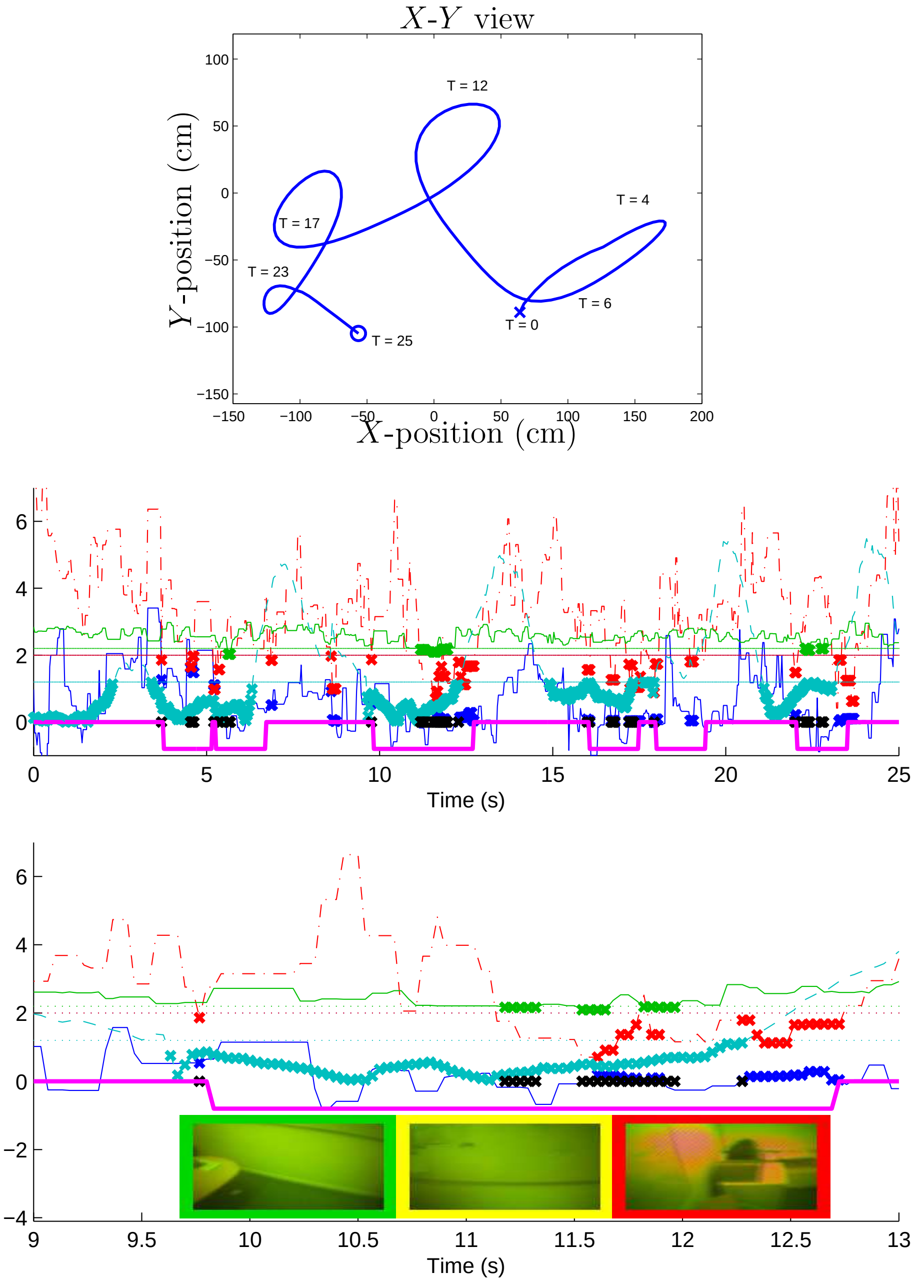

Figure 7: Top: trajectory of the DelFly during one of the experiments. Middle: all relevant variables during the entire experiment. Bottom: two obstacle avoidance maneuvers, the first triggered by optic flow and the second by the appearance variation. See the text for further details.

References

[1] Aerovironment inc.: Nano hummingbird - flapping wing mav. http://www.avinc.com/nano, 2011.

[2] S. Ahrens. Vision-based guidance and control of a hovering vehicle in unknown environments. Master's thesis, MIT, 2008.

[3] H. Aono, S.K. Chimakurthi, C.E.S. Cesnik, H. Liu, and W. Shyy. Computational modeling of spanwise flexibility effects on flapping wing. In *47th AIAA Aerospace Sciences Meeting*, 2009.

[4] A. Bachrach, R. He, and N. Roy. Autonomous flight in unstructured and unknown indoor environments. In *EMAV, the Netherlands*, 2009.

[5] S.S. Baek. Autonomous ornithopter flight with sensor-based behavior. Technical Report UCB/EECS-2011-65, 2011.

[6] A. Beyeler, J-C. Zufferey, and D. Floreano. 3d vision-based navigation for indoor microflyers. In *ICRA 2007, Italy*, pages 1336–1341, 2007.

[7] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based MAV navigation in unknown and unstructured environments. In *IEEE International Conference on Robotics and Automation*, 2010.

[8] Jean-Yves Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker. description of the algorithm, 2000.

[9] K. Celik, S.J. Chung, and A. Somani. Mono-vision corner slam for indoor navigation. In *(EIT 2008)*, pages 343–348, 2008.

[10] Cheng-Lin Chen and Fu-Yuen Hsiao. Attitude acquisition using stereo-vision methodology. In *IASTED Conference*, 2009.

[11] T.S. Collett. Insect vision: Controlling actions through optic flow. *Current Biology*, 12:615–617, 2002.

[12] A.J. Davison and D.W. Murray. Simultaneous localisation and map-building using active vision. *IEEE PAMI*, 2002.

[13] G.C.H.E. de Croon, K.M.E. de Clerq, R. Ruijsink, B. Remes, and C. de Wagter. Design, aerodynamics, and vision-based control of the delfly. *International Journal on Micro Air Vehicles*, 1(2):71 – 97, 2009.

[14] G.C.H.E. de Croon, E. de Weerdt, C. de Wagter, B.D.W. Remes, and R. Ruijsink. The appearance variation cue for obstacle avoidance. In *ROBIO*, 2010.

[15] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 2006.

[16] N. Franceschini, J.M. Pichon, C. Blanes, and J.M.Brady. From insect vision to robot vision. *Philosophical Transactions: Biological Sciences*, 337(1281):283–294, 1992.

[17] S. Grzonka, G. Grisetti, and W. Burgard. Towards a navigation system for autonomous indoor flying. In *(ICRA 2009), Kobe, Japan*, 2009.

[18] A.M. Hyslop and J.S. Humbert. Autonomous navigation in three-dimensional urban environments using wide-field integration of optic flow. *AIAA Guidance, Control, and Dynamics*, pages 147–159, 2010.

[19] F. Iida. Goal-directed navigation of an autonomous flying robot using biologically inspired cheap vision. In *(ISR 2001)*, 2001.

[20] L. Kaufman. *Sight and mind*. NY: Oxford University Press, 1974.

[21] T. Kohonen. *Self-Organizing Maps*. Springer, 2001.

[22] S. Leven, J.-C. Zufferey, and D. Floreano. A minimalist control strategy for small UAVs. In *(IROS 2009)*, pages 2873–2878, 2009.

[23] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging understanding workshop*, pages 121–130, 1981.

[24] F. Ruffier and N.H. Franceschini. Aerial robot piloted in steep relief by optic flow sensors. In *(IROS 2008)*, pages 1266–1273, 2008.

[25] C.E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[26] S. Shen, N. Michael, and V. Kumar. 3D estimation and control for autonomous flight with constrained computation. In *ICRA*, 2011.

[27] N. Takeda, M. Watanabe, and K. Onoguchi. Moving obstacle detection using residual error of FOE estimation. In *IROS*, pages 1642–1647, 1996.

[28] M. Varma and A. Zisserman. Texture classification: are filter banks necessary? In *(CVPR 2003)*, volume 2, pages 691–698, 2003.