International Micro Air Vehicle Conference and Flight Competition (IMAV2013)
17-20 September 2013, Toulouse, France

Optic-flow based slope estimation for autonomous landing

G.C.H.E. de Croon, H.W. Ho, C. De Wagter,
E. van Kampen, B. Remes and Q.P. Chu

Micro Air Vehicle laboratory, Control and Operations department, Faculty of Aerospace Engineering, Delft University of Technology.
`g.c.h.e.decroon@tudelft.nl`

Abstract

In order to ease the lives of authors, editors, and trees, we present an easy-to-read guide to the easy-to-use easychair LaTeX2e document style class for EasyChair-based electronic and on-paper publishing of workshop and conference proceedings.

Abstract

Micro Air Vehicles need to have a robust landing capability, especially when they operate outside line-of-sight. Autonomous landing requires the identification of a relatively flat landing surface that does not have too large an inclination. In this article, a vision algorithm is introduced that fits a second-order approximation to the optic flow field underlying the optic flow vectors in images from a bottom camera. The flow field provides information on the ventral flow (v_x/h), the time-to-contact ($h/-v_z$), the flatness of the landing surface, and the surface slope. The algorithm is computationally efficient and since it regards the flow field as a whole, it is suitable for use during relatively fast maneuvers. The algorithm is subsequently tested on artificial image sequences, hand-held videos, and on the images made by a Parrot AR drone. In a preliminary robotic experiment, the AR drone uses the vision algorithm to determine when to land in a scenario where it flies off a stairs onto the flat floor.

Contents

1 Introduction

Autonomous landing is an important capability for Micro Air Vehicles (MAVs), especially if they have to operate outside line-of-sight. Active sensors such as a laser scanner [2] or multiple cameras as in stereo vision [1] would instantaneously provide distances to many points on the landing surface.

However, such sensor setups only work at lower heights and are not energy or weight efficient. Since such efficiency is important for MAVs, it would be beneficial to have a robust landing strategy based on a single downward-pointing camera.

With a monocular camera setup, two main approaches to autonomous landing can be employed. The first approach is visual Simultaneous Localization And Mapping (SLAM) [6, 5], in which the 3D locations of all features in sight are determined. Various approaches to visual SLAM have been proposed over the years, which have consistently improved with respect to accuracy and computational effort [7, 19]. Still, SLAM delivers a lot of detailed information on the environment that is not strictly required for the landing task, and hence uses more computational resources than necessary.

The second approach is bio-inspired in the sense that it directly uses the optic flow field for control. The earliest studies of this approach [12, 16] are based on the biological finding that bees keep the ventral flow constant during a grazing landing [3, 4]. The ventral flow is equal to the translational velocity divided by the height (v_x/h), and keeping it constant results in a soft touch down. Since the ventral flow cannot account for the vertical dynamics, recent studies [13, 14, 17] complement the ventral flow with the time-to-contact, i.e., the height divided by vertical velocity ($h/-v_z$). The advantage of a bio-inspired approach is that light-weight neuromorphic sensors with high sensitivity and update frequency can be used directly for controlling the landing [9, 17].

The above-mentioned bio-inspired landing studies focus on autonomous landing on a flat surface. However, many real-world missions for MAVs will involve unknown landing sites that may be cluttered or have a slope. Indeed, bees can choose their landing site and adapt their body attitude to the slope of the surface on which they are landing [8]. For an MAV the detection of the landing surface's slope would also be of interest, either for adapting the MAV's attitude or for evaluating the suitability of the surface for landing.

In this article, a computationally efficient computer vision algorithm is proposed that uses a bottom camera. First, optic flow vectors are determined and then a second order fit of the entire optic flow field is made. The fit provides information on the ventral flow, time-to-contact, flatness of the landing surface, and surface slope. The algorithm is computationally efficient and robust to noisy optic flow vectors that are likely to occur in a real MAV landing scenario. In addition, it lends itself well for translation to algorithms for novel sensors that mimick insect eyes [18, 11].

The remainder of the article is structured as follows. In Section 2, the vision algorithm is explained. It is tested on image sequences in Section 3. The results of a landing experiment with a Parrot AR drone are discussed in Section 4. Finally, conclusions are drawn in Section 5.

2 Vision algorithm for slope estimation

The vision algorithm proposed for the slope estimation is based on the early optic flow work in [15]. In the explanation of the algorithm, a pinhole camera model is employed. The algorithm assumes (1) the camera to point downward, (2) the rotation rates to be known from the MAV's state estimation (e.g., using gyros), and (3) the landing surface in sight to be predominantly planar. Under these assumptions, the optic flow vectors in the image follow the equations:

$$u = (-U + xW)/Z, \tag{1}$$

$$v = (-V + yW)/Z, \tag{2}$$

with u and v the (derotated) optic flow in the x- and y-direction of the image, and U, V, W the motion in X, Y, and Z direction, respectively. x and y are image coordinates. The surface height Z can be modelled as a plane:

$$Z = R + \alpha X + \beta Y, \tag{3}$$

with R the height above the surface at $(x, y) = (0, 0)$. By a transformation of coordinates, in [15], the surface height is rewritten as:

$$z = (Z - R)/Z = \alpha x + \beta y, \tag{4}$$

and the velocities are scaled with respect to the height R, i.e., $u_0 = U/R$, $v_0 = V/R$, and $w_0 = W/R$. This leads to:

$$u = (-u_0 + x w_0)(1 - z) \tag{5}$$

$$v = (-v_0 + y w_0)(1 - z) \tag{6}$$

Replacing z with $\alpha x + \beta y$ (from Eq. 4), leads to:

$$u = -u_0 + (u_0 \alpha + w_0)x + u_0 \beta y - \alpha w_0 x^2 - \beta w_0 xy \tag{7}$$

$$v = -v_0 + v_0 \alpha x + (v_0 \beta + w_0)y - \beta w_0 y^2 - \alpha w_0 xy \tag{8}$$

In [15], the rotational optic flow terms were left in, and the discussion hence goes toward how to find the Focus-of-Expansion (FoE) and determine the first and second order spatial flow derivatives at that point. In that way, all terms can be identified (translational, rotational, and the slopes of the surface). However, determining the FoE is a difficult task in itself, and errors in its location can have a large influence on the subsequent results. In addition, since MAVs typically have access to gyro measurements, the rotational components do not have to be determined. So instead of finding the FoE, the vision algorithm proposed in this article immediately determines the parameters of the optic flow field:

$$u = \mathbf{pu}[1, x, y, x^2, xy]^T, \tag{9}$$

$$v = \mathbf{pv}[1, x, y, x^2, xy]^T, \tag{10}$$

The parameter vectors **pu** and **pv** are estimated separately with a maximal likelihood linear least squares estimate within a robust RANSAC estimation procedure [10], with 5 points per fit and 20 iterations. Figure 1 illustrates the result of such a fit.

The so-determined parameters can then be used to estimate the following variables of interest for an autonomous landing. The **ventral flow** is set to $(u_0, v_0) = (pu_1, pv_1)$. The **time-to-contact** and **slopes** can be retrieved from the first order spatial derivatives at the center of the image $(x, y) = (0, 0)$:

$$u_x = u_0 \alpha + w_0 = pu_2 + 2pu_4 x + pu_5 y = pu_2, \tag{11}$$

$$u_y = u_0 \beta = pu_3 + pu_5 x = pu_3, \tag{12}$$

$$v_x = v_0 \alpha = pv_2 + pv_5 y = pv_2, \tag{13}$$

$$v_y = v_0 \beta + w_0 = pv_3 + 2pv_4 y + pv_3 x = pv_3, \tag{14}$$

The slopes can then be retrieved as $\alpha = pv_2/v_0$ and $\beta = pu_3/u_0$. However, these equations become ill-conditioned and hence sensitive to even the smallest of noise if u_0 or v_0 are small. If there is insufficient motion in X and Y direction, then it may still be possible to estimate the slopes on the basis of the second order derivatives:

$$u_{xx} = -2\alpha w_0 = 2pu_4, \tag{15}$$

$$u_{xy} = -\beta w_0 = pu_5, \tag{16}$$

$$v_{xy} = -w_0 \alpha = pv_5, \tag{17}$$

$$v_{yy} = -2\beta w_0 = 2pv_4, \tag{18}$$

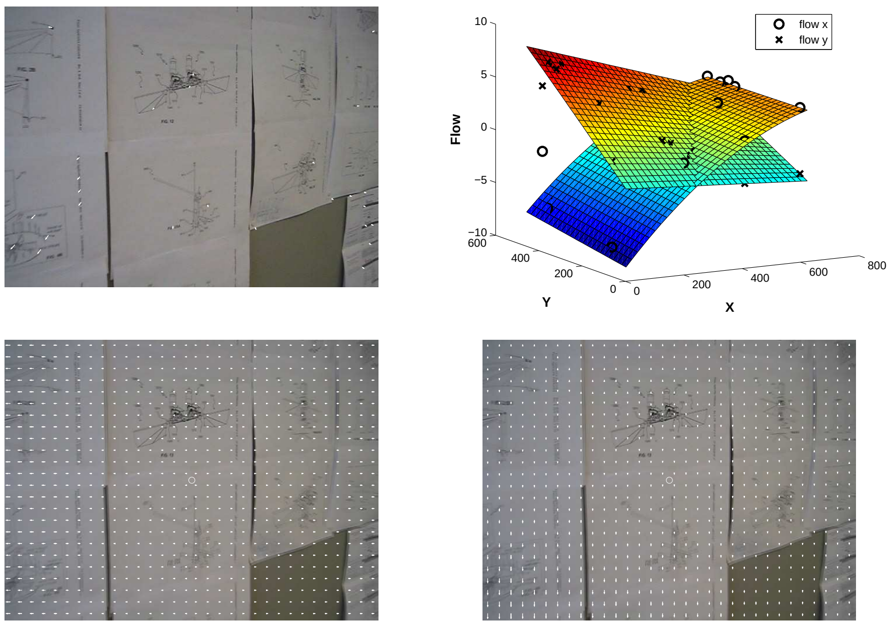


Figure 1: Illustration of the vision process. Top left: the determined optic flow vectors. Top right: quadratic fits for the flow in x-direction (circle markers) and y-direction (cross-markers). Bottom left: estimated flow field in x-direction. Bottom right: estimated flow field in y-direction. The motion of the camera is straight towards the wall, which has an angle of $\sim 45°$ to the camera axis.

which leads to two formulas for α and two formulas for β. However, all of these formulas depend on w_0, the relative vertical velocity. When looking at the formulas for the first order spatial derivatives, one will notice that w_0 cannot be determined without knowing α or β - seemingly introducing a chicken-and-egg problem. The solution lies in the realization that we only need these second order derivatives if the ventral flow estimate(s) are small. If, for example, $u_0 \approx 0$ then:

$$u_x = u_0\alpha + w_0 \approx w_0, \tag{19}$$

and, similarly, for $v_0 \approx 0$:

$$v_y = v_0\beta + w_0 \approx w_0, \tag{20}$$

In summary, when the horizontal flow is not sufficient, the relative velocity w_0 can be determined. In turn, this leads to slope estimates $\alpha = -pu_4/w_0$, $\alpha = -pv_5/w_0$, $\beta = -pu_5/w_0$, and $\beta = -pv_4/w_0$. All these equations become ill-conditioned if w_0 becomes small. This is intuitive, since if there is also little motion in the Z-direction, then no estimates of slope are possible. The remaining question is then how to deal with cases in which an MAV has velocities in multiple directions. For optimal estimation, one should fuse the different α and β estimates on the basis of their certainties and possible prior probability distributions. However, in this preliminary work slopes are determined on the basis of the first-order optic flow derivatives if there is sufficient motion in the X, Y-plane and only on the basis of the second-order optic flow derivatives if there is not.

The time-to-contact τ is determined on the basis of the divergence $D = u_x + v_y$, with $\tau = 2/D$. Hence, τ includes the ventral flow and the slopes (see Eqq. 11 and 14). This makes sense, because in the case of pure horizontal motion, the MAV can still intersect the landing surface if it has a slope. However, if one is only interested in the component in the Z-direction, $\tau = 1/w_0$ could be used. In the current work the first definition is employed.

Finally, the **flatness** of the slope is related to how good the optic flow vectors fit with the above-described quadratic model. The RANSAC procedure returns a number of inliers and an error, which can both serve as measures for flatness.

3 Experiments on image sequences

In this section, a preliminary test of the vision algorithm is performed by applying it to various image sequences. The MATLAB code of the vision algorithm is available at `http://www.bene-guido.eu/`, so that the reader can test it on his / her own image sequences[1]. The algorithm is tested on three types of image sequences: (1) image zooms, (2) manually made videos, and (3) images from the Parrot AR drone. Figure 2 shows five example images.

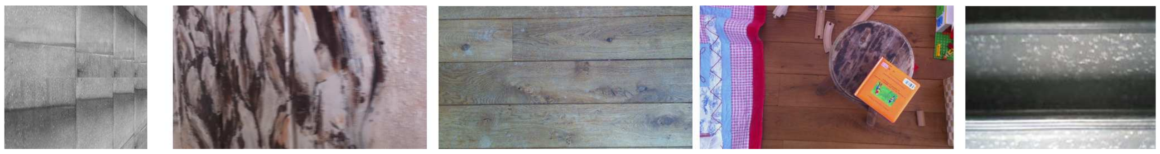


Figure 2: Example images. From left to right: artificial image, images from manual videos (wall, flat scene, structured scene), image from the drone's bottom camera.

The test on image zooms serves to test the estimation of time-to-contact and ventral flow, and is motivated by the fact that in this setup the ground-truth values are known. Five image zooms were performed on a roof texture. The ground-truth time-to-contact decreases at a constant rate from 200 to 5. The results of the time-to-contact estimation are shown in Figure 3. The estimated time-to-contact is very accurate, even up to 200 frames from contact.

In order to test the slope estimation, three sets of videos have been made of a textured wall. In the first set the camera moves in the Y-direction (first up and then down), in the second set in the X-direction (first left and then right), and in the third set the camera moves in the Z-direction (in this case toward the wall). Put in a landing context, the first two sets cover horizontal motion, while the third set covers vertical motion toward the landing surface. Each set contains 5 image sequences in which the angle of the wall roughly iterates over the angles $\{0, 15, 30, 45, 60\}$ degrees. Please remark that the camera is moved in-hand, so additional motions and even rotations are present in the images. Since there is no clear ground-truth, this experiment mainly serves the goal of verifying that the slope α increases with an increasing angle to the wall. Figure 2 shows an example image from the sequence at ~ 45 degrees.

Figure 4 shows the results of this experiment for motion in the Y-direction (left plot) and X-direction (right plot). The plots show the estimated slope α (y-axis) over time (x-axis). The brightness of the lines are determined by the angle with the wall, with the darkest color corresponding to 0 degrees and the brightest color corresponding to 60 degrees. The lines are somewhat noisy, especially when the

[1]Please note that the code will be made available only after publication of the article.

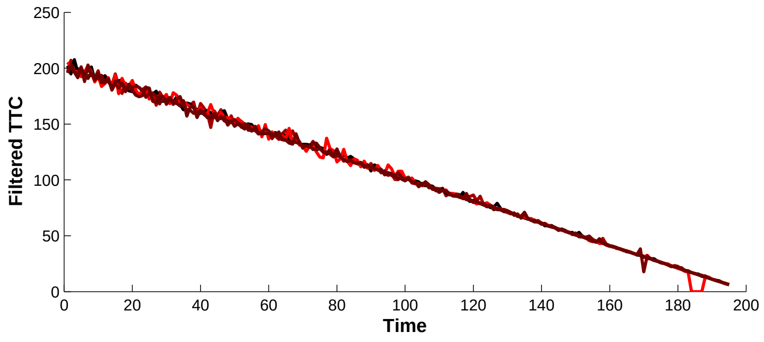


Figure 3: Results of time-to-contact estimation on five image zooms. The ground-truth time-to-contact goes from 200 to 5.

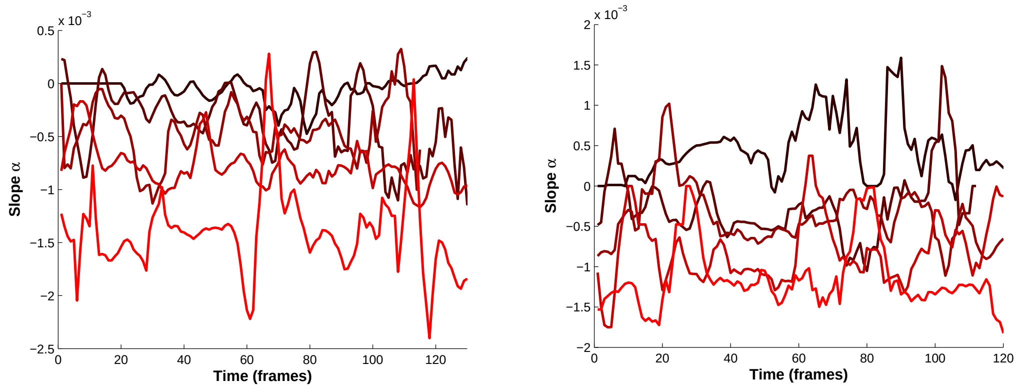


Figure 4: Results of slope estimation on video sequences. Left: motion in Y-direction. Right: motion in X-direction. The plots show the estimated slope α (y-axis) over time (x-axis). The brightness of the lines are determined by the angle with the wall, with the darkest color corresponding to 0 degrees and the brightest color corresponding to 60 degrees.

motion gets small, e.g., at reversal points of the movements. However, in both plots, a clear ordering can be seen from dark to bright slopes, as desired.

Figure 5 contains the results for motion in the Z-direction, illustrated in the same manner. In this case, the results are slightly less clear. The lower degree slopes switch between positive and negative values, while the higher degree slopes are constantly negative. The slightly less good results may be explained by the fact that the second-order spatial derivatives of the optic flow field are more difficult to determine than the first order ones.

Finally, the detection of flatness is tested with the help of two sets of each three videos. The first set contains flat surfaces, while the second set contains considerable 3D structure. The videos are made by moving toward the ground surface, i.e., in the Z-direction. The results are shown in Figure 6, which shows the mean absolute errors of the quadratic fits. The black lines illustrate the results for the flat surfaces, while the red lines illustrate the results for the scenes with 3D structure. The videos of flat surfaces have clearly a lower error than the videos of structured scenes. However,

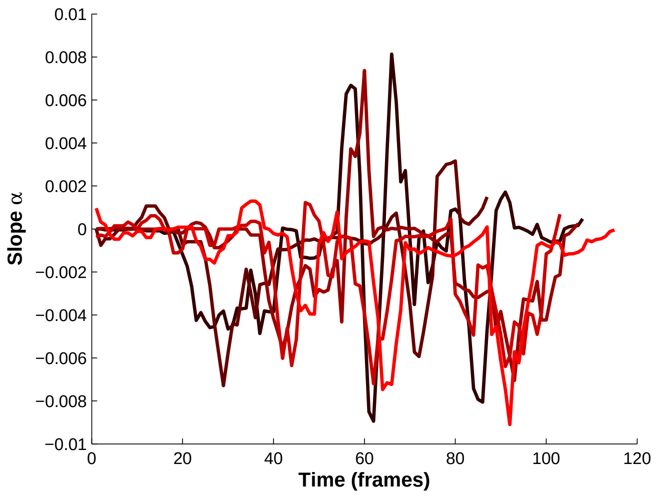


Figure 5: Results of slope estimation on video sequences with the camera moving in Z-direction. The plots show the estimated slope α (y-axis) over time (x-axis). The brightness of the lines are determined by the angle with the wall, with the darkest color corresponding to 0 degrees and the brightest color corresponding to 60 degrees.

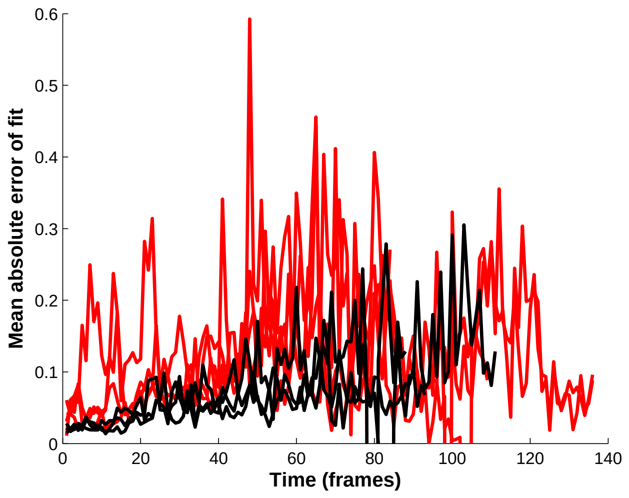


Figure 6: Results for estimation of landing surface flatness. Mean absolute error of the quadratic fits (y-axis) over time (x-axis). The black lines illustrate the results for the flat surfaces, while the red lines illustrate the results for the scenes with 3D structure.

placing a threshold is not straightforward, as the error also seems to depend on the time-to-contact. In particular, lower time-to-contacts entail larger optic flow, which leads to larger errors.

Finally, the slope estimation algorithm is run on the 160×120 images of the Parrot AR drone's

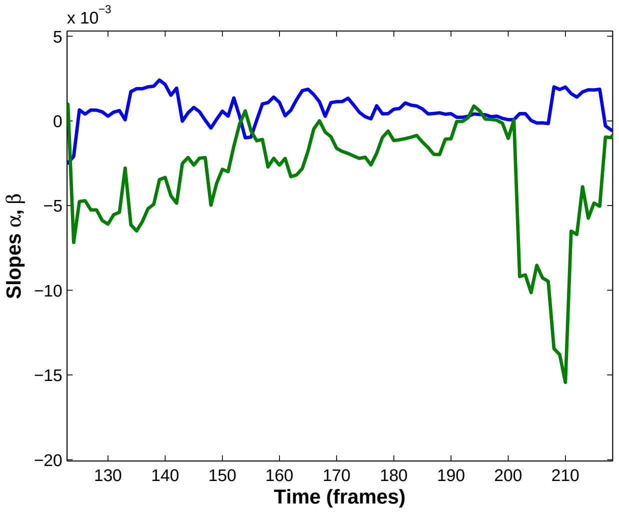


Figure 7: Results for estimation of slope with the Parrot AR drone's bottom camera. The plot shows α (blue) and β (green) over time (x-axis).

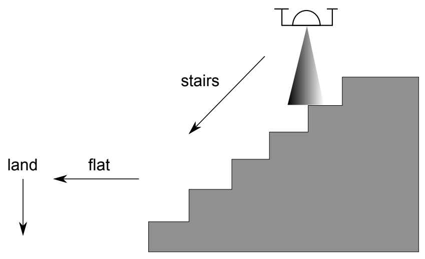
bottom camera, while it moves down a stairs. Figure 7 shows the estimated slope in α in the x-direction (blue) and β in y-direction (green). The y-direction is in the direction of the stairs. Indeed, while α is reasonably small and switches from positive to negative and back, β is larger and mostly negative.

In summary, the vision algorithm provides slope estimates that vary clearly with the angle of the wall and provides a cue for the detection of 3D structure below the MAV. The information is easier to obtain when the MAV makes movements in the X, Y-plane, as this then depends only on the first order spatial derivatives of the optic flow.

4 Landing experiment

The above-described landing algorithm is implemented in TU Delft's *SmartUAV* ground control station, and tested on a Parrot AR drone. In order to show that the landing algorithm can be used in a generic way, a staircase was chosen as the test environment. Stairs are challenging, since it is not a flat inclined surface. The setup for the experiment is presented in Figure 8. The experiment started with a forward motion of the drone (in the Y-direction) generated by a small pitch angle. This forward motion was controlled by proportional controller using the feedback of the velocity estimated from the AR drone optic flow measurement. The purpose of moving the drone in forward direction was to generate ventral flow to estimate the slope in real-time. The drone thus travels down the stairway and to a flat ground in the end. The height of the drone was under control of the Parrot firmware, implying that it remained at a fixed height above the steps of the stairs.

A simple landing strategy was used in this preliminary experiment, i.e. when the value of the estimated slope was close to zero $\beta \in [-0.0005, 0.0005]$, a landing command was given to land the drone immediately. The results of the estimated slope β from the onboard images captured during the flight is shown in Figure 9. It is clearly seen that the value of the slope was always negative when



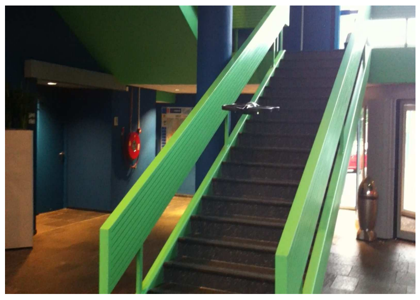


Figure 8: Landing Experiment Setup

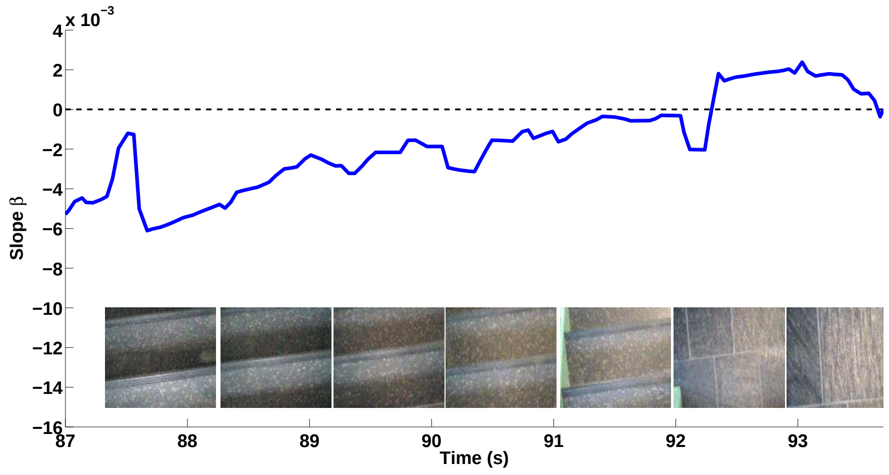


Figure 9: Real-time Slope Estimation Results with the Parrot AR drone's bottom camera. The plot shows the estimated slope β and images taken from the onboard camera over time (x-axis)

the drone was flying above the stairs. At the instant when it reached the flat surface, the value of the slope changed its sign and the autoland was activated to bring the MAV to the ground. Please note that the drone moves forward by pitching forward. The experimental results show that this considerably influences the estimated slope, which was not accounted for in the code. This led to slightly less negative slope values above the stairs and slightly more positive slope values above the flat floor[2].

5 Conclusions

The main conclusion is that the proposed vision algorithm is able to extract ventral flow, time-to-contact, flatness, and slope of the landing surface beneath an MAV. A preliminary landing experiment shows that the algorithm is computationally efficient enough to run in real-time and can discriminate between the stairs and a flat surface. The experiments show that it is important to take into account the pitch angle of the MAV while determining the slope of the surface.

[2]A video of the experiment can be found here: `http://www.youtube.com/watch?v=TXIPb1NvUJY&feature=youtu.be`

Future work will focus on a further investigation of how optic flow can be used to best estimate the variables of interest. For example, combining slope estimates in a Bayesian fashion could considerably improve the estimates and allow for the exploitation of translation in multiple directions. Moreover, a more elaborate landing procedure utilizing the estimated variables will be devised and tested in a more complex environment.

References

[1] M.W. Achtelik, A.G. Bachrach, R. He, S. Prentice, and N. Roy. Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments. In *Unmanned Systems Technology XI, SPIE*, 2009.

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

[3] E. Baird, M.V. Srinivasan, S. Zhang, and A. Cowling. Visual control of flight speed in honeybees. *Journal of Experimental Biology*, 208:3895–3905, 2005.

[4] E. Baird, M.V. Srinivasan, S. Zhang, R. Lamont, and A. Cowling. Visual control of flight speed and height in honeybee. In *Simulation of Adaptive Behavior*, pages 40–51, 2006.

[5] 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.

[6] K. Celik, S.J. Chung, and A.K. Somani. Mvcslam: Mono-vision corner slam for autonomous micro-helicopters in gps denied environments. In *Proceedings of GNC'08*, 2008.

[7] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[8] C. Evangelista, P. Kraft, M. Dacke, J. Reinhard, and M.V. Srinivasan. The moment before touchdown: Landing manoeuvres of the honeybee apis mellifera. *Journal of Experimental Biology*, 213:262–270, 2010.

[9] F. Expert, S. Viollet, and F. Ruffier. Outdoor field performances of insect-based visual motion sensors. *Journal of Field Robotics*, 28(4):529–541, 2011.

[10] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[11] D. Floreano, R. Pericet-Camara, S. Viollet, F. Ruffier, A. Brueckner, R. Leitel, W. Buss, M. Menouni, F. Expert, R. Juston, M.K. Dobrzynski, G. LEplattenier, H.A. Mallot, and N. Franceschini. Miniature curved artificial compound eyes. *Proceedings of the National Academy of Sciences*, 2013.

[12] N. Franceschini, S. Viollet, F. Ruffier, and J. Serres. Neuromimetic robots inspired by insect vision. *Advances in Science and Technology*, 58:127–136, 2008.

[13] B. Hérissé, T. Hamel, R. Mahony, and F.X. Russotto. The landing problem of a vtol unmanned aerial vehicle on a moving platform using optical flow. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1600–1605. IEEE, 2010.

[14] D. Izzo and G.C.H.E. de Croon. Landing with time-to-contact and ventral optic flow estimates. *Journal of Guidance, Control, and Dynamics*, 35(4):1362–1367, 2012.

[15] H.C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of Royal Society, London B*, 208:385–397, 1980.

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

[17] G. Sabiron, P. Chavent, T. Raharijaona, P. Fabiani, and F. Ruffier.

[18] Y.M. Song, Y. Xie, V. Malyarchuk, J. Xiao, I. Jung, K.-J. Choi, Z. Liu, H. Park, C. Lu, R.-H. Kim, R. Li, K.B. Crozier, Y. Huang, and J.A. Rogers. Digital cameras with designs inspired by the arthropod eye. *Nature*, 497:95–99.

[19] S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular-slambased navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.