

Aerial Object Tracking from an Airborne Platform*

Andreas Nussberger¹, Helmut Grabner¹ and Luc Van Gool¹

Abstract—The integration of drones into the civil airspace is still an unresolved problem. In this paper we present an experimental Sense and Avoid system integrated into an aircraft to detect and track other aerial objects with electro-optical sensors. The system is based on a custom aircraft nose-pod with two integrated cameras and several additional sensors. First test flights were successfully completed where data from artificial collision scenarios executed by two aircraft were recorded. We give an overview of the recorded dataset and show the challenges to be faced with processing videos from a mobile airborne platform in a mountainous area. The proposed tracking framework is based on measurements from multiple detectors fused onto a virtual sphere centered at the aircraft position. To reduce false tracks from ground clutter, clouds or dirt on the lens, a hierarchical multi-layer filter pipeline is applied. The aerial object tracking framework is evaluated on various scenarios from our challenging dataset. We show that aerial objects are successfully detected and tracked at large distances, even in front of terrain.

I. INTRODUCTION

Over the past decade the market for Unmanned Aerial Vehicles (UAVs) is increasing continuously. However an accurate prediction of developments, especially in the civil market, is currently very difficult because there is one major challenge remaining: the integration of UAVs into the civil airspace. The civil airspace is a heavily regulated area with strict rules to ensure a safe operation for all participants. A simplified schematic overview of the available safety layers is shown in Figure 1. First there are procedures that every airspace user has to follow. For a controlled airspace there is also air traffic management available which organizes all participants in a given area. If we look at a closer area around a given aircraft there are transponder based technologies available to make an aircraft visible to others. Up to now not every aerial object is forced to include such a transponder

*This work is supported by armasuisse Science and Technology, affiliated with the Swiss Federal Department of Defense, Civil Protection and Sport
¹Andreas Nussberger, Helmut Grabner and Luc Van Gool are with the Computer Vision Laboratory, ETH Zurich, Switzerland {nussberger, grabner, vangool}@vision.ee.ethz.ch

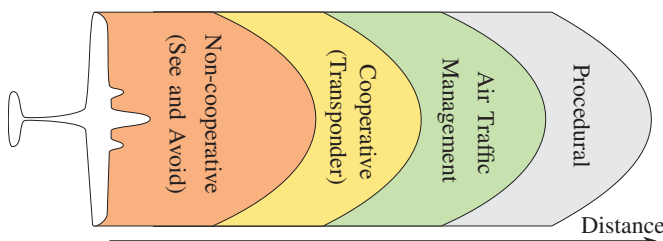


Fig. 1. Schematic overview of airspace safety layers. In this paper we focus on the See and Avoid part to detect aerial objects with cameras.



(a) Aircraft with sensor nose-pod (b) Example camera image

Fig. 2. The aircraft shown on the left was used to record a dataset containing image and meta data of real aircraft encounter scenarios.

based device. Especially gliders, paragliders and balloons are usually not equipped with such a device. Therefore the last safety layer is always the pilot himself who has to look outside and search for other aerial objects. This principle is called "See and Avoid".

If UAVs shall be integrated into this complex environment they have to comply with the existing standards and regulations. On the other hand these standards and regulations have to be extended to correctly handle the differences between a directly piloted aircraft and a remotely operated aircraft. Because there are no standards available yet, there is a large number of working groups and special committees working on the integration of UAVs into the civil airspace (e.g. ASTM F38, EUROCAE WG73, ICAO UASSG, RTCA SC-228).

Despite the ongoing activities for establishing the required regulations there is also one big technical challenge remaining: replacing the "See and Avoid" capability of the pilot by a technical system. This research area is also known as "Sense and Avoid" or "Detect and Avoid". As early results of different working groups have shown [1]–[3], a Sense and Avoid system shall provide an "equivalent level of safety compared to a human pilot".

First research activities with focus on Sense and Avoid have already started more than ten years ago within the NASA ERAST project [4] using a RADAR to detect other aircraft during the test flights. A similar project was started by the DLR in Germany [5] also based on a RADAR sensor. In parallel the Airforce Research Lab performed initial flight tests to detect other aircraft by electro-optical (EO) sensors based on an FPGA accelerated optical flow algorithm [6]. In 2009, the European Defense Agency started the "Mid Air Collision Avoidance System" (MIDCAS) project to develop an experimental Sense and Avoid system based on EO, infrared and RADAR sensors.

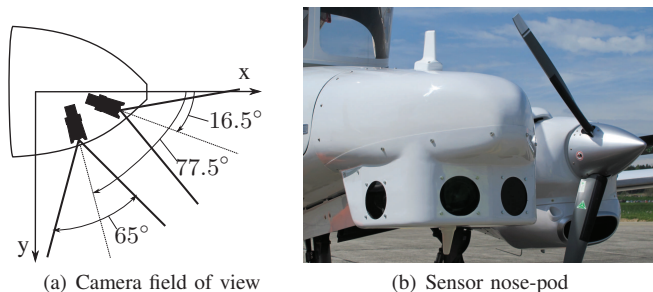


Fig. 3. Overview of the camera installation in the aircraft sensor nose-pod. The given camera orientation was chosen to fully cover slightly more than half of the proposed field of view for a Sense and Avoid system (horizontally $\pm 110^\circ$, see [1]), which is sufficient to simulate all relevant scenarios.

A popular way of detecting aerial objects within camera images is to use morphological filters for the sky region [7]–[9]. First closed-loop passive Sense and Avoid test flights based on morphological filters were demonstrated using a GPU accelerated real-time implementation [10]. There also exist other solutions e.g. based on a RADAR as primary sensor which provides an initial estimate of the aerial object angular position and a camera to increase the angular accuracy. The RADAR measurement is used to initialize a search window within the camera image where an edge detection algorithm is used to identify the aircraft [11].

In this paper we present an experimental Sense and Avoid system (see Figure 2) based on multiple sensors. In contrast to recent activities in obstacle avoidance with micro aerial vehicles [12], [13] we focus on detecting aerial objects using EO sensors (two cameras) at large distances and track them to decide if a given object is on a collision path. The cameras are a key component of the system because many smaller airspace users are not equipped with a transponder based device and some gliders, para-gliders or balloons will be hard to detect by a RADAR within ground clutter. The presented image processing pipeline is able to robustly detect aerial objects in the sky as well as in front of terrain. Measurements from multiple detectors and cameras are integrated into a sensor-independent spherical tracking framework with a multi-layer filter pipeline to remove false detections from ground clutter, clouds or dirt on the lens. We evaluate the proposed aerial object tracking framework on various scenarios from our challenging dataset recorded in the mountainous area of Switzerland. Experiments show that the traffic aircraft is successfully detected and tracked at large distances (average initial track distance greater than 1500 m) with only few pixels visible, even in front of terrain.

The structure of this paper is as follows. Section II describes the experimental system used to record the dataset containing various aircraft encounter scenarios. Section III gives an overview of the introduced processing pipeline to detect and track aerial objects. In Section IV, experimental results are presented, and in Section V we conclude the paper and discuss future work.

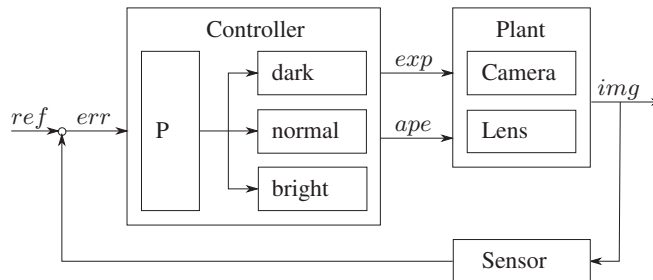


Fig. 4. The exposure controller adjusts the camera exposure time and the lens aperture value. Correct exposure is determined by evaluating the image histogram.

II. EXPERIMENTAL SYSTEM

In order to develop and measure the performance of a Sense and Avoid system, example data of real aircraft encounter scenarios is required. Therefore an experimental Sense and Avoid system was built up consisting of a data logger in the back of a Diamond DA42 aircraft and a custom nose-pod (see Figure 3(b)) containing the following sensors: an ADS-B¹ receiver and a FLARM² device to detect so called "cooperative traffic" which is actively transmitting its own position and velocity. On the other hand aerial objects which do not actively share their own position are called "non-cooperative traffic". To detect these types of airspace users (e.g. para-gliders or balloons) we use two cameras. Additionally an inertial measurement unit (IMU) and a GPS receiver are also integrated.

A. Hardware

The built-in cameras are based on an 8 mega pixel sensor with a bit depth of 8-bit or 12-bit and 20 fps or 10 fps respectively. Together with the installed lens each camera provides a field of view (FOV) of $65^\circ \times 51^\circ$, which results in an angular resolution of about 0.02° . For comparison, the human eye usually provides an angular resolution of approximately 0.01° , but only at 2° around the center of fixation [14]. The cameras have a global shutter which is synchronized across the cameras by an external trigger signal. A schematic overview of the camera installation in the aircraft nose-pod is shown in Figure 3(a).

Exposure control: having a robust exposure controller which correctly handles the huge range of different lighting conditions (e.g. haze, dark terrain, direct sunlight, etc.) is another important part of the system. Therefore a custom controller according to Figure 4 was implemented to dynamically adjust the exposure time (*exp*) and the lens aperture (*ape*) values based on the mean image intensity as reference. Special care had to be taken for very bright situations with

¹Automatic Dependent Surveillance Broadcast is a transponder based technology which is transmitting the own GPS position and velocity every second to other airspace users. Depending on the transponder power the maximum range can exceed 100 km.

²Flight Alarm is a proprietary, non certified traffic collision warning system. The maximum range is typically in between 3-5 km. Even though it is not a certified aviation product most of the gliders in countries around the Alps in Europe are equipped with such a device.

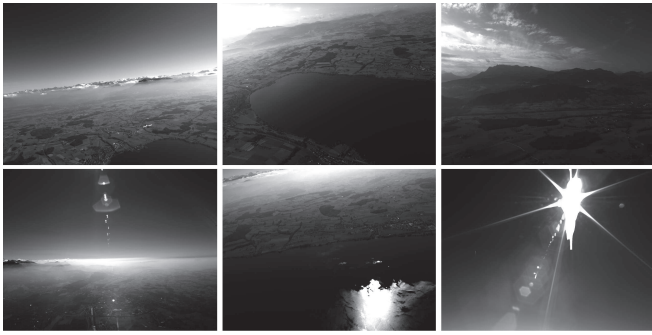


Fig. 5. Different lighting conditions extracted from the dataset.

e.g. direct sunlight to make sure we do not lose details in the dark parts of the image. On the other hand for example if only terrain is visible in the image, a trade-off has to be made between lighting up the image and motion blur introduced by the ego motion. Therefore the controller reference value was automatically adjusted based on the number of pixels above or below a given intensity value. The final parameter tuning was performed during pre-test-flights.

Lighting conditions: example images of various lighting conditions in the dataset are shown in Figure 5. The top row shows examples of common situations found in most of the recorded scenarios. The bottom row contains some challenging conditions such as reflections from haze, water, direct sunlight or the lens itself.

During the test flights all sensors produced continuously about 300 MB/s of data which was handled by a custom logging software to assure an accurate time handling in between the different sources. To enable the pilots to focus on the scenarios, the system was supervised and controlled from a ground control station.

B. Dataset

The main focus of a Sense and Avoid system is to detect and successfully avoid aerial objects on a potential collision path. To simulate this scenario two different aircraft were

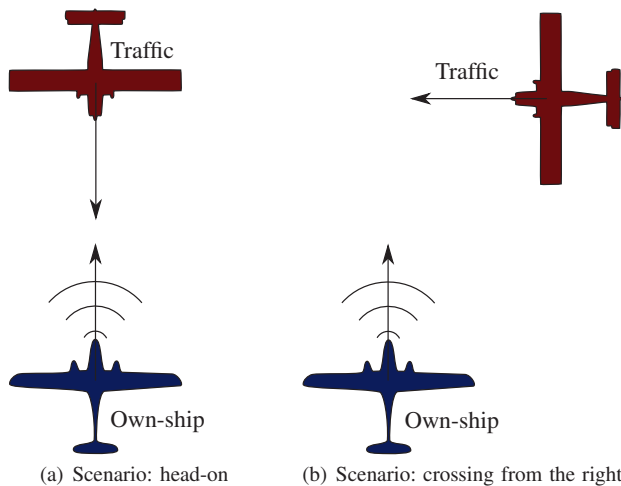


Fig. 6. These base scenarios were used as a reference for all test-flights.



Fig. 7. Comparison of aircraft patches (100 x 100 pixel) from a Pilatus PC-6 at different distances: 3.0 km, 1.5 km and 1.0 km. The top row shows a crossing from the right and the bottom row a head-on scenario.

used to fly artificial pre-defined scenarios on a collision course. For flight safety a minimal vertical separation was respected. The average aircraft velocity was around 100 knots resulting in closing speeds up to 200 knots. All scenarios were derived from one of the base scenarios shown in Figure 6, where the aircraft with the sensor nose-pod (own-ship) is shown at the bottom and the traffic aircraft (a Pilatus PC-6) at the top.

The standard head-on scenario was used to simulate a direct collision where no translational motion of the traffic aircraft is visible in the camera reference frame and only the size of the shape is increasing. The crossing from the right is the more general case where two aircraft are on a constant angle collision course, another dangerous situation every pilot is aware of. By modifying the closing angle and speed in between the two aircraft various situations were recorded, e.g. traversal of both cameras by the traffic aircraft. Other applied variations include for example a wing-rock of the own-ship to simulate massive ego motion or an avoid maneuver of one of the aircraft or even both of them. To include a representative overview of available lighting conditions, all scenarios were repeated with different orientations with respect to the sun and in front of terrain or with the sky as background.

The final recorded dataset includes more than 40 scenarios and 5 hours of video and meta data. This includes the ADS-B, FLARM, IMU, GPS and EO sensors of the own-ship which were recorded for each separate scenario. In addition the ground truth of the traffic aircraft was recorded using a differential GPS (D-GPS).

Distance comparison: small equally sized patches of the traffic aircraft at fixed distance intervals are shown in Figure 7. The top row is taken from a crossing from the right scenario where the traffic was flying at a slightly higher altitude than the own-ship. In the bottom row, cutouts of a head-on scenario are shown. These scenarios are usually more difficult because the visible cross-section of the traffic aircraft is minimal.

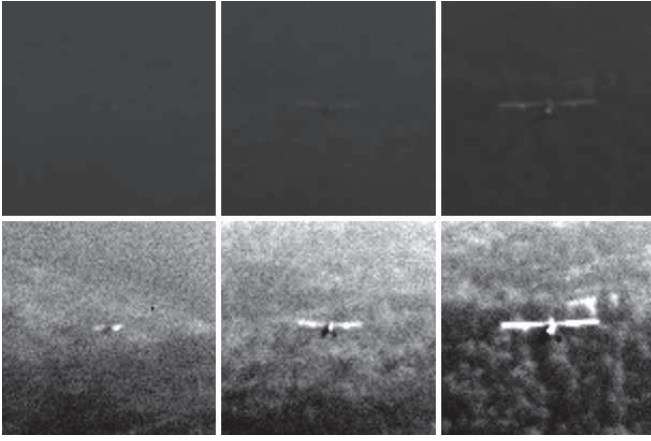


Fig. 8. Comparison of aircraft patches (100 x 100 pixel) from a head-on scenario in front of a mountain at 8-bit (top) and 12-bit (bottom) and different distances: 3.0 km, 1.5 km and 1.0 km.

Bit depth comparison: a head-on scenario where the traffic aircraft is coming closer in front of a mountain, while the sun was simultaneously visible in one of the camera edges, is shown in Figure 8. We show a direct comparison between image data available in 8-bit and 12-bit. The upper row shows the 8-bit equivalents at different distances while the lower row shows the corresponding 12-bit image. For visualization purposes the original 12-bit image was requantized into an 8-bit band. It is obvious that especially for such challenging lighting conditions a higher bit depth provides significant advantages.

III. DETECTION AND TRACKING OF AERIAL OBJECTS

This section focuses on the image processing framework proposed for the detection and tracking of aerial objects from an aircraft. We use a tracking by detection approach with the main visual steps shown at the top of Figure 9. First, we estimate the horizon line to separate each frame into a sky and a terrain region. Second, different detectors are applied based on the estimated horizon. Third, the detections from all detectors and cameras are converted to sensor independent measurements and fused for the tracker. The fourth step

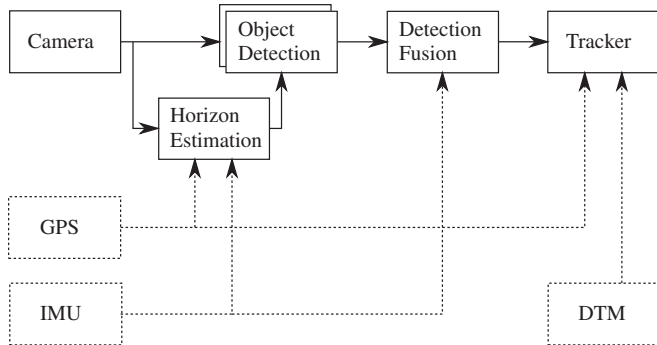


Fig. 9. The image processing pipeline with the main visual steps at the top. Additional meta information from a GPS receiver, an inertial measurement unit (IMU) and a digital terrain model (DTM) is shown by dotted lines.

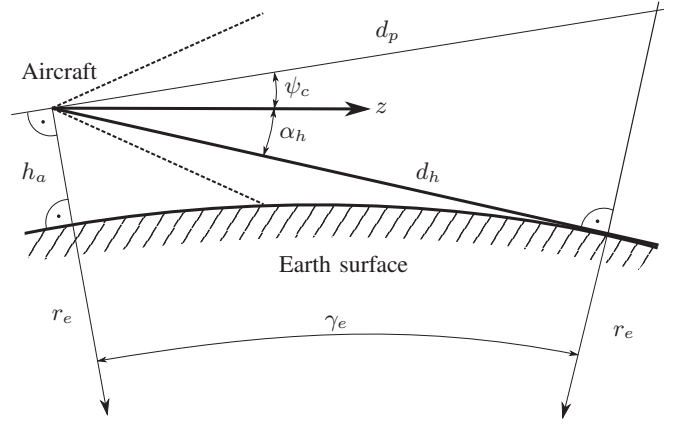


Fig. 10. We calculate an initial horizon estimate based on the the aircraft altitude, attitude and the assumption of the earth being a sphere with constant radius.

includes the spherical tracker and the track verification steps. Additional meta information such as GPS or IMU measurements and a digital terrain model (DTM) which are used throughout the processing pipeline are visualized by dotted lines. The detailed explanation of the different blocks is part of the following subsections.

A. Horizon Estimation based on Aircraft Attitude

The horizon estimation is a multi-step procedure required by the aerial object detectors (see Section III-B) where we use different detectors and parameters for the sky and terrain parts of the images.

Initial estimate: first we calculate an initial estimate based on the aircraft position and attitude according to Figure 10. The aircraft attitude is mapped from the aircraft reference frame to the camera reference frame using the extrinsic camera calibration. The horizon pitch angle α_h is given by the tangent from the aircraft to the surface of the earth and the z-axis of the camera reference frame. With the assumption of the earth being a sphere with radius r_e we can derive α_h from the aircraft altitude h_a and the camera pitch angle ψ_c , see equations (1) to (4).

$$d_h = \sqrt{(r_e + h_a)^2 - r_e^2} \quad (1)$$

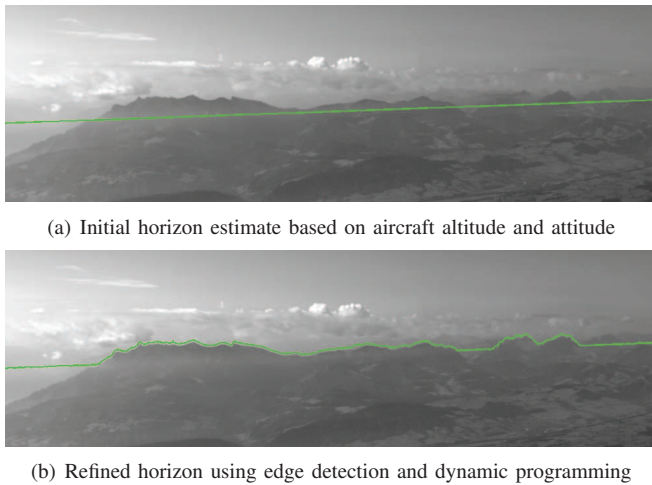
$$\gamma_e = \arccos\left(\frac{r_e}{r_e + h_a}\right) \quad (2)$$

$$d_p = \tan(\gamma_e) \cdot (r_e + h_a) \quad (3)$$

$$\alpha_h = \arccos\left(\frac{d_h}{d_p}\right) - \psi_c \quad (4)$$

The horizon roll angle in the camera reference frame is directly given by the camera roll angle. The accuracy of the estimated horizon line is primarily affected by the angular accuracy of the IMU. When flying at high altitude or above nearly flat terrain this initial estimate already provides reasonable results. At lower altitudes and especially in mountainous terrain a refinement is required.

Refinement: in a second step we calculate a refinement based on the initial estimate. Because this initial estimate



(a) Initial horizon estimate based on aircraft altitude and attitude

(b) Refined horizon using edge detection and dynamic programming

Fig. 11. Example images for the horizon estimation steps.

indicates the position of the horizon on flat terrain, we just allow a refinement above the existing estimate to account for protruding mountains. The refinement is based on edges and a dynamic programming algorithm [15]. Example images are shown in Figure 11.

There are currently still issues with scenarios where we have heavily textured clouds at the horizon. Because we make sure by our initial estimate that the horizon is not too low, the precision of the later refinement is not critical. It is usually not an issue if we apply an aerial object detector tuned for terrain background to a sky region. In contrast applying a detector optimized for sky regions to terrain will result in lots of false detections. To further improve the robustness additional cues (e.g. intensity or gradients [16]) could be integrated into the estimation process.

B. Object Detection

Our processing framework allows to include multiple independent detectors which provide measurements for the tracker. The main challenges are the very small size of the aerial objects and the ego-motion of the own-ship. Currently we use the following two detectors.

Morphological operations: morphological close-minus-open filters are widely used to detect aerial objects in a Sense and Avoid environment [8], [10] because they provide reasonable results at relatively low computational costs. A main issue of this approach is the limitation to the sky region of an image. When applying the morphological filters to an area in the image with terrain as background they would only provide a massive amount of false detections.

Image differencing: to extend the aerial object detection to image areas with terrain as background, a common scenario when flying at low altitude in a mountainous area, we use an image differencing pipeline [17]. First we extract key-points from every image and search for matches in between consecutive frames. Based on these matches we estimate the transformation from the old frame to the current one. By warping and subtracting the old from the current we get the detection candidates. Based on the estimated horizon we

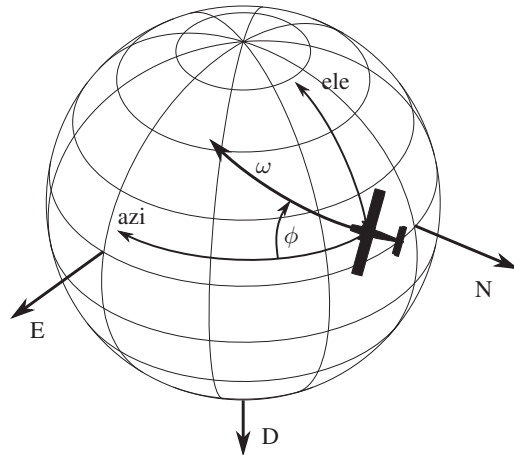


Fig. 12. Virtual tracking sphere fixed to a north-east-down reference frame at the global aircraft position.

apply different thresholds for the sky and terrain region of the image in the following binarization step. The final detections are extracted by labeling the connected components. Using different thresholds for the sky and terrain part usually enables us to achieve larger detection distances for the sky region.

In contrast to the morphological filters the image differencing allows us to detect aerial objects not only above the horizon but also in front of terrain. However even if we use different thresholds for the sky and terrain parts we typically achieve larger detection distances with the morphological detector for the sky region.

C. Detection Fusion

Another key contribution of our tracking framework is the decoupling of detections and tracks from the sensor reference frame. We propose a virtual sphere around the aircraft as shown in Figure 12. The sphere is independent of aircraft attitude and fixed to a north-east-down reference frame centered at the global aircraft position.

All detectors provide their measurements in the camera pixel reference frame. In the Detection Fusion step, the raw detections from every single camera and detector are transformed from pixel coordinates to spherical azimuth and elevation angles and fused to measurements for the tracker as points on the surface of the virtual sphere. The pixel values are first converted to azimuth and elevation angles in the camera reference frame using a standard camera pinhole model accounting for radial and tangential lens distortions [18]–[20]. The second step includes the transformation from camera reference frame to the tracking sphere by using the camera attitude given by the aircraft IMU. In the last step, detections from overlapping cameras or multiple detectors are fused based on their position and size.

FOV restriction: according to [1] a Sense and Avoid system shall provide a vertical FOV of $\pm 15^\circ$. Therefore we discard detections outside the recommended area.

This architecture allows us to easily integrate additional sensors to the system by a proper external calibration with

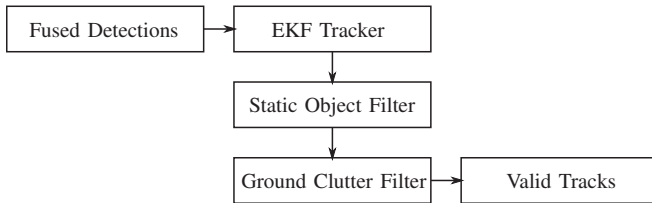


Fig. 13. False detection filter steps for track verification.

respect to the tracking sphere, e.g. additional cameras with different resolutions or wavelengths. In addition we do not have to take special care for overlapping or non-overlapping camera FOVs.

D. Tracking and Verification

As described in the previous section we project all detections to a global virtual sphere around the aircraft position shown in Figure 12. For the aerial object tracking we have implemented a constant angular velocity Extended Kalman Filter (EKF) with the states shown in equation (5) and the non-linear update equation (6). The azimuth (azi) and elevation (ele) angles specify the global track position on the surface of the unit sphere. The track velocity is given by the track heading ϕ and the angular velocity ω defined as the track velocity on the surface of the sphere divided by the sphere radius. Changes in track velocity or heading are handled by the corresponding process noise ν_ϕ and ν_ω .

$$\mathbf{x}_{ekf} = [azi, ele, \phi, \omega]^T \quad (5)$$

$$\mathbf{x}_{ekf}(t+1) = \mathbf{x}_{ekf}(t) + \Delta t \cdot \begin{bmatrix} \cos(\phi) \cdot \omega + \nu_{azi} \\ \sin(\phi) \cdot \omega + \nu_{ele} \\ \nu_\phi \\ \nu_\omega \end{bmatrix} \quad (6)$$

Tuning the detectors requires making a trade-off between detection range and false-detection rate. With increasing distance the objects get smaller, visually less distinctive and represented by fewer pixels. To achieve a useful detection range we have to take into account a large amount of false positives. To separate the aerial objects from clutter we propose the multi-layer filter architecture shown in Figure 13.

EKF Tracker: in a first step we update the tracks in our EKF filter with new detections. If a detection is not assigned to an existing track, a new track is initialized. A detection is successfully assigned to an existing track if it lies within a given area defined by the predicted track position and the corresponding covariances. All created tracks undergo a verification phase for the next couple of frames. During this time additional measurements are required and the candidate track has to successfully pass all the additional filter steps to be declared valid. For all tracks we keep a history of their previous pixel and angular positions which will be used throughout the filter steps.

Static object filter: in the second step, we filter static objects which are typically created due to lens pollution, clouds or if parts of the own aircraft are visible in the

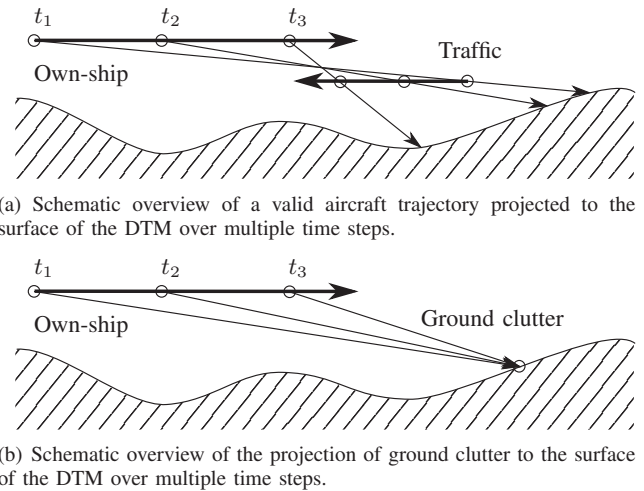


Fig. 14. Comparison of an aircraft trajectory and ground clutter projected to the surface of the digital terrain model.

camera. Note that static can refer to 'constant pixel position' or 'constant position on the virtual sphere'. In our dataset e.g. the pitot probe from the right wing is partially visible in the right camera. Due to vibrations and minor flapping of the wings the pitot probe is an ideal candidate for the image differencing detector. Dirt on the lens is usually less critical because it is heavily out of focus. Due to our virtual tracking sphere and the ego motion of the aircraft we are able to filter these static objects by analyzing the history of a track's pixel positions and compare them with the global track motion in azimuth and elevation angles. To avoid discarding aerial objects on a constant angle collision course a candidate track is only removed if its size is constant over time.

Ground clutter filter: the third step is to reduce the amount of ground clutter generated by the image differencing detector, typically a result of stationary objects on the ground changing their appearance when flying over at low altitude. To separate ground clutter from a valid aerial object we project the candidate track history to the surface of a DTM. In Figure 14 we show a comparison between the projection of a valid aircraft trajectory and the projection of a false track due to a static object on the ground to the surface of the DTM. Using the DTM to analyze the motion of a candidate track on the terrain surface allows us to successfully remove false tracks from stationary objects on the ground.

Valid tracks: finally if a new candidate track passes all the proposed filter steps it will be declared as "valid", but even a valid track has to continuously pass all the filter steps.

IV. EXPERIMENTS

In this section we focus on the evaluation of the processing pipeline based on our challenging dataset. First, we give an overview of the results across different scenarios. Second, we present a detailed analysis of our filter architecture to explain the challenges occurring when tracking aerial objects from an airborne platform.

TABLE I
COMPARISON OF FIRST DETECTION AND VALID TRACK DISTANCES WITH THE CORRESPONDING TIME TO COLLISION

	Scenario details			First detection		Valid track		False tracks	
	Type	Background	Duration	Distance	TTC	Distance	TTC	Num.	t_{max}
A	Head-on	Sky	28 s	2780 m	26.2 s	1874 m	17.6 s	1	1 s
B	Head-on	Sky	27 s	2649 m	24.0 s	1622 m	14.5 s	0	-
C	Head-on	Terrain	25 s	1350 m	11.6 s	980 m	7.8 s	0	-
D	Head-on	Terrain	24 s	2810 m	22.4 s	1116 m	9.0 s	11	12 s
E	Crossing	Sky	37 s	2960 m	36.8 s	2830 m	35.3 s	2	2 s
F	Crossing	Sky	48 s	2977 m	48.0 s	2588 m	42.2 s	0	-
G	Crossing	Sky	31 s	2833 m	29.5 s	1831 m	17.7 s	2	2 s
H	Crossing	Terrain	31 s	2370 m	23.3 s	1550 m	14.2 s	18	8 s
I	Crossing	Terrain	33 s	2767 m	30.2 s	1489 m	14.0 s	2	5 s
K	Crossing	Terrain	43 s	2596 m	37.8 s	1593 m	23.4 s	4	4 s
Average	-	-	33 s	2609 m	29 s	1747 m	19.6 s	4	3.4 s

A. Evaluation Criteria

The evaluation of our processing pipeline is based on the scenarios shown in Figure 16. For each scenario we analyze the first detection by one of the detectors and the moment the track is declared valid. As evaluation criteria we chose the remaining distance in between the own-ship and the traffic aircraft and the time to collision (TTC).

Distance: the remaining distance in between the two aircraft is defined as the euclidean distance between the GPS position of the own-ship and the D-GPS ground truth of the traffic aircraft.

Time to collision: for the calculation of the TTC we define the closest point of approach (CPA) as the position of the own-ship at the time where the distance to the traffic aircraft is minimal. With the assumption of a constant velocity the TTC is given by the distance from the current position to the CPA divided by the current speed.

In addition we show the total number of false tracks for each scenario. To allow a comparison between the track alive time of the traffic aircraft and the number of available false tracks we show the maximum alive time of all false tracks (t_{max}). This value can directly be compared with the TTC of a valid track, because once a track is declared valid, we usually do not lose it again due to the continuously increasing size of the traffic aircraft, if it is on a collision path.

B. Results

All of the chosen scenarios start at an initial distance of 3 km with the traffic aircraft already within the FOV of one of the cameras and end at the CPA. An overview of the results achieved across various scenarios from our dataset is shown in Table I. The scenario type defines the corresponding base scenario (see Figure 6), while the background indicates if the traffic aircraft is visible in front of the sky or terrain.

As shown in Table I the average TTC for having a valid track is about 15 to 20 seconds and the corresponding distance is greater than 1500 m. Depending on the own-

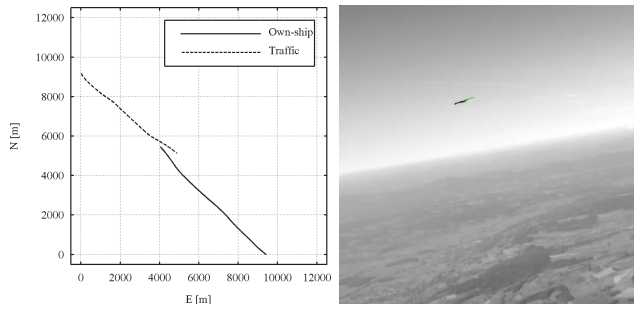
ship this should be enough time to execute an avoidance maneuver, e.g. by the pilot on the ground.

The best results are achieved on crossing from the right scenarios with the sky as background as expected. Here we have a larger visible cross-section of the traffic aircraft and a lower closing speed compared to the head-on scenarios. In addition, the contrast between the traffic aircraft and the background is typically higher for scenarios in front of the sky. The worst TTC is resulting from head-on scenarios in front of terrain. In Section IV-C we explain the reasons based on scenario C which was even flown at equal altitude.

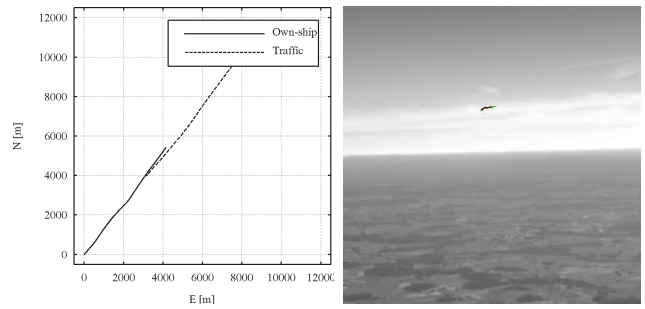
False tracks are usually a result of clouds or ground clutter which was not correctly removed by our filter pipeline, e.g. because of wrong assignments of measurements to existing tracks. There are two scenarios (D and H) which have a significantly higher total number of false tracks and also a longer maximum alive time (t_{max}) compared to the others. The main reason for these false tracks are lens flares which are also discussed in the following subsection.



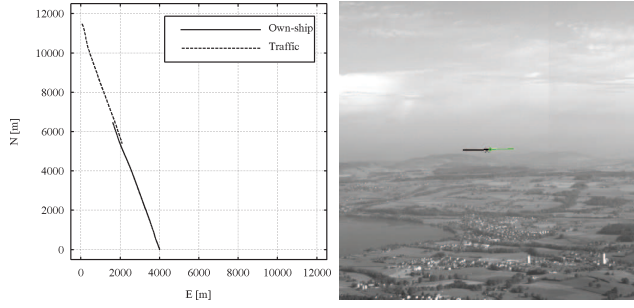
Fig. 15. Comparison of aircraft patches (100 x 100 pixel) from a Pilatus PC-6 for the scenarios C, E, and H (left to right). The top row shows a patch for the initial detection and the bottom row the corresponding image where the track was declared valid.



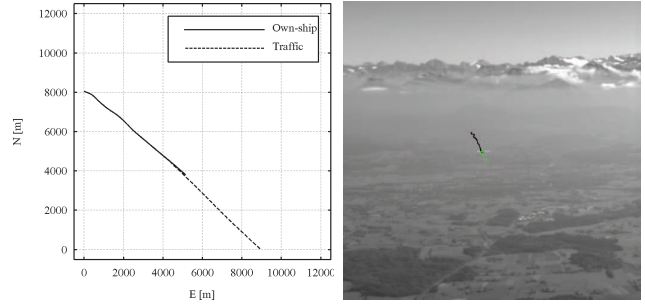
(a) Scenario A: head-on with the sky as background.



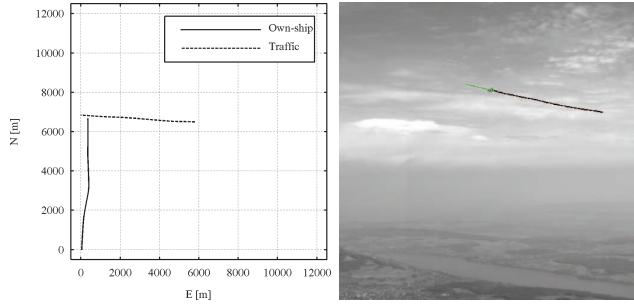
(b) Scenario B: head-on with the sky as background.



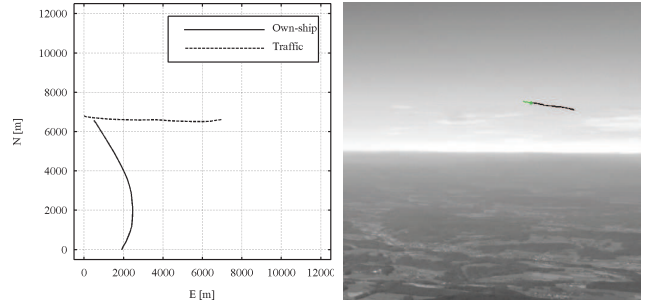
(c) Scenario C: head-on at equal altitude.



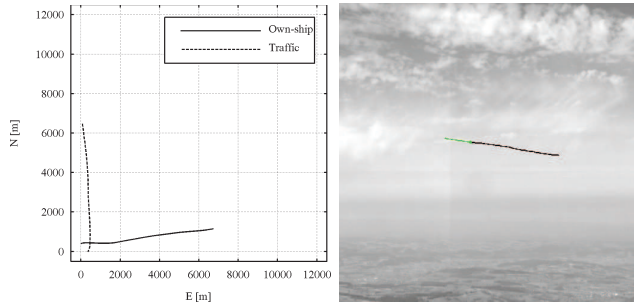
(d) Scenario D: head-on with terrain as background.



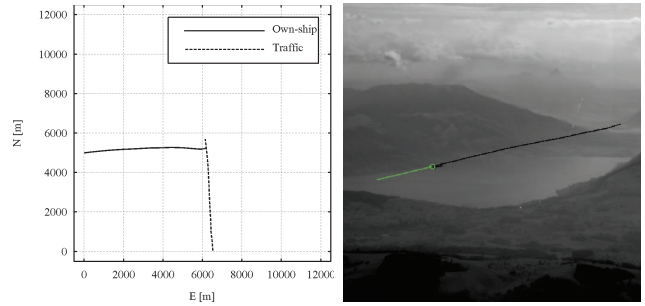
(e) Scenario E: crossing from the right with the sky as background.



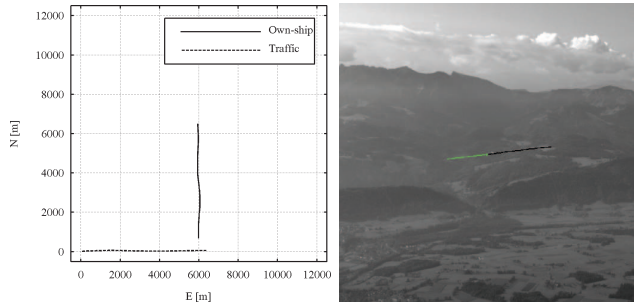
(f) Scenario F: crossing from the right with the sky as background.



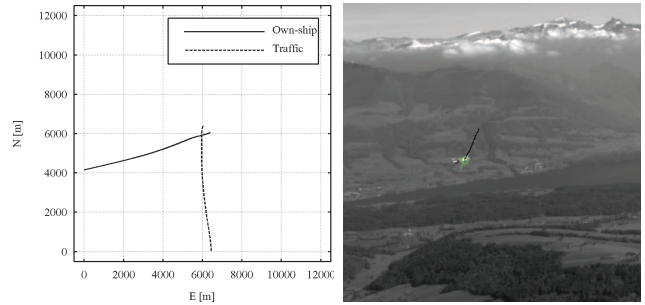
(g) Scenario G: crossing from the right with the sky as background.



(h) Scenario H: crossing from the right with terrain as background.



(i) Scenario I: crossing from the right with terrain as background.



(j) Scenario K: crossing from the right with terrain as background.

Fig. 16. Overview of all evaluated scenarios. On the left we show the GPS positions of the own-ship and the traffic aircraft mapped to a rectangular north-east reference frame. On the right a cropped example image from the tracker is shown with the track history in black and the prediction over one second in green (best viewed in color).

TABLE II
DETAILED ANALYSIS OF THE FILTER PIPELINE

	Scenario C	Scenario E	Scenario H
Average fused detections per frame	7.8	2.3	11.0
Average initialized tracks per frame	1.8	0.4	1.7
Precision for EKF Tracker only	0.42	0.69	0.19
Precision for EKF Tracker + Filters	1.0	0.85	0.21

C. Detailed Analysis

From the above scenarios we select three representative ones to analyze the steps of our processing pipeline in detail. An overview of the chosen scenarios is shown on the left of Figures 16(c), 16(e) and 16(h). Figure 15 shows the corresponding image patches for the first detection (top row) and the moment a track is declared valid (bottom row) from Table I.

The first chosen scenario C is certainly the most challenging one regarding the detection distance but also one of the most dangerous ones: a head-on scenario at equal altitude. In this situation the closing speed is the worst possible and the shape of the incoming traffic is very small while it is only slightly increasing in size at large distances. Only shortly before the potential collision the increase rate starts to grow significantly. The second scenario E is a crossing from the right where the traffic is above the horizon. This represents the common situation of flying at high altitude where other aerial objects are typically visible in front of clouds or the clear sky. The last scenario H is again a crossing from the right but now in front of a mountain, a common situation at lower altitudes and in mountainous areas. In addition the elevation of the sun is very low and therefore lens flares occur in the camera images.

For both crossing scenarios E and H we achieve satisfactory results with initial track ranges greater than 1.5 km, even if the traffic is in front of terrain. On the selected head-on scenario C we perform significantly worse. While for the initial detection until a potential collision there are more than 10 seconds remaining, just about 7 seconds are left when a valid track is available. This might be sufficient for a fast autonomous avoidance maneuver to avoid the collision but during normal operation a longer response time will be preferred. Multiple factors add to this result: due to the head-on scenario we have a high closing speed and the smallest possible visual cross-section. Because of the available terrain and the low altitude above ground the traffic aircraft appears exactly at or below the horizon. Therefore only the image differencing detector is applied which is just able to detect changes in the size of the visual cross-section because hardly any translational motion occurs.

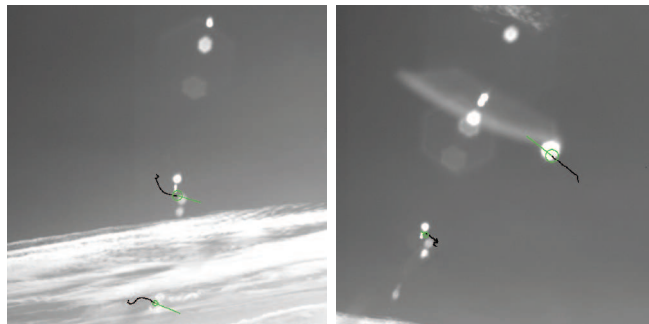


Fig. 17. Examples of false tracks due to lens flares.

In Table II we give a detailed overview of our processing pipeline by analyzing the different stages. In the first row we show the average number of detections per frame for each of the scenarios. This is the fused output of the detectors converted to measurements for the tracker. Based on these measurements the EKF tracker updates existing tracks and initializes new ones if the measurement was not assigned to an existing track. The average number of initialized tracks based on new detections is shown in the second row. The resulting precision for the bare EKF tracker without any additional filter steps is shown in the third row. The definition of precision is given by equation (7). After enabling the static object and ground clutter filters we get the final results shown in the bottom row.

$$precision = \frac{correct\ tracks}{correct\ tracks + false\ tracks} \quad (7)$$

If we compare the three scenarios, two facts attract attention. First, there are a lot more detections per frame for the scenarios C and H compared to the crossing scenario E. Second, there is a significant difference in precision after enabling the filter pipeline between the first and the head-on scenario. Both issues arise from the same reason: in scenarios C and H we have obviously a lot more false detections per frame than in scenario E. There are different reasons for false detections depending on the detector type. As long as the morphological detector is only applied within the sky region of the image it usually just fails on some parts of heavily textured clouds. For the image differencing, errors are already introduced by an imprecise transformation estimation and the warping afterwards. In addition when flying at low altitude even buildings and other stationary objects changing their appearance when flying over are candidates for false detections.

For the head-on scenario C the false detections are primarily induced by buildings on the ground which are correctly removed by the DTM projection filter step. As a result, the precision is increasing significantly when enabling the filter pipeline. The many false detections in the crossing scenario H are mostly due to lens flares occurring throughout the complete scenario. They get detected by the image differencing because they move across the image depending on the aircraft orientation with respect to the sun. Because

this type of false detection is currently not correctly handled by our filter pipeline, the precision is obviously the worst. Examples of false tracks due to lens flares are shown in Figure 17.

The lower number of detections in the crossing scenario E is due to the benign environmental conditions. There is no direct sunlight visible in the lens and there is no city with lots of buildings on the ground underneath. The remaining false detections are due to clouds.

V. CONCLUSIONS AND FUTURE WORK

We have presented an experimental Sense and Avoid system initially used to record example data from real aircraft encounter scenarios. Based on the recorded dataset we have outlined the major challenges to be faced with while processing image frames recorded from an airborne platform. With our multi-detector approach we successfully detect and track incoming aircraft in the sky as well as in front of terrain. The tracker based on a virtual sphere allows us to fuse measurements from multiple sensors across sensor gaps and reduce false detections due to lens pollution or static parts of the own-ship. By combining the spherical tracker with the DTM we are able to efficiently remove false tracks due to static objects on the ground. By evaluating the processing pipeline on challenging scenarios from our dataset we achieve promising results with an average initial track distance of more than 1500 m and a remaining TTC of about 20 s.

In the future we plan to integrate the ADS-B and FLARM messages into our tracking framework. This would allow the tracking of cooperative aerial objects at even larger distances and improve the overall situational awareness.

ACKNOWLEDGMENT

The authors would like to thank armasuisse Aeronautical Systems for the efficient collaboration during the flight campaign. In addition, this work would not have been possible without the help and support of Markus Möckli and Daniel Ambühl from RUAG Aviation, especially during the preparation and realization of the test flights.

REFERENCES

[1] ASTM International, "Standard specification for design and performance of an airborne sense-and-avoid system," ASTM International, 2007.

[2] International Civil Aviation Organization, "ICAO cir 328, unmanned aircraft systems (UAS)," 2011.

[3] Federal Aviation Administration, "FAA order JO 7610.4 special operations," 2013.

[4] R. C. Wolfe, "NASA ERAST non-cooperative DSA flight test," National Aeronautics and Space Administration, Tech. Rep., 2003.

[5] B. Korn and C. Edinger, "UAS in civil airspace: Demonstrating "sense and avoid " capabilities in flight trials," in *Proc. of Digital Avionics Systems Conference*, 2008.

[6] J. Utt, J. McCalmont, and M. Deschenes, "Test and integration of a detect and avoid system," in *Proc. of AIAA "Unmanned Unlimited" Technical Conference*, 2004.

[7] R. Carnie, R. Walker, and P. Corke, "Image processing algorithms for UAV "sense and avoid"," in *Proc. of International Conference on Robotics and Automation*, 2006.

[8] D. Dey, C. Geyer, S. Singh, and M. Digioia, "A cascaded method to detect aircraft in video imagery," *International Journal of Robotics Research*, vol. 30, pp. 1527–1540, 2011.

[9] J. Lai, L. Mejias, and J. J. Ford, "Airborne vision-based collision-detection system," *Journal of Field Robotics*, vol. 28, pp. 137–157, 2011.

[10] L. Mejias, J. S. Lai, and J. J. Ford, "Flight trial of an electro-optical sense-and-avoid system," in *Proc. of International Congress of the Aeronautical Sciences*, 2012.

[11] L. Forlenza, G. Fasano, D. Accardo, A. Moccia, and A. Rispoli, "Image processing algorithm for integrated sense and avoid systems," in *Proc. of Society of Photo-Optical Instrumentation Engineers*, 2010.

[12] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Autonomous obstacle avoidance and maneuvering on a vision-guided MAV using on-board processing," in *Proc. of International Conference on Robotics and Automation*, 2011.

[13] S. Ross, N. Melik-Barkhudarov, K. Shankar, A. Wendel, D. Dey, J. Bagnell, and M. Hebert, "Learning monocular reactive UAV control in cluttered natural environments," in *Proc. of International Conference on Robotics and Automation*, 2013.

[14] D. G. Green, "Regional variations in the visual acuity for interference fringes on the retina," *Journal of Physiology*, vol. 207, pp. 351–356, 1970.

[15] W.-N. Lie, T. C.-I. Lin, T.-C. Lin, and K.-S. Hung, "A robust dynamic programming algorithm to extract skyline in images for navigation," *Pattern Recognition Letters*, vol. 26, pp. 221–230, 2005.

[16] G. Baatz, O. Saurer, K. Kser, and M. Pollefeys, "Large scale visual geo-localization of images in mountainous terrain," in *Proc. of European Conference on Computer Vision*, 2012.

[17] I. Saleemi and M. Shah, "Multiframe many-many point correspondence for vehicle tracking in high density wide area aerial videos," *International Journal of Computer Vision*, vol. 104, pp. 198–219, 2013.

[18] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proc. of Conference on Computer Vision and Pattern Recognition*, 1997.

[19] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. of International Conference on Computer Vision*, 1999.

[20] J.-Y. Bouguet, "MATLAB calibration toolbox," http://www.vision.caltech.edu/bouguetj/calib_doc/, 2008.