Proceedings of the 2010 IEEE
International Conference on Robotics and Biomimetics
December 14-18, 2010, Tianjin, China

The appearance variation cue for obstacle avoidance

G.C.H.E. de Croon, E. de Weerdt, C. de Wagter, and B.D.W. Remes*

***Abstract*— A visual cue is introduced that exploits the visual appearance of a single image to estimate the proximity to an obstacle. In particular, the *appearance variation cue* captures the variation in texture and / or color in the image, and is based on the assumption that there is less such variation when the camera is close to an obstacle. *Random sampling* is applied in order to evaluate the appearance variation fast enough for use in robotics. It is demonstrated that the randomly sampled appearance variation cue can be complementary to optic flow for obstacle detection; combining the two visual cues leads to better obstacle detection performance. Random sampling leads to sufficient computational efficiency for the cue's utilization in autonomous flight: a speed-up of a factor ∼100 is attained, which allows the successful control of the 16-gram flapping wing MAV DelFly II.**

I. INTRODUCTION

While the field of Micro Aerial Vehicles (MAVs) has seen a tremendous progress in the last decades, even basic capabilities such as obstacle avoidance remain hard to attain. Biologically inspired, light-weight MAVs such as the DelFly II (see Fig. 1) cannot carry sensors as heavy as a miniature laser range finder to achieve autonomous flight [1], [2]. Instead, they typically carry a camera onboard. At the moment, there are two main approaches for achieving indoor flight on the basis of onboard camera images.

The first approach accurately estimates the state of the MAV (3D position and attitude). Such a state estimate can be obtained by 'matching' camera images to known locations in a 3D-model of the environment [3], [4], [5], [6]. However, the algorithms still have problems with drift [6] and are still 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 incoming visual inputs [7], [8]. Generally, optic flow is used [9], [10], [11], since it has been shown to play an important role in insect flight [12], [13]. However, both the optics and the optic flow algorithms of flying robots are inferior to their natural counterparts. As a consequence, autonomous flight with optic flow is limited to environments with sufficient texture.

While optic flow is commonly used in efforts for reaching autonomous flight, image appearance has been largely neglected. Image appearance features could be useful for autonomous indoor flight, since they can capture information complementary to optic flow. For example, the absence of texture (a fail-case for optic flow) can be successfully detected by extracting appearance features. The little interest in appearance features is mainly due to the fact that their extraction is computationally expensive.

* MAV-lab, Control and Simulation, Aerospace Engineering, Technical University of Delft. g.c.h.e.decroon@tudelft.nl

Fig. 1. External image of the 16-gram flapping wing MAV DelFly II during the obstacle avoidance experiment.

The **first contribution** of this article is the introduction of a novel appearance cue for obstacle detection. The cue captures the variation in texture and / or color in the image, and is based on the assumption that there is less such variation when the camera is close to an obstacle.

The **second contribution** of the article is the investigation of *random sampling* for significantly reducing the computational effort of extracting image appearance features. The higher computational efficiency comes at the cost of a lower accuracy.

The remainder of the article is organized as follows. In Section II, we investigate the novel appearance cue. In Section III we present a preliminary analysis on the strategy of random sampling. In Section IV it is shown that the novel cue is complementary to the time-to-impact determined by optic flow. In Section V, the flapping wing MAV DelFly II autonomously avoids obstacles using the appearance cue and optic flow. Finally, we draw conclusions in Section VI.

II. APPEARANCE VARIATION AS A CUE FOR OBSTACLE PROXIMITY

Humans use various cues to determine distances to objects in their environment. Monocular visual cues include (cf. [14]): apparent motion as in optic flow, perspective cues such as parallel lines, occlusion, image size of familiar objects, aerial perspective (distance fog), accomodation of the lens, blur, texture gradient, shadows, and image height cues. Apart from a few exceptions (e.g., [15]), roboticists have focused on apparent motion and perspective cues, since these can be extracted with less difficulty and computational effort.

Here, a novel cue for estimating obstacle proximity is introduced for use in robotics. It is termed the *appearance*

978-1-4244-9318-0/10/$26.00 © 2010 IEEE

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 and colors 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 verification of the expected complementarity with optic flow is performed in the Section IV-B. In the remainder of this section, we verify whether the appearance variation indeed decreases towards impact.

A. Measuring appearance variation

For measuring the appearance variation, the term *appearance* is interpreted as textures and / or colors. The approach to estimating the *variation* of these properties is to first estimate the probability distribution of the occurrence of different textures and / or colors in the image. Subsequently, the Shannon entropy [16] of the estimated probability distribution is calculated. Given a discrete probability distribution p with probabilies for n different 'events' p_i, the Shannon entropy $H(p)$ expressed in bits is:

$$H(p) = -\sum_{i=1}^{n} p_i \log_2(p_i), \tag{1}$$

where $p_i \log_2(p_i) = 0$ for $p_i = 0$. A high entropy corresponds to a high variation in appearance, while a low entropy corresponds to the contrary.

For determining the texture / color distribution, the *texton* method [17] 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. In addition, it is amenable to the random sampling approach explained in Section III. 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. The samples are clustered by means of a Kohonen network [18].

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. 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 texton occurrence in the image, with $\hat{p}_i = g_i/s$. This estimate is inserted into (1) to determine the texture variation.

Two aspects of the method are worth mentioning. First, typically all possible local samples are extracted from the image, making $\hat{p}$ equal to p. Second, the texton method captures only texture when it is applied to black-and-white images, but it captures both texture and color when it is applied to color images. In the latter case, the textons actually have the dimension $w \times h \times 3$. In this article, we employ a gray-scale dictionary, leaving color for future work.

B. Experimental setup

Three types of experiments are performed. First, we determine the appearance variation for a set of obstacle approach sequences made with an analog wireless camera. Second, a larger set of approach sequences is studied, which is captured with the camera of a 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) threedimensional visual effects and (b) realistic noise conditions.

For the actual approach sequences, a set of 10 videos was made by holding the analog wireless camera by hand and a set of 65 videos was made in the same manner but then with the mobile phone. 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. The top row of Fig. 2 shows three shots from one of the sequences with the wireless camera.

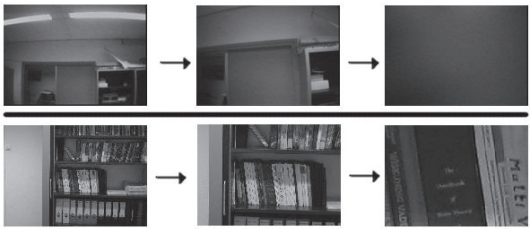

Fig. 2. Example images from an actual approach sequence captured with a wireless camera (top) and a simulated approach sequence (bottom).

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 resolution of 3072×2304 pixels. The simulated camera has a resolution of 160×120 pixels. At the start of the approach sequence the camera captures an area of 2000×1500 pixels, which is resized to the camera resolution with bicubic sampling. At the end of the approach, the captured area equals the camera resolution. Digitally zooming in even further would always result in a lower appearance variation, since no new details can come into sight. Each sequence

[1]All parameter settings will be mentioned in Subsection II-B.

comprises 90 frames during which the width and height of the captured image area decreased linearly. The bottom row of Fig. 2 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 it is learned on a separate set of images not belonging to the approach sequences.

C. Results

Fig. 3 shows the entropy values of TMG over time (thin gray lines) for the 10 wireless video sequences. The x-axis represents the time, while the y-axis shows the entropy. It also shows linear fits (thick black lines), which minimize the mean square error to the data points.

As can be seen in Fig. 3, 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.*

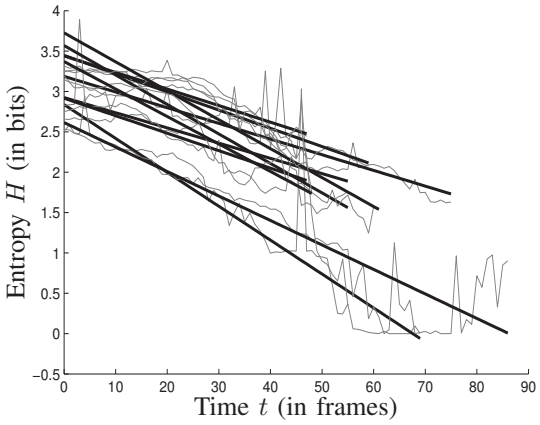

Fig. 3. The entropy of TMG over time (light gray lines) and corresponding linear fits (thick black lines). The entropy decreases towards impact.

In some sequences the entropy increases over time. Investigation of these sequences showed that they resemble the one in the bottom row of Fig. 2. The simulated camera approaches a book shelf that has objects in it with different, highly detailed textures and colors. 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. 3) suggest that the sole use of the appearance variation cue will not lead to perfect obstacle avoidance.*

III. RANDOM SAMPLING

By extracting a random subset of all local image samples the texton method can become computationally much more efficient. In this section, a preliminary theoretical analysis is presented on the effects that such random sampling has on the computational effort (Subsection III-A) and on the accuracy (Subsection III-B) of the texton method. We note that the comparison of random sampling with other methods of selecting the samples falls outside the scope of this article.

A. EFFECT ON COMPUTATIONAL EFFORT

The computational effort c of the texton method is approximately:

$$c \approx snW + nC, \tag{2}$$

where W is the cost of comparing an extracted sample to a texton, and nC is the cost of calculating the entropy. During execution, n is fixed, but s can be varied freely. In our implementation, the samples are extracted at uniformly distributed random positions in the image.

To illustrate the effect random sampling has on the computational effort, consider that a (relatively small) image of 160×120 pixels contains 17825 possible 5×5 pixel samples. This while extracting only 100 such samples already leads to quite accurate estimates of the texton distribution in the image (see Section III-B). *Since the bulk of the computation of the texton method is in the term snW in (2), random sampling easily leads to a speed-up of a factor ∼100 with respect to full sampling.*

Fig. 4 shows the mean execution frequencies (solid line) and corresponding standard deviations of a MATLAB implementation of TMG on a data set of 94 images. The MATLAB-implementation used to obtain these results is available online[2]. The number of samples s is varied from 50 to 2500 with steps of 50. For the experiments, $n = 30$ and $w = h = 5$. The processing times are measured on a 2.26 GHz laptop. For obstacle avoidance with indoor MAVs it is reasonable to state that the execution frequency should at least be 1 Hz (dashed line). The figure shows that this frequency is reached by extracting 2250 or fewer samples.

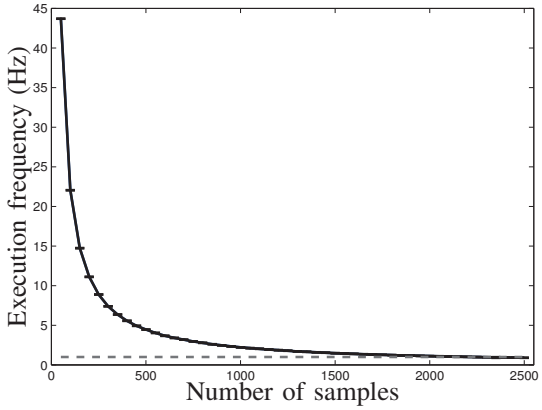

Fig. 4. Mean execution frequencies and corresponding standard deviations of a MATLAB-implementation of TMG for a number of samples s from 50 to 2500 (blue solid line). The 1 Hz limit is shown by a red dashed line.

[2]http://www.bene-guido.eu/guido/

B. EFFECT ON ACCURACY

The main question that arises when using much fewer samples is: *how does random sampling affect the accuracy of the probability estimate?*

Here this question is analyzed in the context of the maximum likelihood estimate of the texton distribution in the image. This distribution is a categorical distribution, and can be fully determined by extracting all samples from the image. Our analysis consists of determining the relation of the number of samples and the expected L1-distance between the maximum likelihood estimate and the actual distribution in the image.

We determine the probabilities for distances between the estimated and actual distributions for the case with resampling. Extracting a fixed number of s samples from random image locations results in texton occurrences $g = \langle g_1, g_2, \ldots, g_n \rangle$. The vector g follows a multinomial distribution and it has the following well-kown probability formula:

$$P(g) = \frac{s!}{g_1! g_2! \ldots g_n!} p_1^{g_1} p_2^{g_2} \ldots p_n^{g_n}, \tag{3}$$

with $\sum g_1 + g_2 + \ldots + g_n = s$. For a given number of samples s, this formula allows one to iterate over all possible vectors g while determining the distance d between the estimated distribution $\hat{p} = g/s$ and the actual distribution p. The probability for distance $d = d(\hat{p}, p)$ can then be incremented by $P(g)$. Iterating over all g permits to calculate $P(d)$, the probability distribution of the distances between the estimated and actual distribution.

Although the above method is not elegant, it is tractable for a limited number of samples, since there are also a limited number of possible distances. To illustrate the effects of random sampling, we apply the method to the categorical distribution with $n = 6$: $p = \langle 0.5, 0.1, 0.1, 0.05, 0.05, 0.2 \rangle$. Fig. 5 shows the distribution $P(d)$ (y-axis) for $s = \{50, 100, 150, 200, 250, 300\}$ (x-axis), where illuminance represents high (white) to low (black) probabilities. The black line indicates the mean distance to the actual distribution and the white dashed line the 95^{th} percentile.

Two main observations can be made from Fig. 5. First, as to be expected, the mean of the distribution gets closer to the actual distribution as the number of samples increases. Second, this effect obeys the law of diminishing returns, so that *a modest number of samples already reduces the probability for 'large' distances considerably*.

When the distribution is known, $P(d)$ can be determined to provide certainty bounds to the distance between the estimated and actual distribution. For example, for the categorical distribution with replacement given above, $P(d < 0.20 \wedge s = 150) = 0.95$. The problem is of course that the actual distribution is not available. Still, we can limit the 95% certainty bound from above by assuming the worst case scenario, which occurs when the entropy of the actual distribution is highest[3]. The relation between entropy and the mean L1-distance is particularly relevant to the appearance variation cue, since it uses the entropy as a measure of obstacle proximity. *Near-collisions are associated with low entropy distributions, which suffer the least from random sampling.*

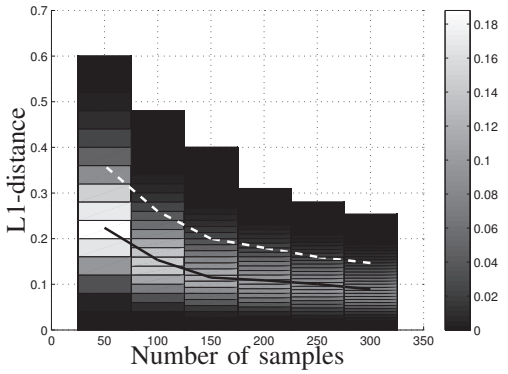

Fig. 5. Relation between the number of samples used in random sampling and the accuracy of the estimate. The y-axis shows the L1-distance between the estimated and the true distribution, the x-axis shows the number of samples s. The intensity represents the probability, where white indicates high probability and black a low probability. The black line indicates the mean distance from the actual distribution, the white dashed line the 95^{th} percentile.

IV. 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 IV-A the optic flow method is explained. In Subsection IV-B the classification performances of the methods are evaluated.

A. OPTIC FLOW FOR CALCULATING TIME-TO-IMPACT

The setting of obstacle avoidance with light-weight MAVs makes computational efficiency and robustness to noisy images the main priorities for the optic flow algorithm. For this reason, simplicity of the optic flow algorithm is preferred over detailed information on the environment. The implementation of the algorithm has 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 [19], [20] from the openCV library is used[4]. The second part is performed as follows. It is assumed that the camera is moving straight towards a flat surface orthogonal to the movement direction, while the camera possibly undergoes pitch and yaw changes. This assumption implies that after removal of the influence of pitch and yaw movements, the Point of Expansion (PoE) is in the center of the image. To estimate the influence of pitch and yaw movements, the median horizontal and vertical flow in the image are determined. These are subtracted from all flow vectors, presumedly leading to the PoE being in the center. Subsequently, each optic flow vector is used to determine the distance from the old location

[3]A formal proof of this matter is beyond the scope of this paper.

[4]http://www.opencv.org/

(x_t, y_t) and the new location (x_{t+1}, y_{t+1}) to the center of the image (x_c, y_c); $d_{c,t}$ and $d_{c,t+1}$ respectively. The difference in distance to the center is $\Delta d_{c,t}$. Each optic flow vector leads to one estimate $\hat{\tau}$:

$$\hat{\tau} = d_{c,t}/\Delta d_{c,t} \quad (4)$$

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 $\sigma_{\tau_{OF}}$. Despite the strong assumptions, this straightforward method works rather well in practice.

B. 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 $< \vartheta_1$', '$\tau_{OF} < \vartheta_1 \wedge \sigma_{OF} < \vartheta_2$', '$\tau_{OF} < \vartheta_1 \vee$ TMG $< \vartheta_2$', and '$(\tau_{OF} < \vartheta_1 \wedge \sigma_{OF} < \vartheta_2) \vee$ 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 omitted from here on.

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

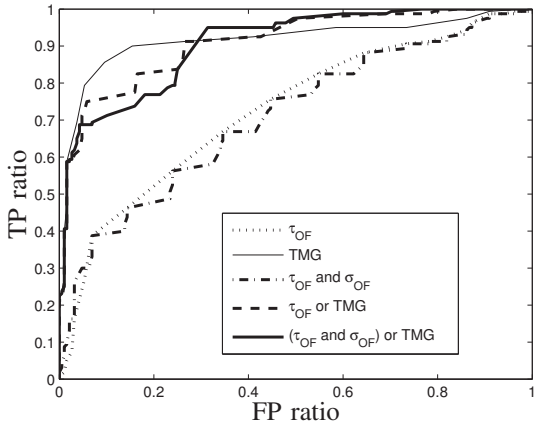

Fig. 6. ROC-curves of different combinations of variables, $\tau^* = 40$, for the video sequences made with the flying DelFly II.

Fig. 6 shows the ROC-curves for 19 video sequences recorded with the flapping wing MAV DelFly II. It includes the methods τ_{OF} (dotted), TMG (thin solid), $\tau_{OF} \wedge \sigma(\tau_{OF})$ (dash-dotted), $\tau_{OF} \vee$ TMG (dashed), and $(\tau_{OF} \wedge \sigma(\tau_{OF})) \vee$ TMG (thick solid) for $\tau^* = 40$ (expressed in frames to impact). The x-axis represents the ratio of false positives, while the y-axis represents the ratio of true positives. The higher the curve, the better. TMG has the lowest FP ratios for a TP ratio smaller than 0.91. For higher TP ratios, the combination of TMG with $\sigma(\tau_{OF})$ and τ_{OF} gives the best results.

TABLE I
AUC FOR COMBINATIONS OF τ_{OF}, $\sigma(\tau_{OF})$, AND TMG. BOLD INDICATES THE HIGHEST AUC. DELFLY II SEQUENCES.

τ^*	τ_{OF}	TMG	$\tau_{OF} \wedge \sigma(\tau_{OF})$	$\tau_{OF} \vee$ TMG	$(\tau_{OF} \wedge \sigma(\tau_{OF})) \vee$ TMG
10	0.787	0.844	0.722	**0.846**	0.824
20	0.727	**0.848**	0.713	0.831	0.810
30	0.775	**0.934**	0.753	0.922	0.905
40	0.726	**0.916**	0.709	0.910	0.906
50	0.682	0.853	0.672	**0.860**	0.850

TABLE II
AUC FOR COMBINATIONS OF τ_{OF}, σ_{OF}, AND TMG. BOLD INDICATES THE HIGHEST AUC. HAND-MADE SEQUENCES.

τ^*	τ_{OF}	TMG	$\tau_{OF} \wedge \sigma_{OF}$	$\tau_{OF} \vee$ TMG	$(\tau_{OF} \wedge \sigma_{OF}) \vee$ TMG
10	0.836	0.881	0.769	0.876	**0.897**
20	0.893	**0.976**	0.848	0.963	0.877
30	0.926	0.927	0.899	**0.942**	0.936
40	0.908	0.819	0.896	0.867	**0.944**
50	0.887	0.733	0.880	0.769	**0.974**

The quality of the classifier can be expressed by the Area Under the Curve (AUC). Table I shows the AUC-values for the different methods for different τ^*. A bold setting indicates the best method. Remarkably, 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 100 samples from each image.

Fig. 7. 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.

On the other image sets such as the hand-held video sequences, optic flow performs on a par with TMG (see Table II). On those sequences, the best results are almost always obtained by a combination of the two methods. Optic flow seems to perform badly on the DelFly video sequences, because the DelFly images are rather degraded. In particular, *the high-frequent flapping movements of the DelFly in combination with the line-by-line recording of the video images lead to image deformations*. Fig. 7 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 such as those in Fig. 7 do not comply anymore with the linear camera model assumed by optic flow algorithms.

V. OBSTACLE AVOIDANCE EXPERIMENT

The final experiment involves the use of both the random sampling method and optic flow to control the flapping wing MAV DelFly II [21] in order to verify whether the accuracy is good enough for successful obstacle avoidance.

The DelFly II is light weight (16 grams) and cannot yet carry its own processing onboard. Therefore, the onboard video images are sent to a ground station that processes them in order to send new control commands. Of course, computational efficiency stays of the essence. If offboard processing is not powerful enough for an algorithm, it is probable that eventual onboard processing will not be either.

The video frame rate is 30Hz. The frames are down-sized to 160×120 images and processed by both the optic flow algorithm and the texton method. The optic flow algorithm runs at 30Hz. The parameters for the texton method are $n = 30$ and $s = 100$.

While the height of the DelFly is controlled by a human pilot, the rudder is entirely controlled by the ground station. The control of the rudder is straightforward: it performs a sequence of open loop commands if either the texton method or the optic flow detects an imminent collision. The sequence is always executed fully before new commands are allowed.

The DelFly successfully avoided obstacles in two different office spaces of approximately 5×5 m with the help of both the texton method and the optic flow. Onboard and external videos are available at http://www.bene-guido.eu/guido/. Fig. 1 is a still from the external video of the first experiment, in which the DelFly avoids a texture-poor wall. Both the texton method and the optic flow algorithm triggered turns of the DelFly during the experiment.

As a result of using the appearance variation cue, the DelFly maintains a slightly larger distance to completely uniform walls than to walls with (some) texture. Although the DelFly could in principle fly further towards these walls without crashing, it will receive no further visual information that reliably indicates at which point the DelFly *should* turn. We regard it as a strength of the algorithm that it avoids such non-informative situations.

VI. CONCLUSIONS

Our main conclusion concerning the appearance variation cue is that for indoor environments the entropy of texture distribution generally decreases when approaching an obstacle. Still, the entropy can increase when the obstacle has a detailed texture. This implies that the appearance cue cannot be used in isolation to detect all obstacles. Nonetheless, the entropy values can be a useful complement to optic flow: the AUC of a classifier using both cues is higher than that of either cue alone. The performance improvement provided by the novel cue increases when the images are more degraded. Furthermore, the results show that random sampling can make the extraction of the appearance variation cue fast enough for use in indoor flight. The computational effort is reduced by a factor $\sim$100, while retaining an acceptable accuracy.

Future work will include a deeper investigation of the appearance variation cue, for example in outdoor environments. Due to the abundance of texture outdoors, we expect the appearance variation cue to work less well and the optic flow to work better in outdoor areas. Still, we expect the combination of the two cues to outperform either one alone.

REFERENCES

[1] A. Bachrach, R. He, and N. Roy. Autonomous flight in unstructured and unknown indoor environments. In *European Micro Air Vehicle conference and competititons, EMAV 2009, the Netherlands*, 2009.

[2] S. Grzonka, G. Grisetti, and W. Burgard. Towards a navigation system for autonomous indoor flying. In *Proc. IEEE International Conference on Robotics and Automation 2009 (ICRA 2009), Kobe, Japan*, 2009.

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

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

[5] K. Celik, S.J. Chung, and A. Somani. Mono-vision corner slam for indoor navigation. In *IEEE International Conference on Electro/Information Technology (EIT 2008)*, pages 343–348, 2008.

[6] 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.

[7] A. Beyeler, J-C. Zufferey, and D. Floreano. 3d vision-based navigation for indoor microflyers. In *2007 IEEE International Conference on Robotics and Automation, Roma, Italy*, pages 1336–1341, 2007.

[8] S. Leven, J.-C. Zufferey, and D. Floreano. A minimalist control strategy for small uavs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009)*, pages 2873–2878, 2009.

[9] F. Iida. Goal-directed navigation of an autonomous flying robot using biologically inspired cheap vision. In *Proceedings of the 32nd ISR (International Symposium on Robotics)*, 2001.

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

[11] 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.

[12] 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.

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

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

[15] A. Negre, C. Braillon, J. Crowley, and C. Laugier. Real-time time-to-collision from variation of intrinsic scale. In *Proc. of the Int. Symp. on Experimental Robotics*, 2006.

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

[17] M. Varma and A. Zisserman. Texture classification: are filter banks necessary? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, volume 2, pages 691–698, 2003.

[18] T. Kohonen. *Self-Organizing Maps, third edition, Springer Series in Information Sciences, Vol. 30*. Springer, 2001.

[19] 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.

[20] Jean-Yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker. description of the algorithm, 2000.

[21] 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.