An Autonomous Solution of the EMAV 2008 Outdoor Gate Crossing

Franz Andert, *Lukas Goormann*, *Florian Adolf*, and *Joerg Dittrich*
German Aerospace Center (DLR), Institute of Flight Systems
Lilienthalplatz 7, 38108 Braunschweig, Germany
{firstname.lastname}@dlr.de

http://www.imavs.org/

Abstract

Gate Crossing was one of the challenges within the 2008 EMAV Outdoor Competition. The ARTIS Team from DLR Braunschweig implemented a full autonomous solution, with the help of GPS localization of the helicopter, a vision based gate detection, and path planning to guide the vehicle through the gates with full autonomy. During the EMAV 2008 flight demonstration, the mission was not completed yet, but after additional research and implementation, several autonomous gate crossings were performed in spring 2009. This work will describe the development towards autonomous flights through gates.

1 Introduction

Topic of this paper is an autonomous solution for crossing gates with an unmanned aerial vehicle, which was one part of the EMAV 2008 outdoor competition [1]. The original scenario includes three gates with a size of each 6×6 m and given positioning and appearance as illustrated in figure 1. The competition differentiated between R/C-controlled flights within line of sight or wireless video link assistance and autonomous flights without manual user interaction except for safety issues. In the latter case, no further limitations were mentioned, which yields a lot of freedom about the vehicle and sensor configuration.

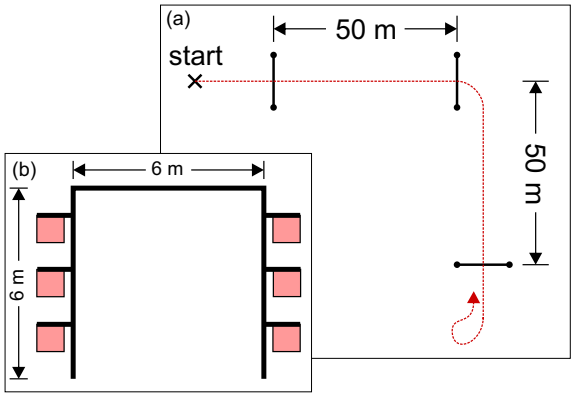

Fig. 1: Setup of the three gates, top view (a) and single gate, front view (b) as per description by the competition rules.

This paper presents the development of an autonomous solution for collision-free gate crossing. Note that the used vehicle (see sect. 3.1) is outside the micro flyer domain due to its size and payload capabilities. Hence, this work acts as a technology and methodology demonstration rather than a presentation how to win the challenge. A more general view on this topic is presented in [2].

The idea to fly autonomously through gates is the following: Basis is an auto-piloted vehicle with waypoint navigation capabilities. Since the gate sizes require a vehicle guidance with an accuracy of less than 2 m with respect to their absolute positions, crossings only with the given geo-referenced gate positions and a (D)-GPS-based vehicle localization are of very high risk. Hence, the vehicle must recognize the gates by itself. This is done using a camera installed at the vehicle. Image processing algorithms to estimate the gate positions are executed onboard. Based on these coordinates, the vehicle is guided through the gates. The paper focuses on image processing and generating waypoint-based flight instructions and concludes with a presentation of full autonomous gate crossings.

2 Motivation and Related Work

The paper addresses scientific challenges in aerial robotics like safe and precise navigation, sensor fusion, obstacle detection and avoidance, and mission and path planning. For example, current research covers obstacle avoidance behaviors [3, 4], mapping [5] and map-based path planning [6, 7], autonomous landing [8], detection and tracking of specific targets [9] or following moving ground vehicles [10].

Flying close to obstacles is a matter of particular interest. The paper discusses two problems that are often negligible in terrain with a lot of free space, but of major importance when flying into narrow passages, under bridges or through windows. As already mentioned, one problem to solve is the environmental recognition to increase knowledge about true object positions and to get the vehicle position relative to them.

Environmental sensing leads to the second problem. Flight instructions based on environmental sensors require a very accurate relative orientation between sensor and navigation solution. Navigation sensor uncertainties, drift, and imprecise calibration lead to biases of the estimated vehicle position and attitude angles. Additional sensors like cameras aligned with the vehicle are not aligned with the navigation system which leads to errors in sensor-based navigation instructions. Beside that, the orientation between

http://www.imavs.org/

sensors and vehicle is manually measured and imprecise as well. Hence, accurate sensor fusion between visual and inertial data is required, see e.g. [11] for an overview.

The presented approach copes with both problems. A calibrated monocular color camera is used for object detection. While flying towards an initially given gate position, the position relative to the camera is estimated with image processing techniques. At the same time, sensor misalignment is estimated by comparing GPS/INS-based vehicle movements with movements of the gate that result from multiple images. The presented algorithms enable these tasks without the need of separate calibration flights. Output is a gate position and orientation estimation relative to the navigation reference. This position is used to guide the vehicle through the gate on a determined heading. Due to drift errors in relative navigation, it is recommended to completely repeat the gate localization and sensor misalignment estimation process if another gate shall be passed.

The approach has similarities with simultaneous localization and mapping techniques [12] where environmental measurements are used not only to locate object positions, but also to re-locate the vehicle so that actual sensing information aligns with a map built from previous measurements. In the presented scenario, the relative vehicle position in the world is of major importance without regarding the global world map consistency. Hence, the *world* is re-located so that the sensor readings match with the map while the vehicle position is kept. An advantage is that no changes of vehicle navigation and guidance are needed. It is easy to readjust this image-based application for other vehicles, including those with proprietary auto-piloting systems having limited modification options.

3 Experimental Setup

3.1 Flight Testbed

The test vehicle is an auto-piloted model helicopter (fig. 2) with all computational power on-board. It is developed within the ARTIS (Autonomous Rotorcraft test bed for Intelligent Systems) project doing aeronautic and robotic research on unmanned helicopters [13]. Avionics payload comprises GPS, IMU and magnetometer for navigation, sonar altimeter for landing, wireless data links, manual R/C and video data links, a telemetry module and a flight control computer. The image acquisition and processing equipment consists of an industry color camera (Imaging Source DFx 31F03, settings: 1024×768 pixels, 15 fps, lens: $60° \times 48°$ field of view) and a separate vision computer (Intel Pentium 4 Mobile, 1.8 GHz).

Basis for robotic applications are an EKF-based navigation filter providing the actual flight state, especially the absolute position and attitude angles, and a flight controller that allows precise vehicle guidance and the input of high-level commands from external applications. Autonomous flights are accomplished with a loosely-coupled approach where the flight controller is not modified. The algorithms presented in this paper act as the application that triggers waypoint position instructions to the autopilot.

Fig. 2: Research UAV ARTIS: a combustion-engine-powered autonomous helicopter, 1.9m rotor diameter, 12kg gross weight

3.2 Scenario Reproduction

Test data with images from manual flights through gates are used as the basis for developing image processing algorithms. Hence, three wooden gates, vertically hold with tent ropes, were constructed by the ARTIS team following the sketch from the EMAV 2008 flight competition guideline. They were constructed in early 2008 before the EMAV competition started. Eventually, the EMAV 2008 hosts took these gates for the outdoor competition as well, one of them is shown in figure 3.

Fig. 3: One of the gates, placed at the glider port in Gifhorn-Wilsche were the EMAV 2008 outdoor competition took place.

4 Waypoint-Based Gate Crossing

Gate crossing is performed autonomously using waypoint flight instructions provided by the ARTIS autopilot. The flight is started with approximate knowledge about gate positions from an obstacle map. In this paper, a gate position denoted as (x, y, z, α) refers to the optimal geodetic point (gate center) and the direction (flight path azimuth) to cross the gate. The vehicle shall detect the gate while flying straight ahead towards its initially given position that may have some translational and rotational deviation to the true position. Hence, the basic maneuver is to fly to a waypoint next to an expected gate, to re-locate the gate with image processing algorithms (sect. 5) and to fly through. Complex object scanning flights are not required. To handle multiple gates, the maneuver can be repeated.

Figure 4 illustrates the process in detail. The pictures show the scenario from bird's eye view with the vehicle and

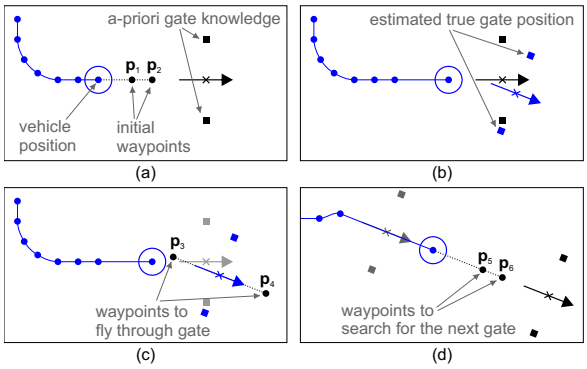

Fig. 4: Mission execution plan to fly through gates. Initial path (a), detecting the true gate position (b), generating fly-through waypoints (c), continuing with the next gate (d)

its already flown path, waypoints and the gate positions. Each gate is represented by its two posts, center and pass direction.

Two initial waypoints $\mathbf{p}_1$ and $\mathbf{p}_2$ are placed in front of a gate with different distances (fig. 4a). For the used vehicle and camera configuration, these waypoints are $30m$ and $10m$ in front of the gate where a good visibility is expected. The vehicle is instructed to fly to $\mathbf{p}_1$ and approach to the gate up to the nearer waypoint $\mathbf{p}_2$. On this way, images are processed to search for the true gate position (fig. 4b).

If the gate is visible and its position is estimated with a high confidence, two new waypoints $\mathbf{p}_3$ and $\mathbf{p}_4$ are added at a distance of $d = 10m$ in front of and behind the gate. Due to the camera's field of view, the posts become invisible when approaching closer than $10m$, and the vehicle must fly blindly through them. This is also a reason why the approach prevents direct vehicle control supported by environmental sensor data. If the search is not successful, another attempt is designated with a flight back to $\mathbf{p}_1$.

After a successful attempt, the mission is either completed or the next gate is searched (fig. 4d). The initial map is transformed by using the deviation between the initial and the estimated positions of the first gate. New waypoints $\mathbf{p}_5$ and $\mathbf{p}_6$ are set and the mission continues with the same procedure used for the first gate.

5 Image Processing Algorithms

This section describes obstacle detection with monocular vision to estimate the relative gate position and orientation for this specific scenario. The approach of this paper can be extended to more general scenarios by modifying this target finding method, for example by using laser range finders or stereo vision. Like the presented method, general approaches have to determine the sensor-relative orientation of a waypoint that leads through a passage.

5.1 Gate recognition

Each gate is constructed with three flags at each post for automatic recognition. Figure 5 shows the main steps used to locate the flag positions that will be the basis for locating the vehicle relative to the gate. The image processing approach to find and locate the gate flags consists of the following steps:

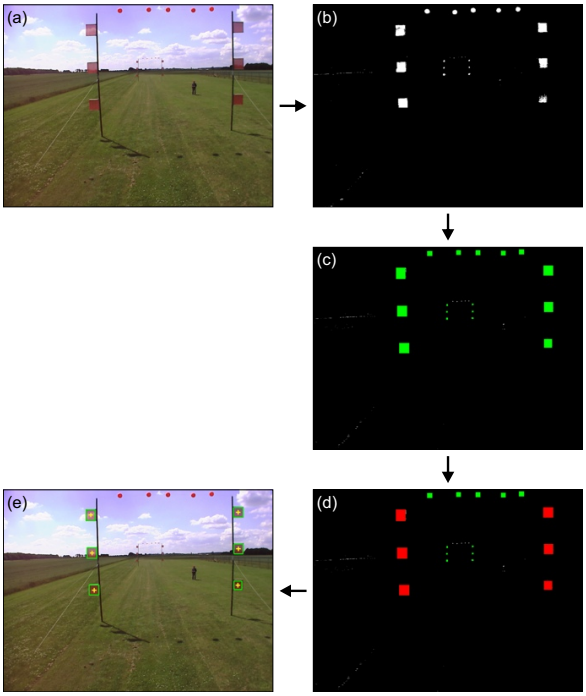

Fig. 5: Vision-based gate recognition. Camera image (a), segmented image (b), potential objects (c), chosen flag objects (d), and projection of this pattern into the camera image (e)

Segmentation: The image is segmented to produce a binary image (fig. 5b) with pixels that may represent parts of the flags (white) and those who do not (black). Segmentation is achieved with a pixel-based color filter (HSV color space) where all of the following criteria must be fulfilled to generate a white output pixel: close to red (hue between $300°$ and $25°$), saturation above a fixed threshold ($\approx 30\%$), and intensity value above a fixed threshold ($\approx 45\%$). These parameters have been identified manually from image sequences and can be modified to detect flags of different appearance, e.g. a different color.

Clustering: Neighboring white pixels of the binary image are merged into clusters. For each cluster, store size (number of connected pixels), centroid (center of gravity, i.e. average pixel position) and bounding box (minimum and maximum row and column, i.e. also width and height).

Cluster Filtering: Clusters are filtered out that are too small or too big (size $<$ 30 or size $>$ 3500 pixels here), and far away from squarish shape (width $<$ 0.5·height or width $>$ 2·height). Remaining clusters are marked in figure 5c.

Cluster Relationship: Use the centroids of the remaining clusters to find lines with exactly three clusters on it (or close to them) and that are almost vertical in the image, and out of these, similar line couples where the segment lengths between the clusters are nearly equal. If more than one couple is found, take the one with larger clusters and

http://www.imavs.org/

http://www.imavs.org/

a higher distance between the clusters. This line couple will probably represent the nearer gate. An example is shown in figure 5d. By projecting the clusters and their centroids back to the camera image, the correspondence between gate flags and the clusters is visible (fig. 5e). The cluster centroids mark six feature positions that can now be used to locate the vehicle relative to the gate. Ambiguities are eliminated by giving unique identifiers to the six features by their position in the image, e.g. beginning with the upper left.

Feature Tracking: There is no need to completely repeat the flag detection process for each image while processing sequences. Once all six flags have been found, they are observed using image feature tracking algorithms. For example, the classic Lucas-Kanade algorithm [14] is very fast and performs a good tracking if the flag corners are significantly distinguishable from the background. If the feature tracking is not successful or is of low confidence, features get lost during the image sequence and the flag detection process starts again from the beginning.

Sequence Cutting: Gate position estimation is based on either a single image, or an image sequence without ambiguities between multiple gates. They are distinguished by analyzing feature tracking results. If detected gate flags get lost by tracking and a new flag detection provides significantly different flag positions, a cut is done and a new sequence is started. Within this paper, an image sequence is defined that all included images refer to the same gate.

5.2 Gate localization

Now, the orientation of the gate relative to the camera is estimated for images where all six flags have been found. It is solved with camera resectioning techniques as used in photogrammetry and requires a calibrated camera with its internal camera parameters such as image center, focal length, and lens distortion. With that, the pinhole camera model is used to estimate the external orientation of the camera with respect to the gate. Figure 6 shows how the image feature points are related to the real flag positions.

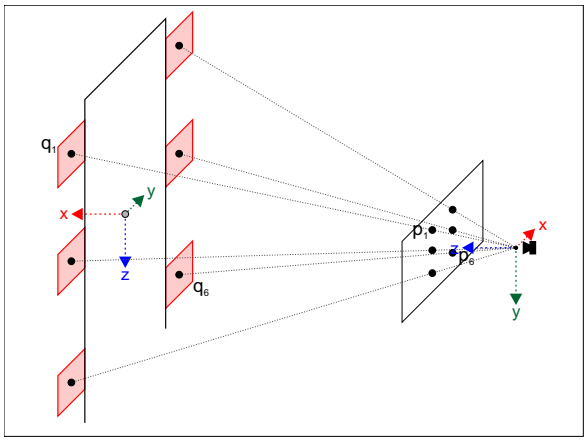

Fig. 6: Pinhole camera model to estimate the relative gate position

Resectioning is solved using the algorithm presented in [8]. It assumes a minimum of four coplanar object points $\mathbf{q}_j$ (here:$j = 1, \ldots, 6$) and their corresponding image projections $\mathbf{p}_j$, and finds the best fitting camera orientation $\mathbf{R}_{\mathbf{q}_i}$ and $\mathbf{t}$ (rotation matrix and translation vector). The algorithm consists of a linear initialization step that produces approximated results of $\mathbf{R}_{\mathbf{q}_i}$ and $\mathbf{t}$, followed by a non-linear multivariate Newton-Raphson iteration that produces a more accurate result by minimizing the reprojection error. Since scaling is not obtained with vision-based estimation algorithms, the size of the gate is given by the flag positions with respect to the gate center.

The origin of the gate coordinate system marks the point where the vehicle shall pass the gate in x-direction. Hence, $\mathbf{R}_{\mathbf{q}_i}$ and $\mathbf{t}_{\mathbf{q}_i}$ give a transformation between the actual camera orientation and the desired target.

With that, the global gate position can be calculated using the vehicle state and the camera alignment on the vehicle. Let $\mathbf{R}_{\mathbf{g}_i}$ and $\mathbf{g}_i$ denote the helicopter orientation relative to the gate estimated using the i-th image. With known camera alignment $\mathbf{R}_{hc}$ and $\mathbf{t}_{hc}$, it is

$$(1) \quad \mathbf{R}_{\mathbf{g}_i} = \mathbf{R}_{hc}\,\mathbf{R}_{\mathbf{q}_i}, \quad \text{and} \quad \mathbf{g}_i = \mathbf{R}_{hc}\,\mathbf{t}_{\mathbf{q}_i} + \mathbf{t}_{hc}.$$

Using the corresponding helicopter orientation $\mathbf{R}_{\mathbf{h}_i}$ and position $\mathbf{h}_i$ from the navigation filter, the global target $\mathbf{R}$, $\mathbf{t}$ is determined by

$$(2) \quad \mathbf{R} = \mathbf{R}_{\mathbf{h}_i}\mathbf{R}_{\mathbf{g}_i}, \quad \text{and} \quad \mathbf{t} = -\mathbf{R}_{\mathbf{h}_i}\,\mathbf{g}_i + \mathbf{h}_i.$$

Based on this rotation, the desired direction α to fly through the gate is determined by the rotation matrix elements. It is

$$(3) \quad \alpha = \text{atan2}(\mathbf{R}_{21}, \mathbf{R}_{11}).$$

Experiments have shown that this estimated gate position and direction is only roughly valid. While the helicopter approaches the gate, the estimated gate position moves dependent on the helicopter position. This strongly suggests systematic errors since the gate does not move in real. Primary error sources are vision-based, for example the relative orientation estimation accuracy dependent on the viewing perspective, alignment-based which means that biases in the camera orientation are existent, and navigation-based which refers to the measured geodetic vehicle orientation.

These errors are not large with few meters and may be negligible when navigating in large canyons. In the case of narrow passages as represented by the obstacle gates, it has become indispensable to perform a calibration of the external camera orientation with respect to the vehicle localization. Especially rotational errors must be minimized since they induce an incorrect gate orientation and too large position errors when viewing the gate from far distances.

5.3 Camera Alignment Optimization

Since multiple image and navigation measurements are taken while the vehicle approaches towards a gate, it is a straightforward procedure to integrate these measurements

to improve the gate detection. The presented improvements focus on the alignment and navigation errors.

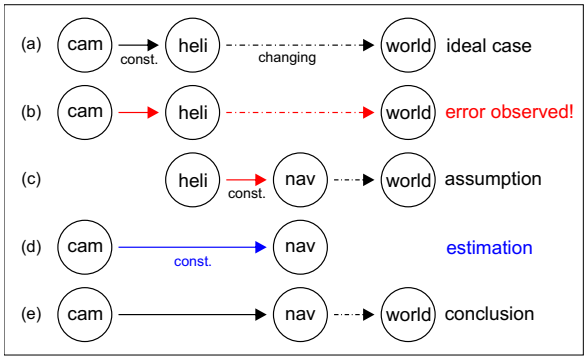

Fig. 7: Compensating errors in camera and helicopter alignment by estimating the transformation between camera and navigation solution

Figure 7 illustrates the approach how the gate positions are obtained anyhow. Case (a) illustrates the initial configuration. The camera is fixed at the helicopter whose orientation changing is due to vehicle movements. Both camera (*cam*) and helicopter (*heli*) orientation are assumed to be known without error, and the transformation of the gate into geodetic world coordinates is done as given in the equations 1 and 2. Since the gate is not moving, its estimated position should only be influenced by noise. This was not observed as illustrated by (b). Camera alignment as well as helicopter localization can be incorrect.

To compensate for these shortcomings, (c) shows the *world* regarded as the geodetic system where the helicopter orientation is absolutely known due to its navigation filter (*nav*). There will be an orientation error between the true helicopter position and its localization by the navigation filter. This error may change over time due to drift errors, but is assumed to be constant for a short time period, i.e. for a single gate passage. Differences between the world used here and real world coordinates, e.g. from obstacle maps, are assumed to be static.

Now, sensor fusion methods are applied, see (d). Both camera and navigation filter provide localization and with that, relative movements in the same reference system. This section will present an approach to compensate the camera misalignment in order to get an accurate gate localization.

The final conclusion is shown in (e) where the estimated orientation between camera and navigation coordinate system, and the orientation between navigation and world coordinates are successively used to get the camera position in the world and with that, the gate position in the world. It is not referenced to a true world of previously known maps, but to the navigation coordinate system, which is necessary to fly through the gate. The true helicopter position is never used since all commands refer to the navigation coordinate system.

The approach for camera alignment optimization is based on the idea that an ideal camera alignment would lead to a projection of all estimated gate positions from different vehicle locations to the same global coordinate as illustrated in figure 8. The problem is solved by searching for a camera alignment $\mathbf{R}_{hc}$, $\mathbf{t}_{hc}$ that minimizes the deviation of global gate positions taken so far from different locations.

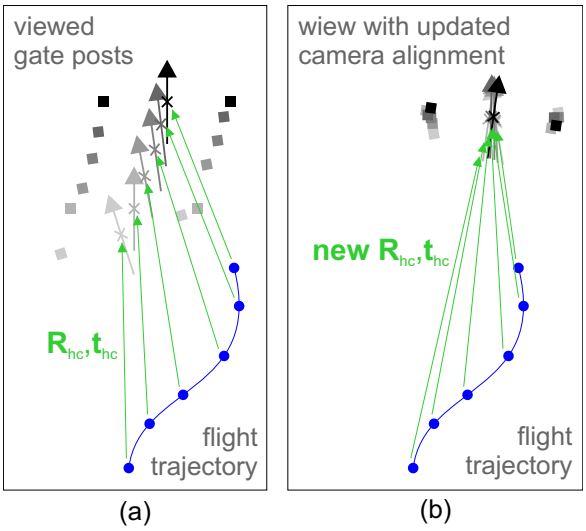

Fig. 8: Minimizing the gate position estimation uncertainty (a) by finding the correct camera alignment (b)

After taking n valid images from different vehicle positions, a point cloud with n points is determined using the global gate translation from equations 1 and 2 for each point. The error vector, denoted by $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z)^T$, represents the standard deviations of the point cloud in geodetic x, y and z-direction.

Let the camera alignment be determined by three position elements x, y, z and three rotation angles ω, ϕ, κ. For fixed gate measurements $\{\mathbf{R}_{\mathbf{g}_i}, \mathbf{g}_i | i = 1, \ldots, n\}$ and their corresponding vehicle positions $\{\mathbf{R}_{\mathbf{h}_i}, \mathbf{h}_i | i = 1, \ldots, n\}$, the point cloud of resulting gates is only dependent on the camera alignment, i.e. $\boldsymbol{\sigma} : \boldsymbol{\sigma}(x, y, z, \omega, \phi, \kappa)$. Tests have shown that the initial alignment translation (x, y, z) can be measured with comparatively high accuracy and that the camera roll angle (ω) is of low importance to locate the gate. The gate position mainly depends on the camera pitch (ϕ) and yaw angle (κ). Only they are optimized, and it is $\boldsymbol{\sigma} : \boldsymbol{\sigma}(\phi, \kappa)$. The optimal camera alignment will result in a minimum of $\boldsymbol{\sigma}$, which is approximated using a multivariate Newton-Raphson iteration.

The algorithm uses the Jacobian matrix $\mathbf{J}_{\boldsymbol{\sigma}}$ of $\boldsymbol{\sigma}$

$$\mathbf{J}_{\boldsymbol{\sigma}} = \left(\frac{\partial \boldsymbol{\sigma}}{\partial \phi} \; \frac{\partial \boldsymbol{\sigma}}{\partial \kappa} \right) = \begin{pmatrix} \frac{\partial \sigma_x}{\partial \phi} & \frac{\partial \sigma_x}{\partial \kappa} \\ \frac{\partial \sigma_y}{\partial \phi} & \frac{\partial \sigma_y}{\partial \kappa} \\ \frac{\partial \sigma_z}{\partial \phi} & \frac{\partial \sigma_z}{\partial \kappa} \end{pmatrix}, \tag{4}$$

where the derivatives of single values ϕ and κ are approximated using the difference quotients

$$\frac{\partial \boldsymbol{\sigma}}{\partial \phi} \approx \frac{\boldsymbol{\sigma}(\phi + t, \kappa) - \boldsymbol{\sigma}(\phi, \kappa)}{t}, \quad \text{and} \tag{5}$$

http://www.imavs.org/

http://www.imavs.org/

$$\frac{\partial \boldsymbol{\sigma}}{\partial \kappa} \approx \frac{\boldsymbol{\sigma}(\phi, \kappa + t) - \boldsymbol{\sigma}(\phi, \kappa)}{t}, \tag{6}$$

with a small $t = 0.0001$.

Now, the iterative approximation starts with ϕ_0 and κ_0 from initially given camera alignment angles, and the n-th iteration is

$$\begin{pmatrix} \phi_{n+1} \\ \kappa_{n+1} \end{pmatrix} = \begin{pmatrix} \phi_n \\ \kappa_n \end{pmatrix} - \mathbf{J}^{\dagger}_{\boldsymbol{\sigma},n}\, \boldsymbol{\sigma}(\phi_n, \kappa_n) \tag{7}$$

where $\mathbf{J}^{\dagger}_{\boldsymbol{\sigma},n}$ is the Moore-Penrose pseudo inverse of the Jacobian, with the partial derivatives calculated for the single values ϕ_n and κ_n. Termination conditions are a maximal number of iterations or $\|\mathbf{J}^{\dagger}_{\boldsymbol{\sigma},n}\, \boldsymbol{\sigma}(\phi_n, \kappa_n)\| < \epsilon$ with a threshold parameter, e.g. $\epsilon = 0.001$.

Finally, a new camera alignment $x, y, z, \omega, \phi_{n+1}, \kappa_{n+1}$ is used to calculate updated $\mathbf{R}_{hc}$ and $\mathbf{t}_{hc}$ and a target position as done in the equations 1 and 2.

6 Flight Tests and Results

Successful flight tests were performed after the EMAV competition in March 2009. The setup includes flight control, waypoint navigation, and the described gate crossing approach. To increase safety, the helicopter is advised to fly slowly with a speed of $1.5\ m/s$ during the automated gate approach and crossing phase. Two-dimensional top view plots in figure 9 show the quality of gate detection from six flights with different translation and rotation to the gate while approaching. The graphics show flight trajectories, and the gate posts and waypoints based on the estimated target position. The navigation solution is initialized while the helicopter is standing at the ground at the center between the two posts, heading perpendicular to the gate. Initial navigation coordinate is $(x, y, z, \alpha) = (0, 0, 0, 340°)$. Since the helicopter shall pass the gate in a height of three meters, the optimal gate crossing waypoint is $(x, y, z, \alpha) = (0, 0, -3, 340°)$ in the ideal case of no sensor drift.

The graphics from (a) to (d) show for successful flights with different a-priori settings for the initial gate position. Plot (a) shows the case where the optimal waypoint is given. As visible, the vision-based detection method provides nearly the same coordinate, and little path corrections are made before the vehicle crosses the gate. Note that an exact measurement of the gate center can be different from the initial given optimum $(x, y, z, \alpha) = (0, 0, -3, 340°)$ due to navigation sensor drift. In the flight presented in (b), the a-priori waypoint is displaced by three meters to the right. Hence, larger corrections are made to guide the vehicle through the gate. The flight path azimuth is similar to the direction of plot (a). As visible and contrary to plot (a), a flight without gate detection would hit the right gate post. Plot (c) shows flight results where the a-priori given waypoint is rotated by 15 degrees, and in plot (d), the initial waypoint is rotated and translated by three meters to the left. Gate crossing was successful, but as visible, the measured gate direction differs significantly from the results in (a) and (b). This direction is incorrect and the helicopter flies transversely through the gate. Errors in the gate direction estimation become visible.

Two other typical errors are presented in (e) and (f). In (e), the gate position was estimated incorrectly but treated as correct by the system. Here, the desired flight height is too large. The autonomous flight is interrupted by the safety pilot to avoid a possible crash. The plot in (f) does also show no completed gate crossing, but this case can be treated as a success in general. No confident gate position is obtained, presumably due to bad lighting conditions, wrong looking direction or frame drops. In this case, the mission execution plan guides the vehicle back to the starting point and starts additional attempts to search the gate. Since no gate is detected, the flight is canceled automatically after a maximum number of attempts.

Several autonomous flights are performed in addition to those from the presented graphics. The final result is that from 16 flights, 13 gate crossings were successful. In two cases, no gate was detected and the mission was automatically canceled. In only one case, the vehicle had to be stopped manually as presented.

These results show that autonomous flights through gates can be performed with the presented image processing and mission execution strategy. Additionally, the overall integration of image processing, mission execution and flight control is proven with the presented autonomous flights.

7 Conclusions

This paper presents an autonomous solution for the for the EMAV 2008 outdoor competition scenario, which addresses obstacle avoidance problems that appear for example in urban environments with limited free space. Here, an unmanned helicopter is guided through narrow gates. The approach is designed to run on auto-piloted aerial robots and consists of vision-based gate detection, camera misalignment compensation, global navigation-referenced gate localization, and flight mission execution steps.

The algorithms run completely onboard the vehicle, and they are tested tested with autonomous flights that show the functionality. Results are successful gate crossings without collision. The flight had to be canceled manually in only one out of 16 flights. Hence, the functionality is shown from a scientific perspective, but the failure rate is obviously too high for applied use. The robustness has to be increased in future applications, for example with additional sensors like laser range finders or multiple camera systems. Although a successful gate crossing is shown, some other errors concern the estimated gate direction, presumably due to bad perspective transformations of the specific planar pattern. Photogrammetric algorithms may provide more accurate results on objects with more feature points or non-planar three-dimensional objects with a “depth” where the perspective is better visible.

The presented approach requires previous knowledge about expected obstacles that was given in the specific scenario. Hence, upcoming research is going to extend the presented algorithm to handle a gate, window, or other small passages where only few attributes are given. For example, it is a challenge to fly through a large variety of rectangular passages where orientation and flight direction are obtained

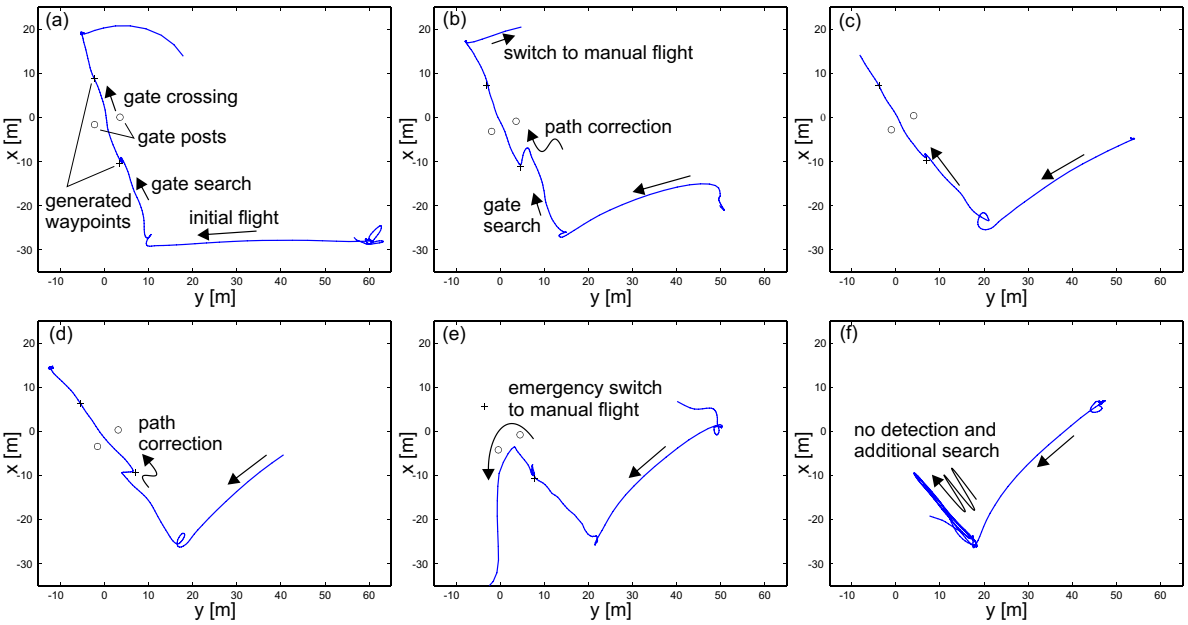

Fig. 9: Plots of vehicle trajectories, and estimated gates, illustrated by calculating the gate pole coordinates and the waypoints guiding the vehicle through the gate. Successful crossings (a) to (d), and error cases (e) and (f)

through the visible perspective projection. Furthermore, absolute obstacle sizes are not required if not only rotation and translation, but also scaling between navigation and image-based trajectories can be estimated accurately. This case does not require range sensors, and monocular vision will be capable of performing basic obstacle recognition.

The used hardware with its specific camera setup implicates that obstacles get out of view when the vehicle comes closer to them. Hence, the presented algorithms are designed to allow blind flights through the passage after measuring it. For best results especially to measure obstacles precisely from near distances, a horizontal field of view of 180 degrees is required. Special camera lenses, multiple cameras or other sensors like the mentioned laser scanners are qualified for that task. For future work, this improved sensor setup can decrease error-proneness. Since the gate will be always visible, this concept allows the implementation of visual servoing concepts with a tightly-coupled vision-based control approach. With that, the vehicle could change its movements while approaching to the obstacle and to fly directly through a passage without previous stopping. The goal is to enable agile flight maneuvers through cluttered environments using helicopters and proceedingly with fixed-wing aircraft also.

References

[1] DGON e.V., "EMAV 2008 flight competition – mission description and rules." Website: www.dgon.de/content/pdf/emav_2008_Rules_v07.pdf, November, 2007.

[2] F. Andert, F.-M. Adolf, L. Goormann, and J. S. Dittrich, "Autonomous vision-based helicopter flights through obstacle gates," *UAV 09 Symposium / Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 259–280, 2010.

[3] D. H. Shim, H. Chung, and S. S. Sastry, "Conflict-free navigation in unknown urban environments," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 27–33, 2006.

[4] Y. Watanabe, A. J. Calisey, and E. N. Johnson, "Vision-based obstacle avoidance for UAVs," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007.

[5] F. Andert and F. Adolf, "Online world modeling and path planning for an unmanned helicopter," *Autonomous Robots*, vol. 27, no. 3, pp. 147–164, 2009.

[6] S. Scherer, S. Singh, L. Chamberlain, and S. Saripalli, "Flying fast and low among obstacles," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 2023–2029.

[7] S. Hrabar, "3d path planning and stereo-based obstacle avoidance for rotorcraft UAVs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 807–814.

[8] C. S. Sharp, O. Shakernia, and S. S. Sastry, "A vision system for landing an unmanned aerial vehicle," in *IEEE International Conference on Robotics and Automation*, 2002, pp. 2793–2798.

[9] B. Ludington, E. Johnson, and G. Vachtsevanos, "Vision based navigation and target tracking for unmanned aerial vehicles," in *Advances in Unmanned Aerial Vehicles*, K. P. Valavanis, Ed. Dordrecht: Springer, 2007, pp. 245–266.

[10] A. Bernatz and F. Thielecke, "Navigation of a low flying vtol aircraft with the help of a downwards pointing camera," in *AIAA Guidance, Navigation and Control Conference*, 2004.

[11] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *International Journal of Robotics Research*, vol. 26, no. 6, pp. 519–535, 2007.

[12] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Vision-based odometry and slam for medium and high altitude flying UAVs," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1-3, pp. 137–161, 2009.

[13] F. Adolf, F. Andert, S. Lorenz, L. Goormann, and J. Dittrich, "An unmanned helicopter for autonomous flights in urban terrain," in *Advances in Robotics Research*, T. Kröger and F. M. Wahl, Eds. Berlin: Springer, 2009, pp. 275–285.

[14] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

http://www.imavs.org/