Off-board processing of Stereo Vision images for Obstacle Avoidance on a Flapping Wing MAV

S. Tijmons*, G.C.H.E. de Croon, B.D.W. Remes, C. De Wagter, R. Ruijsink, E. van Kampen and Q.P.Chu
Delft University of Technology, Delft, The Netherlands

December 11, 2014

Abstract

One of the focuses of Unmanned Air Vehicle research is to develop insect-scale vehicles, which can operate in environments inaccessible to humans. For many applications it will be a key requirement that these vehicles can operate fully autonomously. This implies that the vehicles have the capability to avoid obstacles. Research on this task for Flapping Wing Micro Air Vehicles (MAVs) has been limited to single camera techniques such as optic flow calculation and appearance variation extraction. In this paper the first experiments are described where stereo vision onboard a light weight Flapping Wing MAV with off-board processing is tested for the purpose of obstacle avoidance. An advantage of stereo vision over single camera techniques is that it provides distance measurements instantly. Furthermore Flapping wing MAVs will play an important role in future research on miniaturizing MAVs. The research described in this paper demonstrates the successful implementation of a stereo vision system with off-board processing on the 21-gram DelFly II for autonomous obstacle avoidance.

1 Introduction

Autonomous flight of Micro Air Vehicles (MAVs) concerns many aspects of guidance, navigation and control which are main areas of interest within aerospace research. Therefore a fair amount of studies has been conducted with actual MAV platforms dealing with advanced methods in these areas. However, many of these studies involve relatively heavy platforms, with a weight typically larger than 500 grams. This allows the usage of heavy and often energy-consuming

*All authors are with the Micro Air Vehicle laboratory of the Faculty of Aerospace Engineering, Delft University of Technology, 2629 HS Delft, The Netherlands. Contact: s.tijmons@tudelft.nl, microuav@gmail.com

sensors such as radars and laser scanners which provide long range and millimeter precision information on the environment (Griffiths et al.[1]). These examples form a large contrast with previous studies concerning light weight Flapping Wing MAVs (FWMAVs) where the sensor payload weight is limited to only a few grams. Duhamel et al.[2] describes experiments with the *RoboBee*, a 101-milligram FWMAV equipped with an optic flow sensor. Using external power and off-board processing, the altitude is controlled by pointing the sensor to a highly-textured screen. Baek et al.[3] demonstrated a 13-gram FWMAV equipped with an infrared sensor to track a specific infrared target point. Processing is performed onboard, including 3-axis gyroscope feedback for attitude control. Over 20 trials the vehicle was able to fly to the target with a success rate of 85%. Garcia Bermudez et al.[4] describes experiments on a 7-gram FWMAV equipped with a camera that stores heavily downsampled images during a short period of a flight. The images are uploaded to a pc afterwards to perform optic flow computations. This way it was demonstrated that there is a large coupling between the body motion due to flapping and the measured optic flow.

Several experiments with the *DelFly II* regarding autonomous tasks have been performed as well. De Croon et al.[5](2009) describe experiments where onboard camera images are sent to a ground station that performs height calculation and path following. Based on these measurements control signals are determined and sent to the vehicle. Also obstacle avoidance experiments have been performed using this setup, as described by De Croon et al.[6](2012a) and De Croon et al.[7](2012b). The ground station performed a combination of optic flow computation and appearance variation measurements. Appearance variation was proposed as a new image cue that bases a proximity estimate on the variation in texture in the image and was demonstrated as a useful addition to optic flow measurements for the task of obstacle avoidance.

The experiments with the DelFly II showed that the use of optic flow measurements while performing off-board processing has a limited success. The main reason for this is the relatively low frame rate (30 FPS) of the video stream which results in significant image distortions and large changes in camera attitude between frames (due to the flapping motion). Compensation for these effects is theoretically possible using 3-axis gyroscope feedback. This would require all processing to be done onboard, but the available processing power was not sufficient at that time. Also image noise due to analog transmission has a negative influence on optic flow measurement accuracy.

In this paper stereo vision is proposed as an alternative to optic flow to get around these issues. While optic flow requires a sequence of images from different moments in time, stereo vision can perform distance measurements based on a single pair of left and right images. The frame rate of the transmitted camera images does not play a role in this case, and obstacle information is obtained instantly.

In Section 2 the stereo vision method used in this study is discussed. Section 3 describes the DelFly II and the system overview used during experiments. The performance of the used system is reviewed in Section 4. The obstacle avoidance strategy implemented in this study as well as results from experiments

are discussed in Section 5. The Conclusions are drawn in Section 6.

2 Stereo Vision method

Stereo vision is a technique to extract depth information about the environment from two or images taken from different and known positions. By matching corresponding features in these images, the shift in pixel position of these features is determined, which is called the disparity. If the camera characteristics and positions are known, the disparity value is used to compute the metric position of the feature relative to the camera. By matching all image features, a dense 3D reconstruction of the environment can be made for further purposes such as navigation.

Stereo vision is still a widely studied area of image processing and this has led to a large variety of methods. A recurring aspect of stereo vision is the tradeoff between the quality of the disparity maps and the computational efficiency of the method. These conflicting requirements are the source of a wide range of proposed methods. Good overviews of stereo vision methods are presented by Scharstein&Szeliski[8] and Tombari et al.[**?**]. In this section it is discussed in which groups the methods can be divided and how these relate to the purpose of this study.

In this study stereo vision will be used as a source for obstacle detection on a flying platform. Therefore the main requirement for the stereo vision method is that it has real-time performance. Real-time is defined here as close to the actual frame rate of the video stream, which is 25 FPS in this study. This requirement can be met in two ways. The first way is to use special hardware dedicated to perform optimal for certain image processing routines. In literature it is very common to use implementations on Graphical Processing Units, Field Programmable Gate Arrays, Application-Specific Integrated Circuits and Digital Signal Processors. These types of implementations are in general not applicable to systems with a very small weight and limited power supply. Since the aim of this study is to obtain autonomous behavior of an MAV, it is chosen to focus on implementations that have the potential of being used onboard in future studies. Therefore the chosen stereo vision system should perform in real-time on a normal CPU via another approach: by using extremely efficient algorithms. In other words, a stereo vision method should be used that forms a good tradeoff between accuracy and computational complexity.

Stereo vision algorithms can be roughly divided in three different groups. The first group is formed by the Local methods, where image regions with limited size are tried to be matched based on a cost function. The methods are in general very efficient but lack quality when the variation of texture in the stereo images is low. On the other side there is the group of Global methods, that try to create a disparity map that optimizes a cost function for the complete image. This requires in general a large amount of processing steps and also puts a high demand in required memory. These methods mainly focus on quality only, while real-time performance is neglected. The group of Semi-Global methods forms

a bridge between the other two groups. These methods do use an optimization strategy but in a limited way. For example, some methods optimize the cost function only for a single image line. This group of methods contains the most efficient stereo vision algorithms and was therefore further studied.

The best candidate for implementation on our FWMAV system was found to be the Semi-Global Matching method proposed by Hirschmuller[9]. The method performs optimization along different directions: horizontally, vertically and diagonally. According to literature Semi-Global Matching represents a good trade-off between computational efficiency and performance (Banz et al.[10], Gehrig&Rabe[11], Chen&Chen[12]).

An OpenCV[1] implementation of this method was used in this study. The camera images are subsampled to 160×108 pixels to obtain a processing rate of 25 FPS. To illustrate the efficiency of the algorithm a comparison is made with two other stereo vision methods as illustrated in Figure 1. An example image from the stereo camera used in this study is used in this comparison. The figure shows the disparity result from three different methods: Block Matching , Dynamic Programming (based on Birchfield&Tomasi[13]), and Semi-Global Matching.

Block Matching is a method that does not perform optimization, but purely tries to match image windows. The matching cost for different disparity candidates is compared, and the disparity resulting in the lowest cost is chosen. Dynamic Programming is a method that performs optimization in 1D, along the image line only. These two methods are less computationally demanding compared to the Semi-Global method.

The top-right image of Figure 1 shows the result from the Block Matching method. The bright pixels indicate that image features are close by, dark pixels indicate that the features are further away. Black means that no good match could be made. This example illustrates that local methods perform very bad for image regions with a small variation in texture. Only the image parts that contain clear lines are matched correctly.

The bottom-left image of Figure 1 shows that the 1D optimization strategy of the Dynamic Programming method is not sufficient in many cases and may lead to very bad disparity estimates. This is due to the fact that wrong disparity estimates at one point along the optimization line have a negative effect on the other estimates along that line. This results in the striping effect that is visible in the disparity image.

The bottom-right image of Figure 1 shows the result from Semi-Global Matching. It is clear that this method outperforms the other two example methods significantly. The amount of pixels that has been matched is larger, and the structure of the background is much better indicated by this method.

[1] www.opencv.org

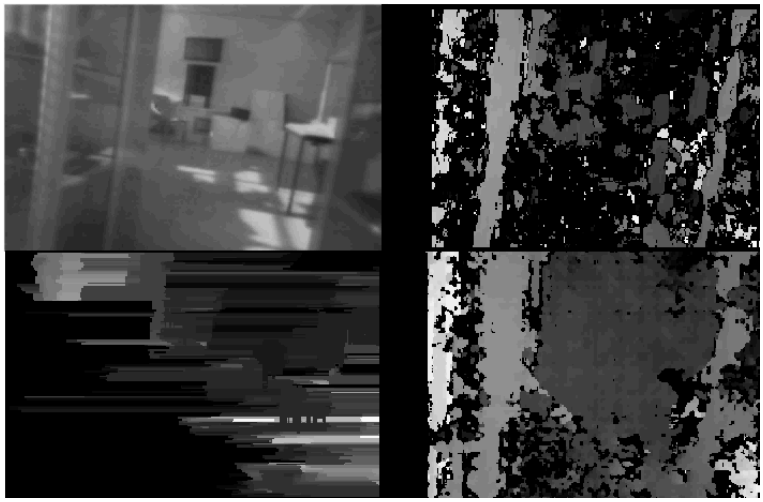


Figure 1: Comparison of three different types of stereo vision methods. **Top-Left:** test image from stereo vision camera. **Top-Right:** Block Matching. **Bottom-Left:** Dynamic Programming (1D Optimization) **Bottom-Right:** Semi-Global Block Matching (Multi-Dimensional Optimization)

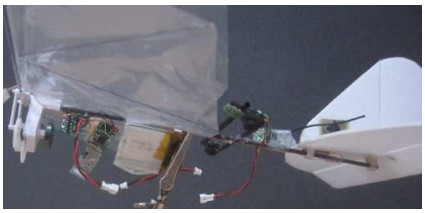
3 DelFly II System Design

The experiments with stereo vision in this study are performed on the DelFly II, a 17-gram flapping wing MAV with a wing span of 28cm. It is characterized by its biplane wing configuration and its standard T-tail, as shown in Figure 2. De Croon et al.[7] describes the design of the DelFly II in more detail.

Apart from these features the DelFly II can fly in different payload configurations. In this study it is equipped with two devices. The first one is a small autopilot board that allows a ground station to communicate to the DelFly via an RC link. It also features 2-axis gyroscopes that provides feedback to the tail rudder for yaw stabilization. Also a barometer is present which can be used for height control.

The other payload device is the stereo vision camera system, which will be described in more detail in the next paragraph. The camera system also contains an analog transmitter to send the stereo images to the ground. The ground station is a 2.30 GHz dual-core pc that receives the image stream via an analog/digital converter. The Semi-Global Matching stereo method discussed in Section 2 is processed on the ground station. Also the control strategy routines for obstacle avoidance that will be discussed in section 5 are performed on the ground station. The resulting control signal is sent to the DelFly via an RC link. Figure 3 shows how the different components interact with each other.

The DelFLy II in the current configuration has a total weight of 21.1 grams. This is much more than the typical weight of 17 grams from previous studies, due to the stereo vision system that has a total weight of 5.2 grams. This



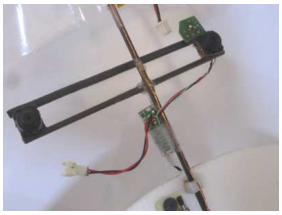


Figure 2: Top: The DelFly II. Bottom: detailed view of the stereo camera system with transmitter

includes an extra battery of 1 gram to power the stereo vision system. Due to this relatively heavy weight the DelFly is not able to hover. In this study the forward speed is chosen to be approximately 0.6m/s. At this speed stable forward flight is possible that allows the DelFly to stay in the air for several minutes. At the same time this speed is still sufficiently low to perform indoor obstacle avoidance.

Stereo Vision System The stereo vision system onboard the DelFly is the main sensor since it enables the vehicle to perform obstacle avoidance. It is visualized in detail in the right image of Figure 2. The stereo system contains two synchronized CMOS 720×x480 cameras with a baseline of 7.6cm and a 2.4GHz NTSC transmitter that transmits at a rate of 25 FPS. The horizontal field of view of the cameras is ±60 degrees. The image streams from the left and right cameras need to be merged into one image stream for transmission. In an initial setup this was done by switching between cameras after each half frame. In the NTSC format the even and odd lines are scanned consecutively: first all even lines are scanned, then all odd lines. In the stereo implementation the even lines would come from one camera, the odd lines from the other camera. Since switching is performed every half frame, the time offset between the left and right images is then approximately 20ms. This is a significant time difference when dealing with the fast body motions of a flapping wing vehicle. Experiments demonstrated that the stereo matching routine suffered significantly from the difference in image distortions due to the vehicle motion.

It was therefore decided to implement the stereo frame merging in a different way. Switching between cameras is done after each individual line, which reduces the time offset by a factor of 240 to approximately 83μs. This way the distortion

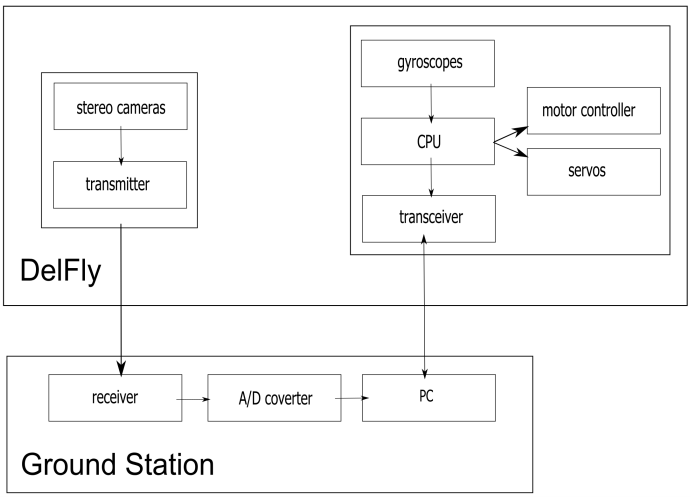


Figure 3: Diagram of the interaction between all system components

of the images themselves is not solved, but the distortion between the left and right frames is almost the same.

The effect of this change was measured by performing a small test. The disparity map as obtained by the ground station was compared for the two implementations under two conditions. During the first condition, the camera is pointed at a static scene. During the second condition, the scene observed by the camera is moving randomly left and right to imitate camera motion.

The results from this test are shown in Figure 4. The top plot shows the measurements using the initial implementation. After 135 frames, the observed scene starts to move. The effect on the disparity measurements is clearly significant. The left and right motion of the scene can be observed in this data by the sinusoidal motion. The bottom plot shows that for the chosen implementation this effect is reduced to additional noise only.

4 Stereo Vision System Performance

The performance of the stereo vision system as implemented on the DelFly II in this study is evaluated by two experiments. The first experiments measures the accuracy of the stereo vision system in measuring the distance to obstacles in the case the camera and the scene are static. This was done by pointing the stereo vision camera to a screen with a large chess board pattern. The distance based on the disparity map computed at the ground station is averaged over 1100 frames and this is repeated at different distances to the screen. Figure 5 shows the result from this test. The left plot shows the mean and standard deviation

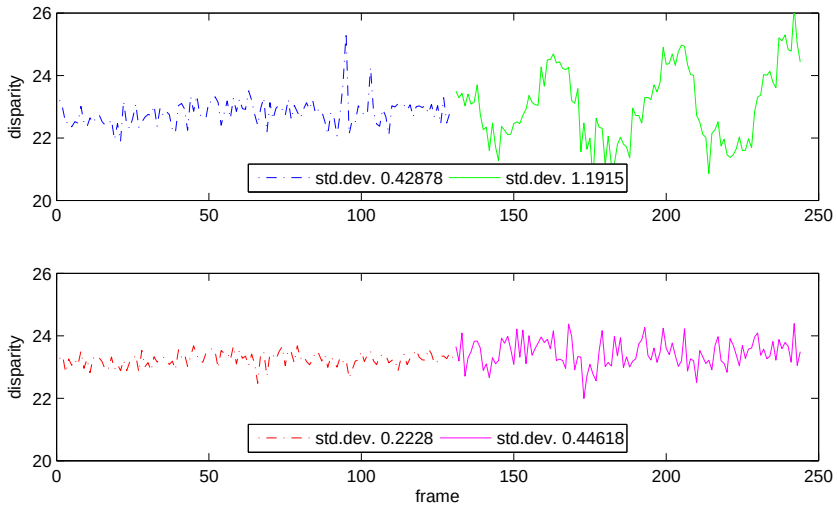


Figure 4: Comparison between the camera reading methods. **Top** initial method **Bottom** implemented method. During the first 135 frames there is no motion (dash-dotted lines), further on there is a relative motion between the camera and the chessboard (solid lines).

for 7 measurement points up to a distance of 5m. The right plot shows the same measurements, as well as another measurement at 6m. From these results it can be concluded that the accuracy of the distance measurements up to 5m has an average error of less than 50cm. This is regarded as sufficient for the task of obstacle avoidance. Note that at smaller distances the measurement accuracy is even higher.

The experiment has been repeated in the form of a flight test with the DelFly. This way the effect of the body motion of the vehicle on the distance measurements is taken into account. The DelFly was flying in the direction of the same screen as used in the previous experiment, starting at a distance of 4.5m. A ground truth distance was obtained by tracking the position of the DelFly with two external cameras.

The results are presented in Figure 6. The left plot shows several measurement points indicated as outliers as they are results from corrupted frames (frames where some synchronization error of the cameras resulted in a swap of the left and right camera). The results in the right plot show that the accuracy during flight is fairly close to that of the static case. The increase in standard deviation due to camera motion was also noticed in Figure 4 in the previous section. Figure 6 shows that the flight test measurements follow the same trend in standard deviation as for the static case, including the dimple at 3m. For distances larger than 4m the accuracy seems to increase significantly. This is partly caused by an initial offset in the flight direction of the DelFly during the

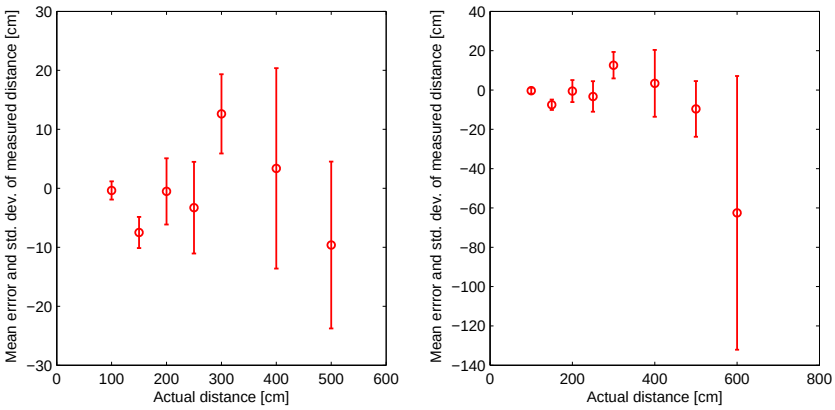


Figure 5: Distance measurement accuracy for the static case. The left plot is a detailed version of the right plot.

first second of the test. This can be seen in the left plot.

5 Obstacle Avoidance

The main focus of this study is to test the stereo vision system implementation as described in the previous sections for the task of obstacle avoidance. A few preliminary tests were performed first to determine requirements for a robust avoidance strategy.

A straight forward approach is a simple reactive controller that checks if there are obstacles at a distance below a predefined threshold. If this is the case, the vehicle will turn left if the obstacle was detected on the right side of the image, and vice versa. This strategy proved to be not robust since situations can occur where this will lead to wrong turn decisions. If the vehicle flies in the direction of a corner it will not turn away from it. Furthermore, the cameras have a limited field of view of 60 degrees. By using this reactive approach, an avoidance maneuver can steer the vehicle in the direction of an obstacle that was out of sight.

The main reason for these issues is the limitation of the DelFly in that it can only fly with a minimal forward speed. A simple test was performed where the DelFly would use its elevator to pitch up 90 degrees. This reduces the forward speed to hover condition, such that the DelFly can turn around without hitting obstacles on the side. This approach proved to be successful. However, the DelFly will descent quickly in these cases because it does not provide sufficient thrust while hovering.

Therefore a new strategy for obstacle avoidance was tested that takes into account these limitations and requirements. In this strategy it is assumed that the DelFly is flying at a constant forward speed. As mentioned earlier, the

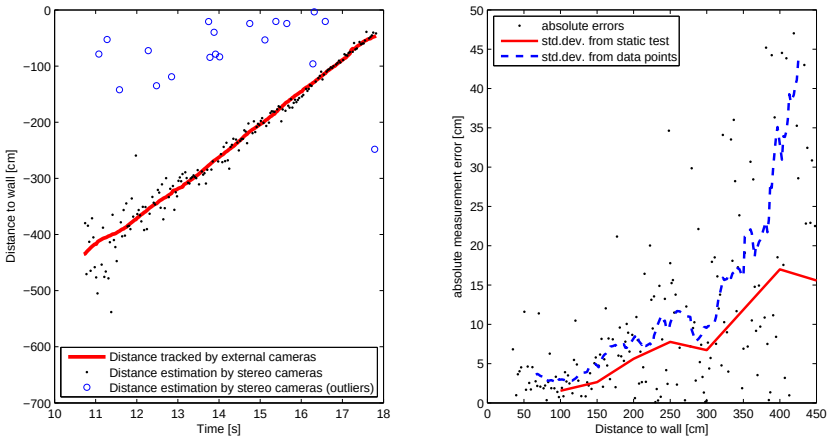


Figure 6: Distance measurement accuracy for the flight test. The left plot shows the actual distance and estimated distance over time. The right plot shows the estimation error with reference to the actual distance.

forward speed of the DelFly is controlled by an elevator, which is part of the tail. In Figure 2 the DelFly is shown while having a small pitch angle. It can also be seen that the elevator surface is deflected downwards. The DelFly would fly with fast forward speed in this condition. For the slow forward flight condition that is needed for obstacle avoidance, the elevator is deflected up, which results in high pitch angles of around 80 degrees. The speed of the DelFly is fairly constant in this case, but any type of wind disturbance has a significant effect on the speed. For this reason tests were performed in a closed room.

Based on the assumption of a constant forward speed, a safety region in front of the DelFly is defined. It is illustrated in the left plot of Figure 7. The safety region is defined in such a way that the DelFly can safely perform an avoidance maneuver within the boundaries of this area. If the system continuously assures that there is no obstacle within this region, the DelFly does not need to avoid anything. If at some point an obstacle is detected at the boundary of the safety area, it is assured that if the DelFly will perform its avoidance maneuver as defined by the safety region, it will have a guaranteed collision-free escape route.

The safety region as shown in Figure 7 is composed in the following way. Since it is not known beforehand in which direction the DelFly will continue after an avoidance maneuver, it is required that the vehicle can make a 360 degrees turn within the safety region. Due to the minimum forward speed of 0.6m/s, the radius of this turn was measured to be 50cm. On top of that the wing span of the DelFly (28cm) needs to be taken into account, as well as the uncertainties of the distance estimates that were evaluated in Section 4. At a distance of 3m, an uncercainty of 20cm is regarded as a safe choice. The total

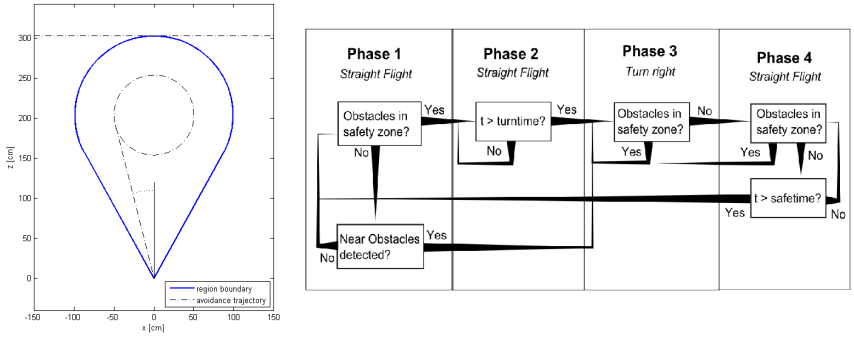


Figure 7: **Left:** Definition of the safety zone, the area in front of the DelFly that needs to stay free of obstacles. **Right:** Turn logic that defines whether the DelFly needs to perform an avoidance maneuver.

radius is a combination of these sizes, and this radius defines the upper part of the safety region as in Figure 7. The lower part of the region is defined by the field of view of the cameras. The turn region is at a position ahead of the DelFly, such that the turn circle including safety margins fits within the field of view. In the figure, the avoidance trajectory of the DelFly is indicated. It can be seen that the vehicle first keeps flying in a straight line, and starts to turn after almost 2m. Also note that there is an offset between the middle line of the field of view and the avoidance trajectory. For indoor flight it is important to keep the size of the safety region as small as possible since it defines the smallest area where the DelFly can operate. To obtain a minimal size for the safety region, only turns to the right are chosen as an option. By introducing the offset between flight direction and camera field of view as shown in the plot, the area is further minimized. This offset corresponds to a pointing offset of the camera on the body of the DelFly. This means that the camera points a bit to the right.

The safety region is translated into a reference disparity map. Disparity maps obtained during flight are compared to this reference map. Pixels that have a disparity value higher than the reference value are indicators of obstacles within the safety region.

A logic has been developed to implement this avoidance strategy in the system. This logic is shown in the right image of Figure 7. The logic is based on a definition of phases. Within each phase the DelFly is instructed to either fly straight or to turn. Turns are performed by using the tail rudder. The rudder input for a turn is a fixed value that is tuned to obtain the turn radius of 50cm.

Phase 1 is the phase where the DelFly is instructed to fly in a straight line. It is checked if there are any obstacles detected within the safety region. This is done by counting the amount of pixels with a disparity value that is higher

than that of the disparity map. If this exceeds a threshold of 250 pixels, this measurement is counted. If this happens more than ten times within a second, it is decided to go to Phase 2. This is a way of filtering bad or noisy measurements.

In Phase 2 the DelFly is flying the straight trajectory as indicated in the safety region in Figure 7. The system waits until a certain time has passed to go to Phase 3. This time is based on the distance to the turn point and the assumed forward speed of the DelFly.

In Phase 3 the DelFly starts to turn by giving the predefined rudder input. At the same time it is checked again if the safety region contains obstacles or not. As soon as the safety region is found to be clear (less than 200 pixels have a higher disparity than the reference) the system switches to Phase 4 immediately. Since the DelFly is turning relatively fast, it is important to react quickly to measurements that indicate a safe direction of flight. Small time delays will result in significant overshoots in heading.

In Phase 4 the DelFly is instructed to fly straight again. However, since an error could have been made in Phase 3, it is checked again if the safety region is still clear. If this is not the case, the DelFly will go back to Phase 3 and continue its turn. If this does not occur within one second, the system returns to Phase 1.

Furthermore, in Phase 1 it is always checked if there are obstacles at a really short distance of less than 1m. If this is the case, it is important to react to that situation immediately, even if it is not known if there are obstacles on the side. The system will go to Phase 3 if more than 500 pixels indicate an object at this short range.

Experiment The obstacle avoidance routine was tested in a room of 8.2×7.3 meter. This room was not adapted for the experiment. It contains both white walls with poor texture as well as walls with cabinets on the sides. During the best flight trial, the DelFly flew autonomously for 72 seconds. No operator interventions were made during this period. The operator decided to end the flight after the 72 seconds by catching the DelFly.

During the experiment the flight trajectory of the DelFly was tracked by two external cameras. The result is shown in Figure 8. The colors in the trajectory indicate the phases from the control logic. The numbers indicate the flight time. Several remarks can be made on these results, and for this reason the full trajectory is discussed.

The tests starts at time=0, where the DelFly has just been turned away from the corner by the operator. From this point on, the operator did not control the DelFly anymore. During the next ten seconds the DelFly is in Phase 1 and is supposed to fly straight. However, it can be observed that the flight trajectory is far from a perfect straight line. This is mainly due to a ventilation system in the ceiling. After the first 10 seconds Phase 2 and 3 are initiated consecutively. Note that this part corresponds to the avoidance trajectory from Figure 7. After this turn the DelFly flies to another wall (around time=20) which it avoids initially, but it needs to avoid it again a bit later because the

DelFly turns back to the wall again. This is the effect of the rudder control, which makes the DelFly unstable. Using the rudder for heading control is not ideal, since a rudder input initiates a yawing motion that disturbs the roll angle (heading angle).

Three successful avoidance turns are shown between time=20 and time=50. At time=50, a situation is shown where the control logic decides to go from Phase 3 to 4 while the DelFly is still flying in the direction of a wall. After a short moment the controller switches back to Phase 3 again and the turn is continued. This is caused by the poor texture of the wall, since the wall is completely white. This results in very uncertain measurements that often result in measurements with high disparity values, but sometimes also very low values.

It can also be noted that in many cases the DelFly seems to keep turning for a while even though it has already switched to Phase 4. In the first place this is caused by a delay due to RC communication between ground station and the DelFly. Furthermore the instability effects due to rudder inputs have the effect that the DelFly takes a bit of time to recover from a turn. While the rudder is back to the trim position, the DelFly still turns a bit more before it flies straight.

The test demonstrates that the DelFly is able to detect obstacles ahead of the vehicle using is stereo vision system and that this allows for successful avoidance. The recorded flight time of 72 seconds is a major improvement to the record of 30 seconds presented by De Croon et al.[7] using a DelFly in combination with optic flow.

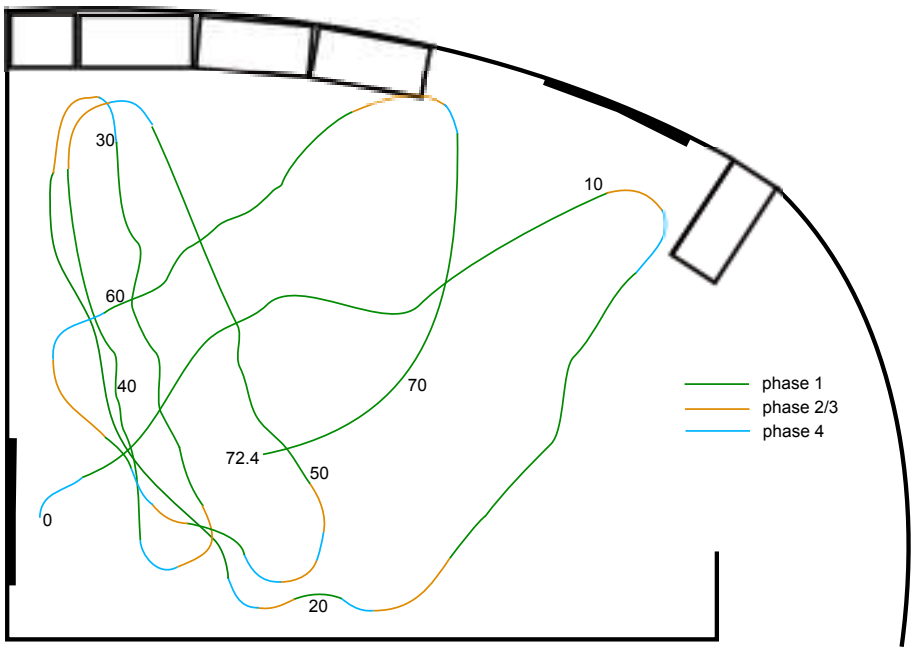


Figure 8: Flight track of the DelFly during the experiment. The numbers indicate the flight time, the colors represent the flight phases.

6 Conclusions

The study described in this paper has demonstrated that robust obstacle avoidance on a Flapping Wing MAV can be successfully realized by using stereo vision cameras onboard. It was shown that the distance estimates obtained by stereo vision can be used as a cue to implement a robust avoidance strategy. This enables the FWMAV to avoid situations where it would get locked up. It was also shown that the severe body motions due to flapping do not have a large influence on the distance measurements. This shows that stereo vision can outperform optic flow techniques.

In this study stereo vision was implemented by using a ground station to perform the stereo vision processing. It is still a large step to go from off-board to onboard processing. Nevertheless, this study proves the potential of using stereo vision for the purpose of obstacle avoidance on a FWMAV. Future studies will focus on onboard implementations of the whole system including processing and control loops.

References

[1] Griffiths, S., Curtis, J., Barber, B., McLain, T., and Beard, R., "Obstacle and terrain avoidance for miniature aerial vehicles," *Advances in Unmanned Aerial Vehicles*, 2007, pp. 213–244.

[2] Duhamel, P., Pérez-Arancibia, N., Barrows, G., and Wood, R., "Altitude feedback control of a flapping-wing microrobot using an on-board biologically inspired optical flow sensor." 2012, pp. 4228–4235.

[3] Baek, S., Bermudez, F. G., and Fearing, R., "Flight control for target seeking by 13 gram ornithopter." *IEEE/RSJ Int Conf on Intelligent Robots and Systems.*, 2011.

[4] Bermudez, F. G. and Fearing, R., "Optical flow on a flapping wing robot," *IROS 2009*, 2009, pp. 5027–5032.

[5] de Croon, G., de Clerq, K., Ruijsink, R., Remes, B., and de Wagter, C., "Design, aerodynamics, and vision-based control of the DelFly," *International Journal on Micro Air Vehicles*, Vol. 1, No. 2, 2009, pp. 71 – 97.

[6] de Croon, G., de Weerdt, E., de Wagter, C., Remes, B., and Ruijsink, R., "The appearance variation cue for obstacle avoidance," *IEEE Transactions on Robotics*, Vol. 28, No. 2, 2012, pp. 529–534.

[7] de Croon, G., Groen, M., Wagter, C. D., Remes, B., Ruijsink, R., and van Oudheusden, B., "Design, Aerodynamics, and Autonomy of the DelFly," *Bioinspiration and Biomimetics*, Vol. 7, No. 2, 2012.

[8] Scharstein, D. and Szeliski, R., "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, Vol. 47, No. 1/2/3, 2002, pp. 7–42.

[9] Hirschmuller, H., "Accurate and efficient stereo processing by semi-global matching and mutual information," *Computer Vision and Pattern Recognitio (CVPR)*, Vol. 2, 2005, pp. 807–814.

[10] Banz, C., Hesselbarth, S., Flatt, H., Blume, H., and Pirsch, P., "Real-time stereo vision system using semi-global matching disparity estimation: architecture and FPGA-implementation," *IEEE Conf. on Embedded Computer Systems (SAMOS)*, 2010, pp. 93–101.

[11] Gehrig, S. and Rabe, C., "Real-time semi-global matching on the CPU," *IEEE Conf. on Computer Vision and Pattern Recognition Workshop(CVPR)*, 2010, pp. 85–92.

[12] Chen, B. and Chen, H., "A realization of semi-global matching stereo algorithm on GPU for real-time application," *International Symposium on Multispectral Image Processing and Pattern Recognition (MIPPR2011)*, 2011, pp. 80040W–80040W.

[13] Birchfield, S. and Tomasi, C., "Depth discontinuities by pixel-to-pixel stereo," *International Journal of Computer Vision*, Vol. 35(3), 1999, pp. 269–293.