Efficient Terrain Following for a Micro Aerial Vehicle with Ultra-Wide Stereo Cameras

Marcus G Müller
German Aerospace Center (DLR)
Robotics and Mechatronics Center
Münchener Str. 20
82234 Weßling
Marcus.Mueller@dlr.de

Samantha Stoneman
German Aerospace Center (DLR)
Robotics and Mechatronics Center
Münchener Str. 20
82234 Weßling
Samantha.Stoneman@dlr.de

Ingo von Bargen
German Aerospace Center (DLR)
Robotics and Mechatronics Center
Münchener Str. 20
82234 Weßling
Ingo.Bargen@dlr.de

Florian Steidle
German Aerospace Center (DLR)
Robotics and Mechatronics Center
Münchener Str. 20
82234 Weling
Florian.Steidle@dlr.de

Wolfgang Stürzl
German Aerospace Center (DLR)
Robotics and Mechatronics Center
Münchener Str. 20
82234 Weßling
Wolfgang.Stuerzl@dlr.de

Abstract—**In recent years, Micro Aerial Vehicles (MAVs) have drawn attention to the aerospace community. With such autonomous flying platforms, it is possible to explore foreign extraterrestrial bodies in an efficient and faster manner than other robotic platforms, like rovers. In addition, they can be equipped with a variety of different sensors. Cameras are especially well suited, since they are light, energy-efficient and deliver a broad spectrum of information. Following the underlying terrain in a defined height is a fundamental task for any exploring MAV. To accomplish this, many systems possess a designated height sensor, which in most cases only delivers a single height estimation taken from nadir. In such a setup, the MAV is just adjusting its height based on the current height estimation and does not take any terrain lying ahead into account, which results in delayed height adjustments. In this paper, we propose a novel method based on a wide-angle stereo camera setup, which is attached to the MAV, to overcome such problems. Due to the wide vertical field of view, the vehicle is able to not only measure its current height, but also the terrain lying ahead. Therefore, the MAV is able to perform a better terrain following compared to other methods, which use just a single nadir height sample. Our algorithm only needs to take the depth image, calculated by the stereo cameras, and the estimated gravity vector into account. Therefore, our method is very fast and computationally efficient, compared to other methods, which build up an entire map beforehand. As a result, the procedure presented here is also suitable for tiny flying systems with low computational capabilities and memory resources. The terrain following algorithm runs in real-time and on board the system, and is therefore also suitable for confined environments, like caves, and where communication delays are present. We evaluate our method with simulated data and real tests on an MAV. To demonstrate that our method works in a variety of different terrains, we show experiments with different slopes and obstacles in the flight path. We also compare our method to a basic terrain following by using just a single height measurement in a more classical approach.**

Table of Contents

1. Introduction 1
2. Related Work 2
3. System Setup 2
4. Terrain Following Methods 3
5. Simulation and Experiments 5
6. Conclusion 7
Acknowledgments 7
References 8

978-1-7281-2734-7/20/$31.00 ©2020 IEEE

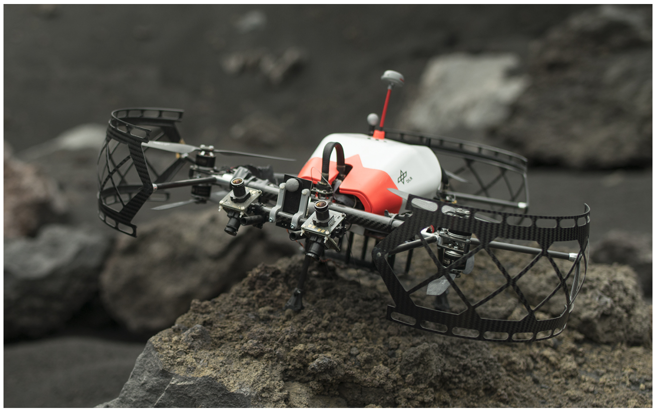

Figure 1. ARDEA on Mt. Etna

1. Introduction

Micro Aerial Vehicles (MAVs) have drawn attention in the aerospace community as they have shown potential for space exploration applications. Since they are lightweight, small, and can carry a variety of sensors, it makes them attractive for different kinds of scouting tasks. As a result of their small size these flying robots are suitable for exploration of complex 3D structures such as caves and lava tubes on other planets with an appropriate atmosphere. However, they must be able to navigate fully autonomously in order to cope with the communication delays which arise during space missions beyond Earth's orbit. Autonomous navigation is also attractive for terrestial missions when channel bandwidth is low and communication quality cannot always be ensured.

A key capability for autonomous navigation is the ability to follow the terrain closely. Terrain following ensures higher map quality and therefore improved collection of valuable scientific data. In such situations, it is sufficient to use a less complex terrain following procedure and switch to a more complete motion planning method in case a complex 3D structure is reached.

When following terrain at high altitudes, it is sufficient to measure the current vehicle height and adjust it accordingly. For this sensors are required, which can measure a single distance from nadir. Such sensors include ultra sonic modules for closer distances and single beam LIDAR and RADAR sensors for larger distances. However, such ranging sensors are usually heavy and power demanding, which makes them more diffucult to use on an MAV. Even more important, this described technique has a major drawback. Since the sensor only measures the current height and adjustment of vehicle height is not instantaneous, the terrain will be followed with some delay. This effect is particularly intense when flying low to the ground and can even cause a collision with the terrain when steep elevation changes occur.

To overcome this problem, it is beneficial to look ahead instead of just in the direction of the nadir. Camera systems are well suited for such a setup. Equipped with a suitable lens, they can cover a wide angle of view. As a result, the system perceives the entire terrain lying ahead. Compared to other sensors they are also lightweight and therefore ideally suited for small aerial vehicles. The use of stereo cameras is particularly useful for directly perceiving the 3D structure of the terrain.

We have developed such a stereo vision based method for terrain following and have applied it to our flying system, ARDEA, a hexacopter which is equipped with ultra wide-angle stereo cameras. Figure 1 depicts an image of ARDEA. Our method uses only the current image data taken from the stereo cameras and the current state estimation of the visual-inertial component. The resulting point cloud is truncated to those points that are relevant for the terrain following. Although our MAV constructs a persistent map of its environment, the calculations for the terrain following procedure are limited to the current image and relevant points, which makes our approach fast compared to a full motion planning algorithm.

Different metrics can be used to measure the quality of terrain following. We have chosen to define the optimal trajectory as that which maintains a constant, predefined minimum distance to the terrain. When the robot is simplified as a single point, this is equivalent to a sphere rolling along the surface. Since the robot has a particular footprint, the sphere can be replaced by a cylinder.

The paper is structured as followed: First, related methods are presented and discussed. Then we present a description of our flying system ARDEA. In Section 4, our approach is described. In Section 5, we demonstrate our method in a simulation and real world experiments. The paper concludes with a summary.

2. Related Work

Some of the main use-cases for terrain following are civilian aircrafts which aim to fly at a constant height over ground, military strike aircrafts for flying very close to the ground to avoid detection of enemy radars, and military helicopters. The underlying technology is commonly a Terrain-following radar (TFR) system that scans the ground below and in front of the aircraft with narrow radar beams. The height changes of the upcoming terrain are computed by analyzing the received radar signals. Using this information, the autopilot system is able to maintain a reasonably constant height above ground [1].

Another method is the terrain profile matching (TERPROM). It is a military navigation Ground Proximity Warning System commonly used in aircraft and missiles. The height above ground is computed by combining a radar altimeter and a navigation system with stored digital elevation data.

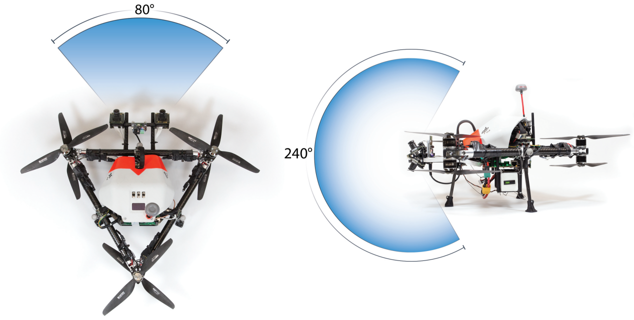

Figure 2. Top and side view of the multicopter. Bluish areas indicate the horizontal and vertical FOV of the multi-camera system.

Also in the field of robotics there have been contributions to terrain following. In [2] two optical flow sensors are used to follow the relief of a terrain. One optical flow sensor is facing vertically downwards and one 45 degrees to the front. The frontal sensor is used to compensate early for relief disturbances.

In [3] a motion-sensing visual system is used to follow the terrain and avoid obstacles. Signals from the camera, which contains 20-photoreceptors, are processed by 19 custom Elementary Motion Detection (EMD) circuits, which are derived from those of an insect. In order to compensate for future terrain changes, the eye is tilted so that it covers the forward and downward region.

3. System Setup

Our flying system ARDEA consists of a frame (propulsion system) and a navigation stack, which are mechanically decoupled. This means, that the navigation stack is independent of the propulsion system and can even be used as a handheld device. The navigation stack consists of the following components:

- Low-level real-time embedded computer: BeagleBone Black (single core 1 GHz CPU, 512 MB RAM) embedded single-board computer with a custom cape / breakout PCB. It contains a watchdog safety circuit, power supplies for 3.3 V, 5 V and 12 V and a buzzer.
- High-level embedded computer: Intel NUC5i7RYH (dual-core 3.1 GHz i7 CPU, 8 GB RAM).
- Spektrum 2.4 GHz DSMX satellite RC receiver.
- Analog Devices ADIS16367 IMU. It consists of 3 DOF accelerometers and 3 DOF gyroscopes, which are factory-calibrated and temperature compensated.
- Four wide-angle cameras for the dual stereo setup.
- Xilinx Spartan 6 LX75T FPGA running the SGM stereo algorithm [4].
- Ubiquiti Bullet M5-HP 5 GHz WLAN access-point.

While all other components are rigidly mounted on the navigation stack, the IMU is mounted on rubber dampers in addition to the mechanical decoupling between frame and navigation stack. A dedicated low-level computer runs all hard real-time critical code such as the attitude and position controller along with the 500 Hz IMU readout. The high-level computer runs all computational intensive soft, real-time critical visual navigation components, such as visual odometry, local reference filter and mission- and trajectory-planning.

The frame is arranged in a triangular shape. This allows the cameras to be as close as possible to the navigation stack while having a clear camera view without propellers. On each corner of the triangle, there are two motors, totaling to 6 motors. This gives the system redundancy in case a motor fails during flight. Each motor is equipped with a 10 inch carbon propeller. The field of view of the four wide-angle cameras and the MAV frame design are shown in Figure 2.

The stereo camera pairs are the primary sensors on-board ARDEA and their output is used for the controller as well as higher-level autonomy software, including state estimation and map generation. To achieve a large vertical FOV with reasonable resolution, we use two wide-angle camera pairs aligned such that the larger image dimension lies along ARDEA's vertical body axis. More information about our system can be found in [5].

4. Terrain Following Methods

In the following, we first present a terrain following method, which is scanning just the current height in nadir direction, since it is used as comparison in our simulation. Then we introduce our terrain following method.

Nadir Scanning Method

As mentioned in the introduction, the most simple solution for terrain following is the usage of a sensor, which can measure the current height of the vehicle. The calculated difference of the measured and the desired height is then sent to the controller for compensation. In order to follow a given terrain, this procedure is repeated continuously. The left side of Figur 3 illustrates this method. This solution is commonly used for flying vehicles since it is a computationally cheap method that relies on uncomplicated sensor setups. As sensors, ultra sonic modules, or single beam RADAR and LIDAR systems are used. The sensor is mounted on the flying system to face along the nadir direction of the vehicle. However, this nadir scanning method has some major drawbacks.

First, in order to measure the distance along the nadir direction, the sensor has to be actively moved to compensate for pitch and roll rotations. This results in a complex mechanic component. Another problem is caused by the usage of just a single height measurement at nadir, which reults into the current delta to the desired height, but not the future one. Therefore, only the height difference for a sample point in the past is corrected, which results in a shifted trajectory from the terrain. Although this shift might not play a major role if the vehicle flies in high altitudes, it can become problematic for those flying in low altitudes since a delayed response can lead to a collision with the terrain.

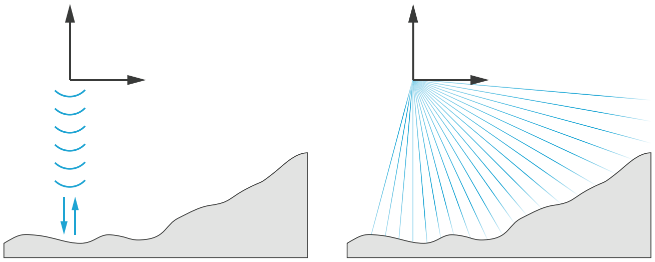

Figure 3. Left: Nadir method with single beam sensor. Right: Proposed method with wide angle stereo cameras.

Proposed Method

The greatest disadvantage of the naive method is that it is strictly reactive. Our proposed method, on the other hand, uses a wide angle stereo setup, which can foresee height changes and plan ahead to keep the desired height above ground, even for steep height gradients as illustrated on the right side of Figure 3. With the data of the cameras, one can calculate the corresponding 3D points, and the desired goal waypoint is projected into the point cloud.

Afterwards, this cloud is reduced to just the relevant points for the terrain following and rasterized to a grid. For each cell, the maximum height value is taken and the grid is projected into 2D. To follow the terrain, only the height values along the previously defined path have to be changed accordingly. In order to do this, sample waypoints along the predefined path are taken and their height is adjusted to follow the terrain. For each sampled waypoint, the corrected height value is calculated by keeping a minimum distance from it to each point of the 2D projection. To smooth the resulting list of waypoints, a spline is fitted through them. This spline is then executed by the controller. This process is repeated periodically until the waypoint goal is reached. In the following, the different components to realize the algorithm presented here are explained in more detail.

3D point cloud from images—First, 3D data points have to be measured from the current scene to get an estimate of the current terrain. For that, our approach uses the camera information of our flying system. As shown in Figure 2, ARDEA's vision sensors consist of four wideangle cameras arranged in two stereo configurations covering a large stereo field of 240° vertically and 80° horizontally. The Field of view (FOV) of each camera is about 80° × 125°. As shown in Figure 4, each camera image is remapped to two pinhole images, resulting in four pinhole images (with FOV approx. 80° × 60°) for each side, which are then sent to an FPGA for stereo processing using the SGM algorithm (see [6] for details). The computed depth/disparity maps are displayed as grey scale images, where brightness of pixel values increases with distance to objects. Black pixels indicate regions where no reliable depth values could be computed, e.g., due to occlusion or insufficient image texture.

For each pixel with image coordinates (u, v), the distance z from the camera can be calculated from stereo disparity $d(u, v)$ according to

$$z(u,v) = \frac{b\,f}{d(u,v)} \quad , \tag{1}$$

where b is the baseline between the two stereo cameras and f the focal length in pixels. The corresponding 3D point in the camera reference frame (defined by the left camera of a stereo pair) is given by

$$\vec{x}(u,v) = z(u,v)\left(\frac{u-u_c}{f}, \frac{v-v_c}{f}, 1\right)^{\top} \quad , \tag{2}$$

with (u_c, v_c) being the coordinates of the camera center. Note that for depth estimation from stereo, the error of distance increases with the square of distance:

$$|\Delta z(u,v)| = \frac{b\,f}{d(u,v)^2}|\Delta d| = \frac{z(u,v)^2}{b\,f}|\Delta d| \quad . \tag{3}$$

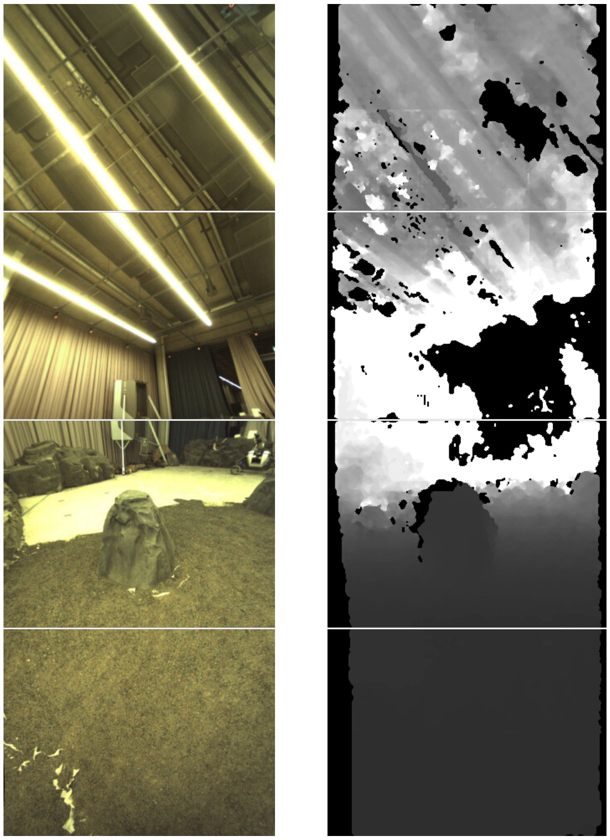

Figure 4. Left: Examples of the four rectified RGB pinhole images generated from the images of the two left cameras of ARDEA. Right: The four corresponding depth images.

Finally, the points of all pinhole stereo cameras $i = 1, 2, 3, 4$ are transformed into the body frame, the common reference system of the multicopter:

$$^{\mathrm{b}}\vec{x} = {^{\mathrm{b}}\mathbf{T}_i}\, {^{i}\vec{x}} \ . \tag{4}$$

Height correction—Once the pointcloud of the current terrain is obtained, the height correction of the predefined path can be calculated. For the height adaption of the trajectory, not all 3D points from the stereo cameras have to be taken into account, since some are too far away to affect the current path. Therefore, it is beneficial to reduce the pointcloud to the set of necessary points before conducting further computations. In order to do that, the goal waypoint is projected into the pointcloud. Then, a vector pointing from the current pose to the goal is calculated and the pointcloud in the body frame is aligned with that vector to get a pointcloud in the goal frame. Points that exceed certain boundary limits are truncated from the pointcloud. As a result, a rectangle swath remains, which is rasterized along the vector direction towards the goal waypoint to obtain a grid. These steps are illustrated in Figure 5. The size of the grid depends on the used camera system, the flight altitude, the robot velocity and the update rate of the terrain follower. In our case we found values between 2 and 3 m for the step size l_g to be a good choice. The width w_g corresponds to the long axis of the cylinder.

For each cell c_j, the maximum height $h(j)$ is taken, which corresponds to a projection into 2D space. The resulting function is a simplified version of the underlying terrain, where its resolution is determined by the cell size. We found that slices of size $\Delta l_g = 0.2$ m suit the most terrains. The function is sampled equidistantly from start to end. For each sample, the corresponding corrected height value h_c is calculated, which ensures that a circle with a predefined radius r is sliding along the function, Therefore, for each sample point the to be corrected height value resulting from each terrain point of function $h(j)$ is calculated by

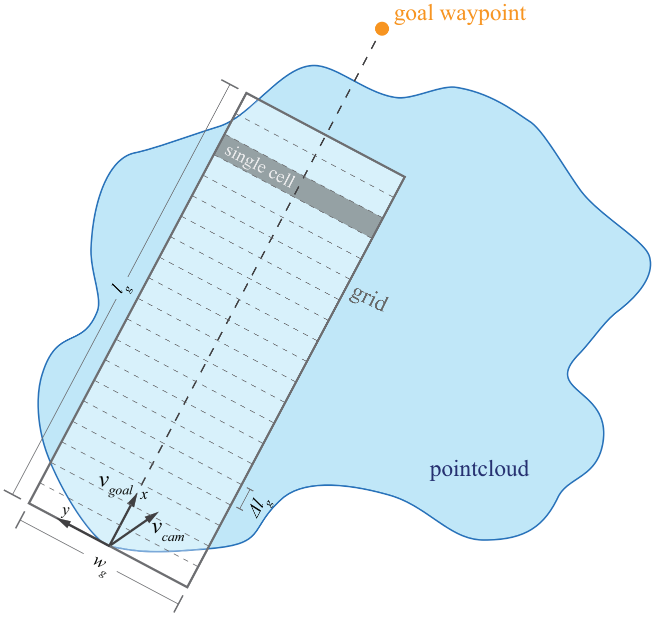

Figure 5. Top view of pointcloud and grid.

$$h_c(i) = \max_j \left(h(j) + \sqrt{r^2 - (j \cdot \Delta l_g - p_x(i))^2} \right) \ . \tag{5}$$

Figure 6 illustrates this projecting of the 3D space into 2D and the following height correction. The truncated pointcloud, grid and corrected height for the scene shown in Figure 4 is depicted in Figure 7. Afterwards, the resulting waypoints are sent to the spline interpolation for smoothing and execution by the controller.

Spline Fitting— In order for the position controller to smoothly track the resultant discrete waypoints it is necessary to fit them to a continuous reference trajectory. We selected clamped, uniform Basic Splines [7] of order $k = 6$, degree $d = 5$ for the parameterization of the reference trajectories. Basic splines are a practical choice for such a problem for several reasons. The curves are linear functions of the coefficients, such that data fitting is also a linear problem. Additionally, the order and respective degree of continuity are independent of the number of coefficients.

Basic Spline curves $\mathbf{S}(\mathbf{p}, \tau_\mathbf{i})$ are function of vertices $\mathbf{p}$ and an independent knot parameter τ_i, as well as the basis functions $\mathbf{N}(\bar{\tau}, \tau_\mathbf{i})$ and a knot vector $\bar{\tau}$. We equate the independent parameter τ_i with actual time t and represent each independent flat output [8], defined as the cartesian position $\boldsymbol{p}$ and heading ψ, as a single DOF curve, such that a trajectory is composed of four splines, defined by a matrix of vertices,

$$\mathbf{S}^r(\mathbf{p}, \tau_i) = \mathbf{N}^r(\bar{\tau}, \tau_i)\ \mathbf{p} \ , \tag{6}$$

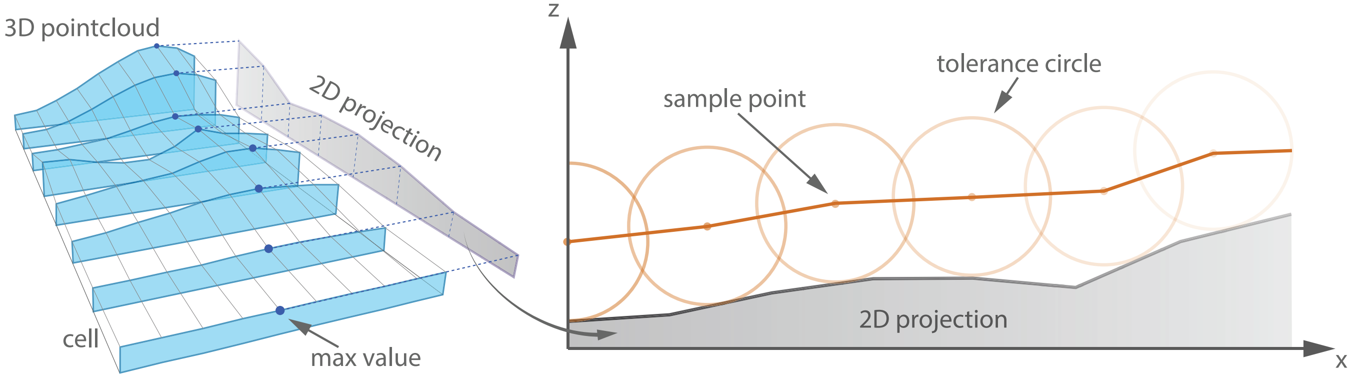

Figure 6. **Left side of the image shows the extraction of the highest points for each cell, which results in a 2D projection. The right side depicts the height correction on the 2D function.**

where

$$\mathbf{S}^r(\mathbf{p},\tau_i) \in \mathbb{R}^{n_{\text{via}}\times 4} \quad ,$$
$$\mathbf{N}^r(\bar{\tau},\tau_i) \in \mathbb{R}^{n_{\text{via}}\times n_{\text{vts}}} \quad ,$$
$$\mathbf{p} \in \mathbb{R}^{n_{\text{vts}}\times 4} \quad .$$

In practice there are boundary conditions on the reference trajectories. At time t_0 they are defined by the on-board position and velocity estimate, and at t_f by a desired zero relative motion in order to bring the system to a stable hover. (6) can then be modified:

$$\mathbf{S}^r(\mathbf{p},\tau_i) = \mathbf{N}^r(\bar{\tau},\tau_i)\begin{bmatrix}\mathbf{p}_0\\ \mathbf{p}^*\\ \mathbf{p}_\text{f}\end{bmatrix} \quad ,$$
$$\left[\mathbf{p}_0^\text{T}\mathbf{p}^{*\text{T}}\mathbf{p}_\text{f}^\text{T}\right]^\text{T} \in \mathbb{R}^{n_{\text{vts}}\times 4} \quad ,$$

and solved for some vertices $\mathbf{p_0}$ and $\mathbf{p}_\text{f}$ given $\mathbf{S}^r$ at t_0 and t_f for the desired number of time derivaties $r \in [0 \quad n_\text{k})$. The free vertices $\mathbf{p}^* \in \mathbb{R}^{n_p\times 4}$ for which $n_p = n_{\text{vts}} - 2n_{\text{bcs}}$ and $n_{\text{bcs}} = 2$, are then the linear control handle for fitting. We do this by solving a simple linear least squares (llsq) problem using the Jacobi SVD implementation from the Eigen Linear Algebra library [9] for each new set of waypoints. The llsq problem is unconstrained for performance, therefore to prevent dynamically infeasible motions the total duration of the trajectory is a sum of the time steps between each consecutive pair of waypoints, which are in turn computed given a desired average velocity.

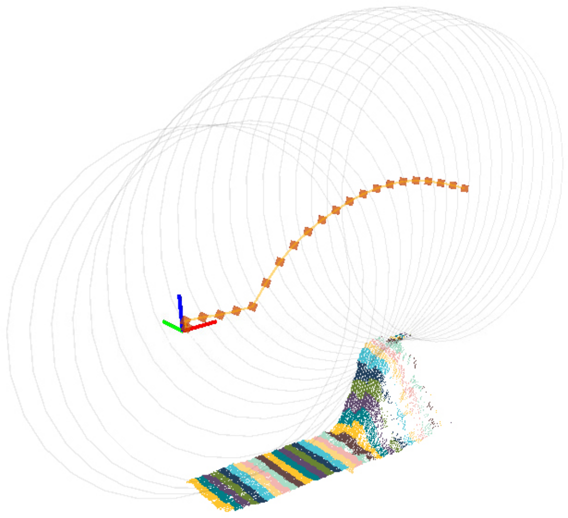

Figure 7. **Terrain following applied on scene of Figure 4. The different colors of the pointcloud depicts individual cells. The circles represents the tolerance radius for each sample point.**

Controller—ARDEA is an underactuated dynamical system with four control inputs (collective thrust and three torques) and six degrees of freedom. The controller makes use of the system's differential flatness, receiving the flat outputs and their derivatives and computes the corresponding control inputs. A cascaded structure of position and attitude controllers are used to control ARDEA's position and heading, as depicted in Figure 8. This structure is identical to that described in [5] with the exception of the spline interpolation component which provides the references to the position controller in place of a simple polynomial interpolation between single waypoints.

In the spline interpolation component, new fitted spline references are computed asynchronously in a functional block before the position controller. Upon each new controller tick, the spline interpolation component also provides cyclic transmission of the current reference i.e. the current point on the interpolated curves, given the relative time since the motion was initiated.

The controller is implemented entirely in C++, including the spline interpolation component, and runs on-board the BeagleBone Black embedded board, which also runs a real-time patched Linux kernel. The waypoints are transmitted to the spline interpolation component from the terrain following process running on the non real-time Intel NUC.

5. Simulation and Experiments

In the following we present our results from simulations and experiments with ARDEA. First we present our results in simulation. Afterwards, the terrain follower is tested on our flying system ARDEA in an lab environment. We conclude the section by an impression of the terrain follower in a field test on Mt.Etna.

Simulation and Results

Before implementing and testing our method on an actual robotic platform we simulated it. To ease the simulation

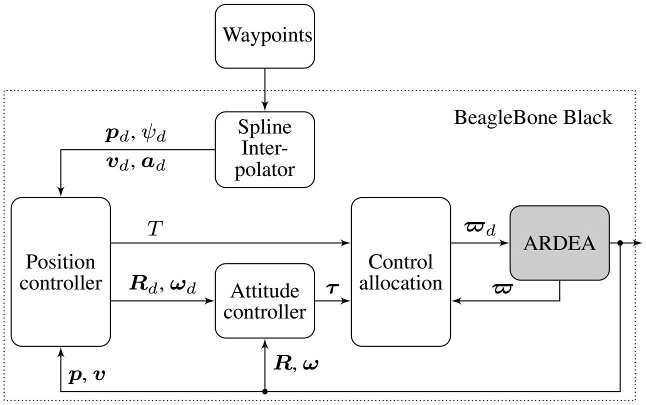

Figure 8. Structure of the cascaded position and attitude controller. The Spline Interpolator component provides an acyclic waypoint interpolation service in addition to cyclic querying of the trajectories to forward the flat output reference to the position controller.

complexity, the simulation environment is 2D and does not include any motion model. In the simulation, we compare the basic trajectory without height correction, the trajectories with height correction using either the nadir method or our method, and the best possible trajectory (ground truth). In the simulation we simulate our stereo cameras with a fan of rays as illustrated in Figure 9.

Figure 10 illustrates the simulation for an example terrain. The shape of the terrain is generated randomly as it is important to test our method on many different environment and to avoid focusing on special terrain layouts. The simulation generates the rough envelope of the terrain by using a sinusoid function with random phase. To provide finer details, the terrain is overlayed by higher frequency noise. For better visualization of the method, the terrain always starts with a short flat plateau.

As shown in Figure 10, the basic trajectory without height correction (blue line) stays very close to the surface and might lead to a crash of the system. In comparison the nadir method would avoid such a crash but the resulting trajectory (green curve) follows the underlying terrain with a delay. Our method (black curve) has no delay and is almost indistinguishable from ground truth (red curve).

To further compare the methods, we run the simulation 10000 times and calculated the error between points of the ground truth trajectory and the trajectories calculated using the different terrain following techniques. The results presented in Table 1 show that our proposed method follows the ground truth significantly closer compared to the nadir method.

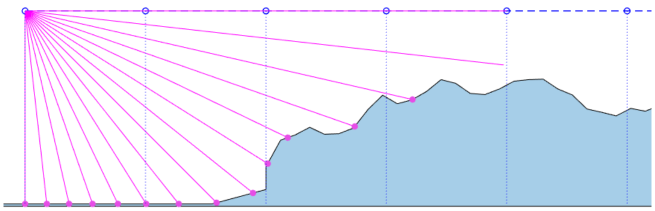

Figure 9. Illustration of ray simulation. The purple lines demonstrate the depth samples taken by the stereo camera system. Since the quality of the stereo depends on the baseline of the stereo setup and camera resolution the depth ray samples have a maximum range.

Table 1. Simulation Results

Method	Mean Error	Standard Deviation
Nadir	0.5431 m	0.0839 m
Our Approach	**0.1590 m**	**0.0659 m**

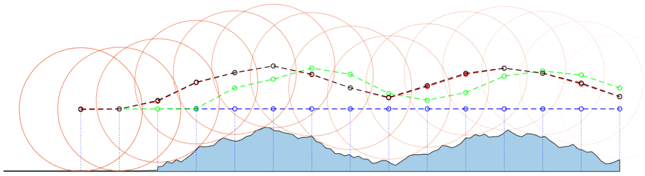

Figure 10. Illustration of our terrain following simulation. The blue terrain depicts one sample terrain out of the set of randomly generated terrains. The blue line shows the flight path without any height adaption. The black line depicts the desired height corrected path. The green one illustrates the simulation results of the nadir methord. The red one illustrates our approach. Dots on each flight path depicts waypoint samples. Orange circles depict the tolerance circles.

Lab Experiments

After verifying our method in simulation we implemented and tested our method on the actual robotic system. To evaluate our method, we constructed a variety of terrains in our laboratory. In the following, we present the result of three of these experiments. For all tests, the algorithm was running online and on-board the system, while ARDEA was flying the commanded trajectory.

As sanity check, we choose a fairly flat surface as first test. The flown path is depicted in Figure 11 as a sequence. The corresponding positions and velocities as commanded to the controller and as estimated by the on-board visual-inertial odometry are shown in Figures 14 and 15.

As second experiment, we placed a boulder in front of ARDEA's path. Figures 12 and 16 illustrate the output of our proposed method and the flown path. In the last image of the sequence shown in Figure 12, one can see that the corrected trajectory is raising again in height. This is because the terrain follower already perceives the next boulders, which are laying ahead of the robot.

In the last experiment, ARDEA is facing a pile of rocks, followed by a boulder steeper than in the second experiment. The sequence is illustrated in Figure 13, the commanded and estimated positions are shown in Figure 17.

Flight on Moon Analogue Side - Mt. Etna

To evaluate our method in an environment, which is similar to an extraterrestrial surface, we conducted a field test on Mt. Etna. The volcano features similar properties as our Moon. Figure 18 illustrates he sequence of ARDEA following the terrain. Because of safety requirements in the area, the system is attached to a tether, which is guided along, while ARDEA flies. However, the person holding the tether is not pulling on it and just acts in a case of emergency.

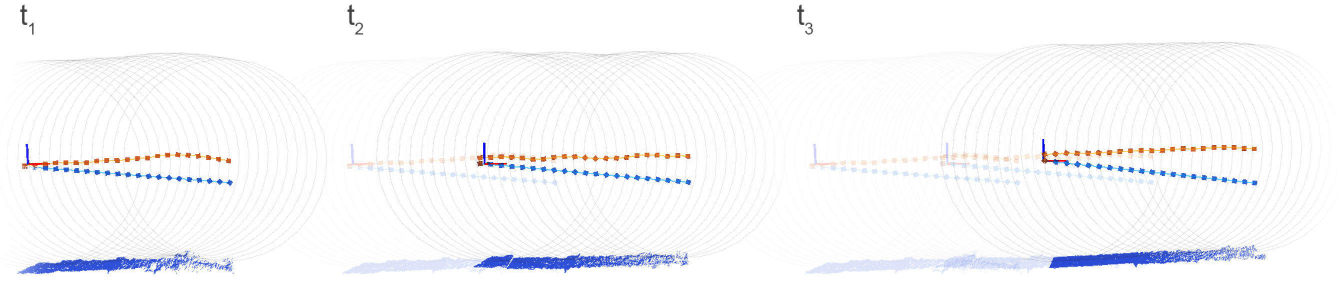

Figure 11. **Terrain Following applied to fairly flat surface. Blue points represent the point cloud. Blue line represents line to goal waypoint. Orange path represents path output by proposed method.**

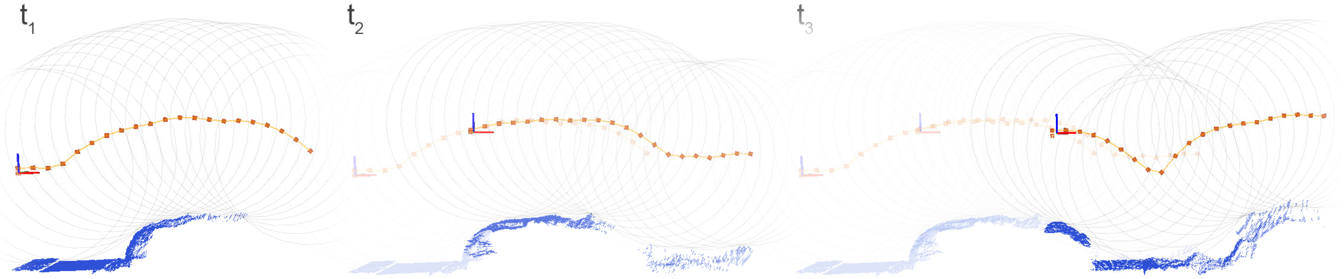

Figure 12. **ARDEA faces a boulder staight ahead. Blue points represent the point cloud. Orange path represents path output by proposed method.**

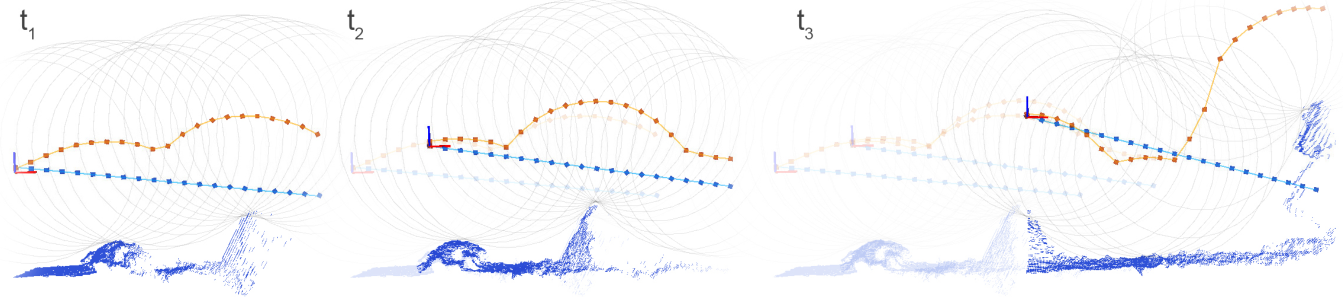

Figure 13. **ARDEA faces a pile of rocks, followed by a boulder. Blue points represent the point cloud. Blue line represents line to goal waypoint. Orange path represents path output by proposed method.**

6. Conclusion

In this paper we described a terrain following method for flying systems. The proposed method uses stereo cameras to perceive the area lying ahead. We first evaluated our method in simulation and demonstrated that it is superior to the nadir approach. Afterwards, we tested our method on a flying platform and showed the performance of our terrain following approach on different kinds of environments. We also gave an impression of the method flown by ARDEA on Mt. Etna. With that we were able to demonstrate that our proposed terrain following method is suitable for flights in low altitude.

The method is not restricted to following the ground floor. It can also be used to follow the curvature of a ceiling, if the stereo cameras can perceive this area, as for instance on our multicopter ARDEA. This can be generalized to follow any surface perpendicular to a predefined axis. Therefore, a potential application of this method is the exploration of long lava tubes.

Future work will include further flights with ARDEA on Mt. Etna with the proposed terrain following method. Furthermore, we plan on longer distance flights on open fields. Depending on the used flying platform and flight path, parameters for the presented terrain following migth have to be adpated.

Acknowledgments

This work was supported by the Helmholtz Association, project alliance ROBEX (contract number HA-304), the project ARCHES (contract number ZT-0033) and the EU-project AUTOPILOT (contract number 731993).

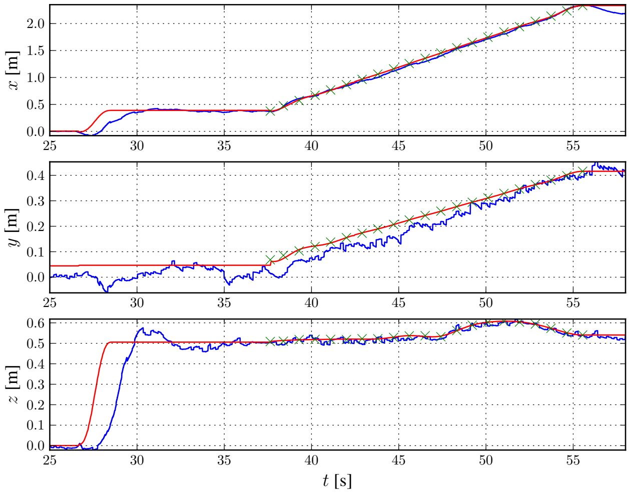

Figure 14. **Flight over flat ground. Shown are the measured positions from the on-board INS estimate (blue), desired position reference commanded to the position controller (red) and discrete waypoints (green ×) commanded to the spline interpolation.**

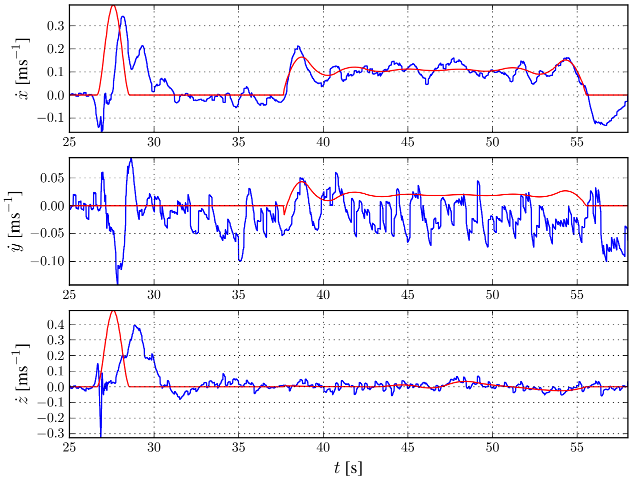

Figure 15. **Flight over flat ground. Shown are the measured velocities from the on-board INS estimate (blue) and the desired velocity reference commanded to the position controller (red)**

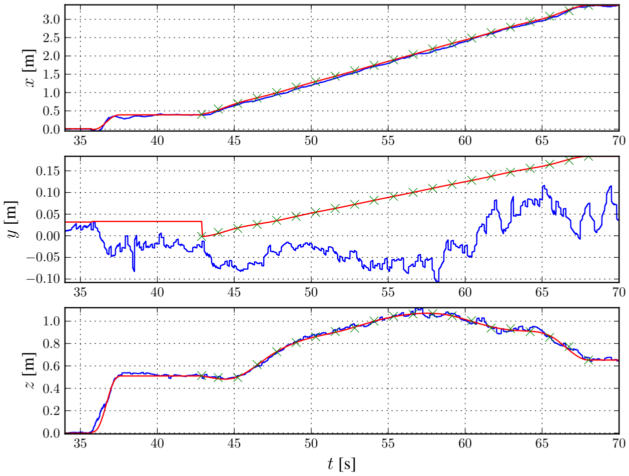

Figure 16. **Flight over a large boulder. Shown are the measured positions from the on-board INS estimate (blue), desired position reference commanded to the position controller (red) and discrete waypoints (green ×) commanded to the spline interpolation.**

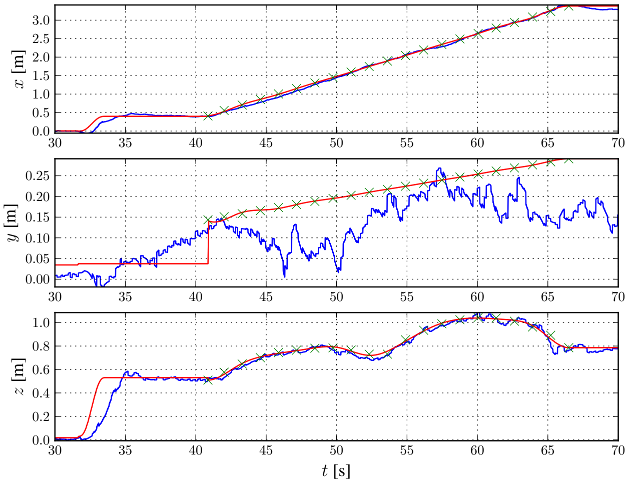

Figure 17. **Flight over mixed terrain. Shown are the measured positions from the on-board INS estimate (blue), desired position reference commanded to the position controller (red) and discrete waypoints (green ×) commanded to the spline interpolation.**

References

[1] W. Jiang, Y. Huang, W. L. Junjie Wu, and J. Yang, "A new approach for terrain following radar based on radar angular superresolution," in *International Conference on Communications, Signal Processing, and Systems (CSPS)*, 2015, pp. 223 – 231.

[2] F. Ruffier and N. Franceschini, "Aerial robot piloted in steep relief by optic flow sensors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 1266 – 1273.

[3] T. Netter and N. Franceschini, "A robotic aircraft that follows terrain using a neuromorphic eye," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002, pp. 129 – 134.

[4] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 328–341, 2008.

[5] P. Lutz, M. G. Müller, M. Maier, S. Stoneman, T. Tomić, I. von Bargen, M. J. Schuster, F. Steidle, A. Wedler, W. Stürzl, and R. Triebel, "ARDEA - An MAV with skills for future planetary missions," *Journal of Field Robotics*, 2019, (the first two authors assert equal contribution), (under review).

[6] M. G. Müller, F. Steidle, M. Schuster, P. Lutz, M. Maier, S. Stoneman, T. Tomic, and W. Stürzl, "Robust visual-inertial state estimation with multiple odometries and efficient mapping on an MAV with ultra-wide FOV stereo vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3701 – 3708.

[7] C. De Boor, *A practical guide to splines*. Springer, 2001. [Online]. Available: http://www.worldcat.org/isbn/9780387953663

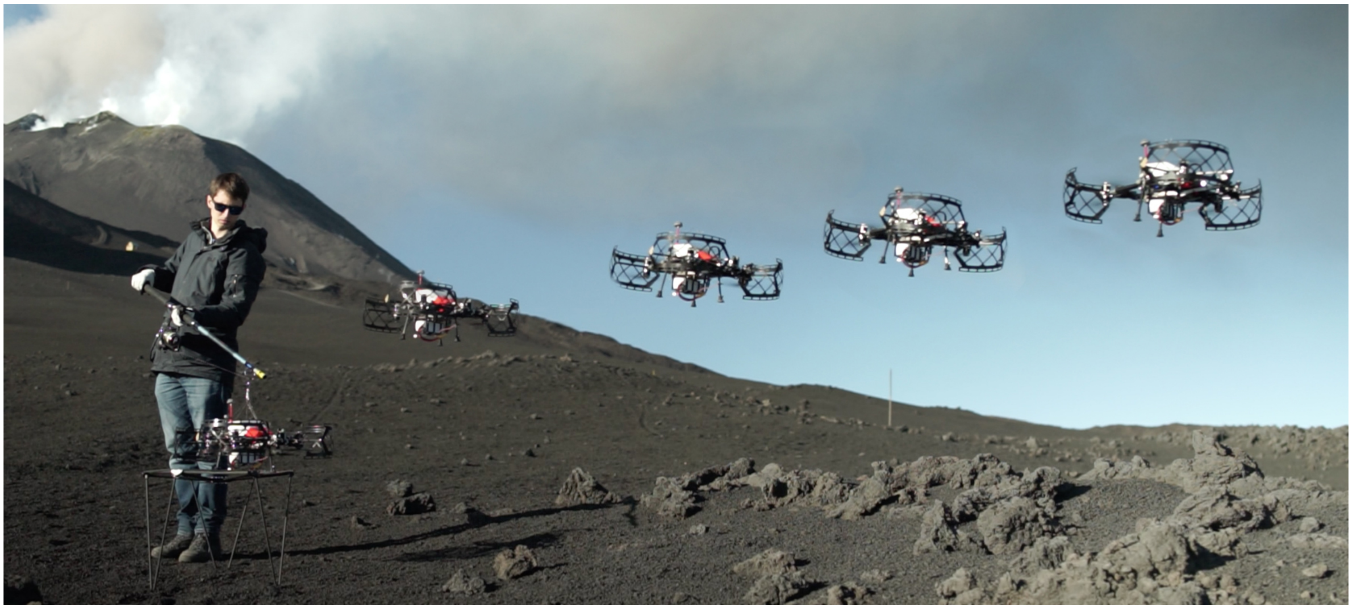

Figure 18. **ARDEA using the proposed terrain following on Mt. Etna. The person in the image is holding a safety tether, which is just used for emergency cases. The person is removed in parts of the image for better visualization.**

[8] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 2520–2525.

[9] G. Guennebaud, B. Jacob *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2010.

Marcus G Müller *is a researcher and Ph.D. student in the department of "Perception and Cognition" at the German Aerospace Center (DLR). He is the leader of the XRotor-Team at the Institute of Robotics and Mechatronics (DLR-RM). Here is working on autonomous navigation algorithms for MAVs. He joined DLR in the year 2016. Before joining DLR he conducted research at the Jet Propulsion Laboratory (JPL) of NASA in Pasadena, USA. Here he worked on visual inertial navigation for MAVs and on radar signal processing. Marcus received his Master's and Bachelor's degree in Electrical Engineering from the University of Siegen, Germany.*

Samantha Stoneman *joined the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR) in 2012. In 2011 she received her B.Sc. in Aerospace Engineering from San Diego State University and in 2014 her M.Sc. in Earth Oriented Space Science and Technology from the Technical University of Munich (TUM). She is now pursuing a Ph.D. on the topic of on-line motion planning at TUM's Computer Science faculty.*

Ingo von Bargen *is a researcher at the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR). In 2015 he received his B.Eng. in Mechatronics and Precision Engineering from the University of Applied Sciences Munich and in 2018 his M.Sc. in Robotics, Cognition, Intelligence from the Technical University of Munich (TUM).*

Florian Steidle *is a researcher in the department of "Perception and Cognition" at the German Aerospace Center (DLR). He received a Diploma in Mechanical Engineering at Technical University of Munich in 2011. His research interests include localization and navigation of mobile robots and multi-sensor fusion.*

Wolfgang Stürzl *is a senior research scientist in the department of "Perception and Cognition" at the DLR-Institute of Robotics and Mechatronics. He holds a Ph.D. (Dr. rer. nat.) from the University of Tübingen, Germany (2004) and a Diploma in Physics from the Technical University of Munich (1999). He was post-doctoral researcher at the Australian National University (2005-2006), at the Technical University of Munich (2007-2008), and at the Center of Excellence in Cognitive Interactive Technology (CITEC) at Bielefeld University (2009-2011). He is author of numerous research papers in the areas of computer vision, in particular using multi-camera and wide-angle imaging systems, and bio-inspired visual navigation of flying systems.*