Detection of UAVs on a collision course using optical flow

Juan José Cabrera
Institute for Engineering Research (I3E)
Miguel Hernández University
Elche, Spain
juan.cabreram@umh.es

Arturo Gil
Institute for Engineering Research (I3E)
Miguel Hernández University
Elche, Spain
arturo.gil@umh.es

Luis Payá
Institute for Engineering Research (I3E)
Miguel Hernández University
Elche, Spain
lpaya@umh.es

Antonio Santo
Valencian Graduate School and Research Network for Artificial Intelligence (ValgrAI)
Institute for Engineering Research (I3E)
Miguel Hernández University
Elche, Spain
a.santo@umh.es

Oscar Reinoso
Valencian Graduate School and Research Network for Artificial Intelligence (ValgrAI)
Institute for Engineering Research (I3E)
Miguel Hernández University
Elche, Spain
o.reinoso@umh.es

David Rodríguez
Abionica Solutions S.L.P.
Alicante, Spain
drodriguez@abionica.com

ABSTRACT

This paper presents a method to detect, track and predict a potential collision with UAVs using an aircraft equipped with a single camera. The method analyses the movement in the camera's image plane by means of sparse optical flow. In this way, the camera's own movement can be modelled and cancelled by estimating a homography matrix from a set of corresponding points. Once the movement caused by the camera is cancelled other moving objects can be isolated and the presence of other UAVs can be detected. Additionally, the method predicts potential collisions by examining the alignment between the position and velocity vectors of the UAV, which are estimated up to a scale factor. The proposed method is effective at detecting and predicting collisions with UAVs, regardless of their appearance, size, or movement, making it useful for applications related to airspace security.

KEYWORDS

optical flow, UAV detection, collision prediction

ACM Reference Format:

Juan José Cabrera, Arturo Gil, Luis Payá, Antonio Santo, Oscar Reinoso, and David Rodríguez. 2024. Detection of UAVs on a collision course using optical flow. In *2024 4th International Conference on Robotics and Control Engineering (RobCE 2024), June 27–29, 2024, Edinburgh, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3674746.3674795

RobCE 2024, June 27–29, 2024, Edinburgh, United Kingdom

ACM ISBN 979-8-4007-1678-2/24/06…$15.00
https://doi.org/10.1145/3674746.3674795

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs), commonly known as drones, have become increasingly popular in recent years, offering a wide range of applications in areas such as surveillance, rescue, defence and package delivery. However, the use of drones has also raised security concerns: as the number of applications involving drones increase, the risk of collision with manned aircraft is higher. As a result, it is of uttermost importance to have an in-flight collision avoidance system to prevent air accidents.

To address these issues, significant research has been conducted on UAV detection and tracking using various sensors such as LiDAR, radar, acoustic sensors and cameras. LiDAR sensors may not be effective in detecting small UAVs at long distances [1, 2], typically due to their low resolution along the vertical axis. Radar sensors in civil applications are subject to regulations regarding frequency bands and power levels in order to avoid interferences with other systems. Acoustic sensors can also be used to detect drones that do not emit radio signals. However, the effectiveness of acoustic sensors is limited by environmental factors such as wind and ambient noise [3], which prevents its usage in airborne applications.

Regarding the usage of cameras, estimating the position and velocity of the UAV in a 3D space using only the information present in the images is a challenging problem. In this sense, binocular cameras with a reduced baseline are not applicable to the problem, due to their lack of sensitivity at long distances. If the stereo camera is designed with a wider baseline, then difficulties may appear during the installation on small drones [4]. Therefore, the employment of monocular systems for the detection of objects on a collision trajectory is attractive, since they possess a low weight at a reduced cost. However, the absence of depth information poses an additional difficulty.

There is a wide variety of techniques that can be used to detect and track UAVs in images obtained by monocular cameras. In

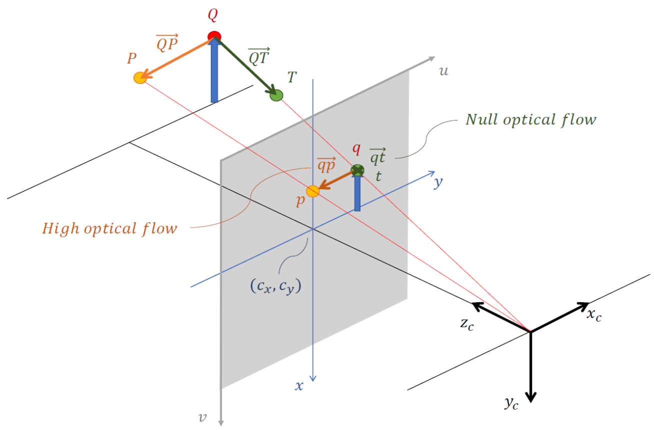

Figure 1: This figure shows the pinhole camera model in order to demonstrate that a movement in space does not always generate apparent motion on the image plane. In this case, $\vec{QT}$ does not generate apparent motion in the image $\vec{qt}$ and $\vec{QP}$ generate a high optical flow $\vec{qp}$ though.

this sense, this paper proposes an analytical method based on its apparent motion, whose main advantages are:

- It provides a generalized detection capability of flying objects, regardless of their appearance, shape, size and movement.
- The method computes an estimation of the object's direction in camera coordinates. This measurement is obtained up to a scale factor.
- The algorithm permits generating a prediction of a possible collision between the UAVs.
- The proposed method does not require a training phase.

The rest of the paper is structured as follows: Section 2 provides a review of the related literature. Next, Section 3 describes the proposed method in detail. Following, Section 4 presents the dataset used to test the algorithm in terms of detection and collision prediction. Finally, Section 5 draws the main conclusions and proposes future work.

2 STATE OF THE ART

Object detection can be approached through different methods depending on the complexity of the task to be solved. From the easiest to the hardest case, object detection could be classified as follows: (1) detecting a moving object from a static camera, (2) detecting a static object from a moving camera, and (3) detecting a moving object from a moving camera. This manuscript presents a solution that can be classified in this last group of applications.

Background Subtraction methods are generally applied to the case in which the camera is static and remove pixels whose intensity remains constant by computing the difference of consecutive images, thereby highlighting and detecting moving objects. However, these methods only work correctly when the background is relatively static [5], which is not applicable to the situation presented in this work.

Other methods are based on the calculation of the optical flow, defined as the apparent motion of pixels caused by the relative movement between the camera and the scene. When the optical flow is computed on every pixel in the image, it is referred to as dense optical flow [6–9]. In some situations, it may be interesting to compute the optical flow on a reduced set of characteristic points, thus obtaining a sparse optical flow [10, 11]. For example, [12] uses SURF descriptors to obtain image correspondences between consecutive frames in order to detect objects moving directly towards the camera (a case imperceptible to dense optical flow techniques). Their method is based on estimating the change in the size of the objects approaching the camera. To this end, the method analyses the increase in the scale associated to SURF descriptors, which permits computing a "time to collision" based on this information. Other authors employ similar techniques, using SURF descriptors and defining the object area through a convex hull. Thus, a collision is assumed to occur when an abrupt change in the object's area is detected in the image [13].

Methods based solely on optical flow have certain limitations, as a moving object does not always generate apparent motion in the image plane, as represented in Figure 1. Additionally, it is necessary to deal with the optical flow generated by the camera's own movement [14]. In this regard, optical flow can be used to model and cancel out the camera's movement, thereby detecting anything that moves differently (e.g., another UAV). The combination of a dense optical flow with a motion cancellation technique enables the detection of moving objects from a moving camera [15]. Moreover, sparse optical flow can also be combined with motion cancellation for the detection of moving objects, such as single-wing drones in flight sequences [16].

Recent research has focused on the use of deep learning algorithms. In this sense, YOLO's architectures are mainly used as backbone for UAV detection [17–19]. Other authors, have opted for NDFT (Nuisance Disentangled Feature Transform) networks for general object recognition from drone-captured images [20]. In addition, deep neural networks can not only be trained for object detection, but also for collision avoidance [21] or distance to object prediction [22]. Nevertheless, these techniques are limited by the number of UAV categories with which the neural network has been trained. Furthermore, aviation software that follows guidelines like DO-178C [23] (avionics software) is required to be "deterministic", meaning that it produces the same output given the same input. However, an AI-powered system may produce different results as it learns and adapts. In consequence, it becomes challenging to achieve the mandatory certification determinism required for aviation software when using AI. This fact justifies the development of analytical solutions to the proposed problem that should not depend on a set of training images.

In this manuscript, we propose a method that combines a sparse optical flow and a motion cancellation technique to compensate the own camera movement. The approach computes a homography matrix based on a sparse optical flow. Thus, the quality of the correspondences employed to generate the optical flow vectors will be crucial for the algorithm's performance.

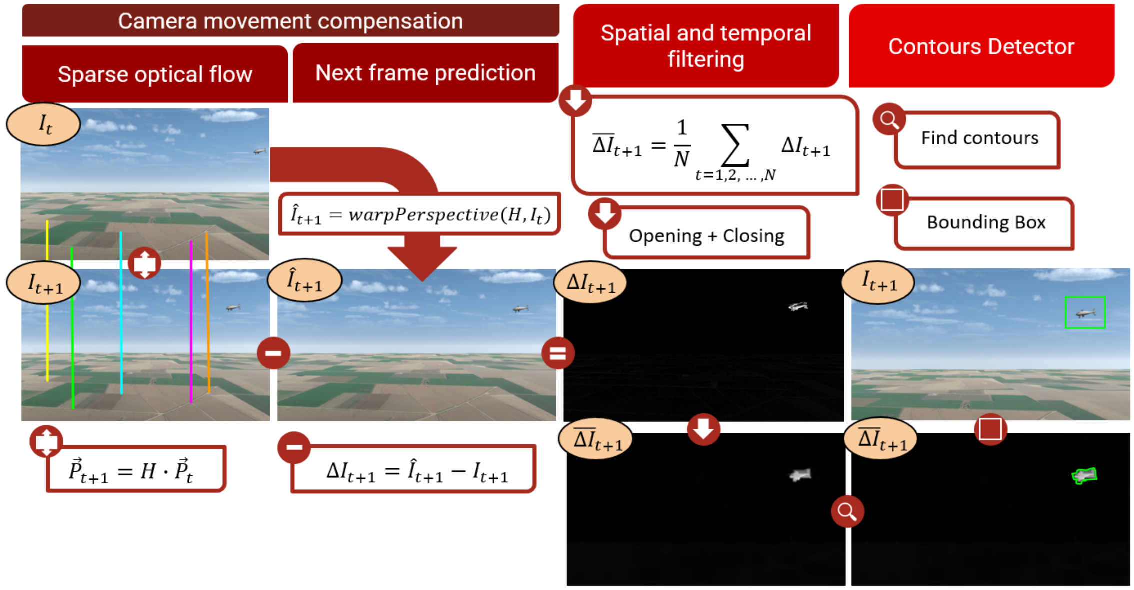

Figure 2: This diagram illustrates the main steps of the proposed method for UAV detection.

3 DESCRIPTION OF THE PROPOSED METHOD

In this manuscript, we propose a method that is able to detect aerial objects and also predicts the possible collision between the UAVs that share the same airspace in a given sequence. Section 3.1 describes the detection method and Section 3.2 describes the collision prediction method.

3.1 UAV detection

The proposed approach is illustrated in Figure 2 and consists of 5 main steps, which are defined next:

(1) Calculation of a sparse optical flow along consecutive frames.
(2) Modelling the movement of the pixels in the sequence using a homography matrix.
(3) Prediction of the next frame and compensation of the camera movement.
(4) Spatial and temporal filtering.
(5) Segmentation of closed contours and detection of the UAV.

First, given two consecutive frames I_t and I_{t+1}, a sparse optical flow is computed over a series of characteristic points obtained with the Shi-Tomasi algorithm [24]. Then, with these points, a sparse optical flow is calculated using the Kanade-Lucas method [25]. This provides a set of correspondences between the sets of points $\vec{p}_t$ (on image I_t) and $\vec{p}_{t+1}$ (on image I_{t+1}). Next, we model the scene's motion through a homography matrix H, since:

$$\vec{p}_{t+1} = H \cdot \vec{p}_t \tag{1}$$

This allows us to estimate H from a series of correspondences, assuming that the points are on a plane (in the presented application, we assume that there exists a plane in the scene). The knowledge of H allows us to predict the next image $\hat{I}_{t+1}$ using a perspective transformation to the frame I_t. It is important to note that the estimation of the homography matrix H is one of the most important elements of the proposed method, as it will condition the proper functioning of the algorithm. To carry out the calculation of this matrix, it is necessary to have, at least, 4 characteristic points belonging to a plane in the scene (in this case, the points belong to the ground plane). But as not all points detected by Shi-Tomasi method belong to that plane, a filtering is required to discard, for example, the points above the horizon (found at infinite distance). The RANSAC algorithm [26] is used for this filtering.

Once the movement of the scene has been modelled by means of a homography matrix and the prediction of the next frame has been made, our own movement is compensated by subtracting the background through the difference between the next frame and its prediction:

$$\Delta I_{t+1} = \hat{I}_{t+1} - I_{t+1} \tag{2}$$

Once the image of differences ΔI_{t+1} has been obtained, a spatial filtering is applied to it (composed of a Gaussian filter, a dilation and an erosion). These operations permit removing some regions that could result in false positives and enhance the object of interest.

Subsequently, a temporal filtering is introduced in which the mean of N difference images $\Delta \bar{I}_{t+1}$ is carried out in order to filter out those areas of the image where noise appears sporadically:

$$\Delta \bar{I}_{t+1} = \frac{1}{N} \sum_{t=0,1,\dots,N} \Delta I_{t+1} \tag{3}$$

Finally, the mean image of differences $\Delta \bar{I}_{t+1}$, resulting from applying the temporal filter, is processed by a contours detector with the objective of finding closed regions. As a result of the detection process, the position (u, v) of the biggest closed region in the image is obtained and, in addition, an estimation of the scale of the object is defined by (b_h, b_w) which represents the height and width of the window enclosing the contour.

3.2 Predicting collisions

In this section we introduce a method to predict the collision between the UAV that carries the camera and the flying object detected in the image (as described in Section 3.1). As we can imagine, the collision prediction problem is highly coupled with the detection phase. In order to address the collision prediction problem, it is necessary to know the position and velocity in a three-dimensional space. In our case, we consider that $\vec{P}_t$ and $\vec{V}_t$ are known up to a scale factor. In addition, we consider that the relative velocity $\vec{V}_t$ is constant during our tests. The collision prediction method is divided into the following steps:

(1) Computing the object's position in camera coordinates.
(2) Computing the relative velocity $\vec{V}_t$.
(3) Calculation of the relative angle.

First, we consider that the position (u, v) and size (b_w, b_w) of the detected object in the image is known (as described in Section 3.1). Next, the position of the object can be back-projected into a 3D space using the following equations:

$$z_h = f_x \frac{h}{b_h} \qquad z_w = f_y \frac{w}{b_w} \tag{4}$$

$$x_c = \frac{u - c_x}{f_x} z_c \qquad y_c = \frac{v - c_y}{f_y} z_c \qquad z_c = \frac{z_w + z_h}{2} \tag{5}$$

where h and w are the arbitrary height and width of the object in meters, b_h and b_w define the height and width of the detected bounding box. This allows us to obtain an estimate of the distance from the object to the camera z_c (up to a scale factor). In addition, u and v are the pixel coordinates where the object is detected and f_x, f_y, c_x, and c_y are the intrinsic parameters of the camera. Finally, the position of the object at time t in camera coordinates is expressed as $\vec{P}_t = (x_c, y_c, z_c)_t$. Once the position of the UAV $\vec{P}_t$ relative to the camera is known at each time instant, we compute its velocity $\vec{V}_t$ as:

$$\vec{V}_t = \vec{P}_t - \vec{P}_{t-1} \tag{6}$$

which constitutes the relative velocity between the UAV and the observing camera, i.e. the direction of movement of the UAV relative to the camera. Next, in order to determine whether the detected UAV is on a collision path, we compute the dot product:

$$\vec{V}_t \cdot \vec{P}_t = |\vec{V}_t||\vec{P}_t| \cos\theta \tag{7}$$

Whenever the UAV's velocity vector and its position vector are collinear (as described in Figure 3) a collision in air is possible. Therefore, this constitutes a first condition to detect that the UAV will collide with the camera:

$$\vec{V}_t \cdot \vec{P}_t = |\vec{V}_t||\vec{P}_t| \cos\theta = \cos\theta \approx -1 \tag{8}$$

since $\vec{V}_t$ and $\vec{P}_t$ have been normalised at this step, thus $\theta \approx \pi$, as represented in Figure 3. However, a significant amount of noise is expected in $\vec{P}_t$ and, thus, $\vec{V}_t$. As a result, as a first condition, we only check that $|\cos\theta| \approx 1$. Next, the second condition considers a filtered trend in the area of the object along several frames. In this way, when the object approaches, the bounding box should grow and vice versa. A first order linear function is approximated using the scale of the object at each frame t. As a result, the second condition to predict a collision considers a positive value in the slope of the function.

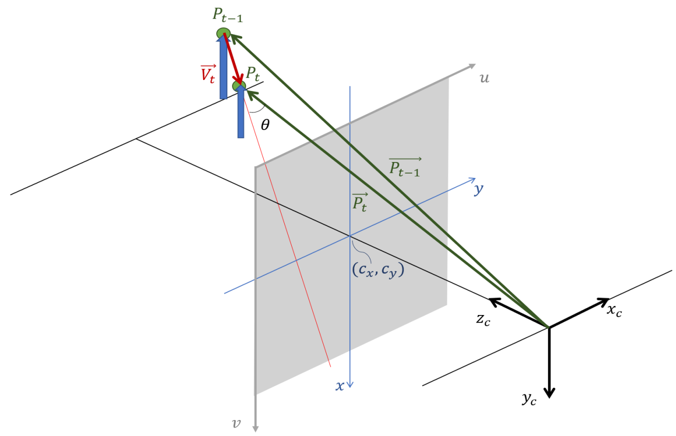

Figure 3: This figure shows the pinhole camera model in order to demonstrate that a collinearity between the velocity vector and the position vector gives information about a possible collision.

4 RESULTS

In this section, the dataset used to test the algorithm is described. Additionally, we assess the ability of the proposed method to detect and track aerial objects (Section 4.2), as well as predicting collisions with such UAVs (Section 4.3). A video description of the whole process can be found in this link[1].

4.1 Dataset

The dataset consists of a total of 40 flight sequences that have been simulated in Unity[2]. Each sequence has been specifically designed to replicate different situations that may occur under typical flight conditions. As a result, the dataset has been designed to test the robustness of the proposed algorithm. During each simulation, two drones are simulated, one of them carrying a standard pinhole camera with known intrinsic parameters. In 16 sequences both aircraft collide, whereas in the other 24 sequences, they do not. In each sequence, the drones have different relative speeds and appear at different orientations.

With the aim of testing the limitations of the current algorithm, 20 simulations were generated where the aircraft flew over crop plains at high altitude, and another 20 simulations were carried out in a mountain environment where the aircraft flew at low altitude. In this latter environment, the optical flow generated by the ground elements is higher and erratic, since it is not generated by a plane. In consequence, this environment is more challenging. Additionally, two types of UAVs were simulated, a Camcopter[3] helicopter (20 video sequences) and a single propeller Cessna[4] airplane (20 video sequences).

[1] https://youtu.be/Xg1GZ55vumw
[2] https://unity.com
[3] https://schiebel.net/products/camcopter-s-100/
[4] https://cessna.txtav.com

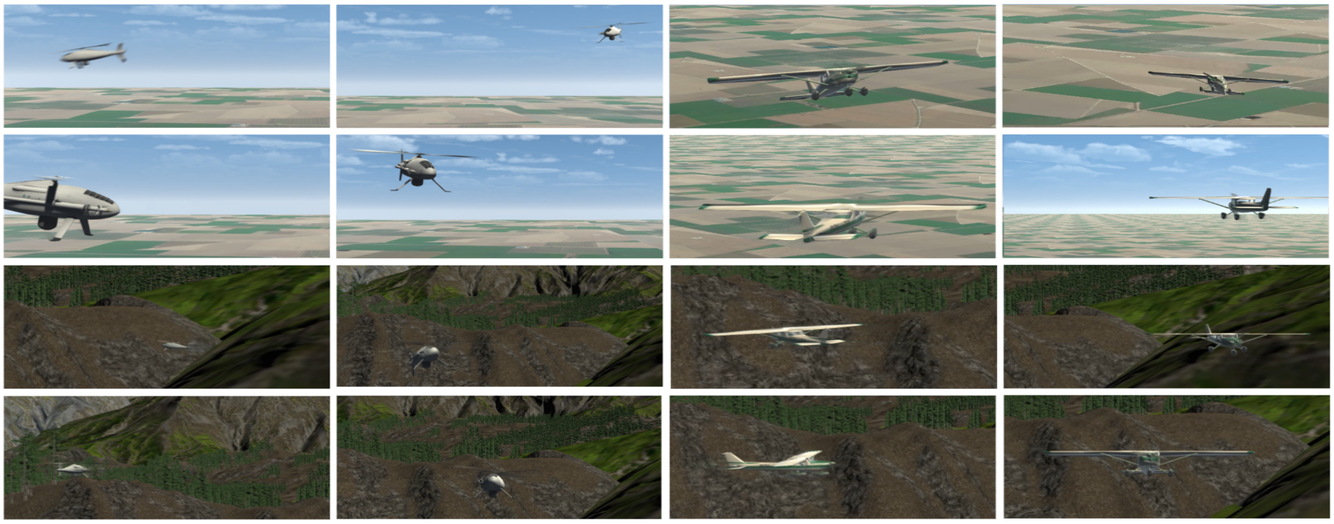

Figure 4: Sample frames extracted from different sequences of the Camcopter© and Cessna© UAV models in the two simulation environments (crop plains and rocky mountains).

Finally, the ground truth includes the position of the UAV in the 3D space and in the image plane. This allows us to evaluate the results in terms of detection and collision prediction. As an example, Figure 4 presents some snapshots extracted from different simulation sequences.

4.2 Results of the detection method

In this section, we evaluate the performance of the proposed algorithm to detect and track UAVs as described in Section 3.1. For this purpose, we calculated precision and recall for the 40 flight sequences that make up the dataset (Table 1). In order to obtain these values, the true positives, false positives, true negatives, and false negatives were considered for each of the frames that compose the different sequences. In this table, it can be observed that the results obtained for the crop plains environment were really satisfactory, generally obtaining high values in precision and recall. If we analyse the results obtained in the rocky mountains environment in detail, the number of false positives increases, thus decreasing the detection precision. It should be noted that, in this environment, false positives can be produced by the detection of trees and nearby rocky surfaces. The detection of these objects is considered, in some cases, correct, and may increase the values of the computed precision in some sequences. However, the modelling of the 3D terrain in simulation is complex and only the position of the approaching UAV is considered in the results. It is also noticeable, when observing Table 1 that recall values are also close to 1. Thus, we can conclude that our algorithm produces a very small number of false negatives (the absence of detection whenever an aircraft exists in the image), which are highly undesirable in a UAV detection system.

Table 1: Precision and recall results for each of the sequences that compose the dataset presented in Section 4.1.

UAV Detection					
UAV category		**Cessna**		**Camcopter**	
Environment	**Seq**	**Precision**	**Recall**	**Precision**	**Recall**
Crop plains	**1**	0.99	1.00	1.00	1.00
	2	0.99	1.00	1.00	1.00
	3	1.00	1.00	1.00	1.00
	4	1.00	1.00	1.00	0.77
	5	1.00	1.00	1.00	0.77
	6	1.00	0.99	1.00	1.00
	7	1.00	0.96	1.00	0.68
	8	1.00	1.00	1.00	0.92
	9	1.00	1.00	1.00	1.00
	10	1.00	1.00	1.00	1.00
Rocky mountains	**1**	0.78	1.00	0.82	1.00
	2	0.84	1.00	0.83	1.00
	3	0.76	1.00	0.76	1.00
	4	0.84	1.00	0.83	1.00
	5	0.87	1.00	0.84	1.00
	6	0.94	1.00	0.86	1.00
	7	0.93	1.00	0.90	1.00
	8	0.89	1.00	0.72	1.00
	9	0.27	1.00	0.65	1.00
	10	0.71	1.00	0.70	1.00

4.3 Results obtained in collision prediction

In this section, the proposed method for collision prediction (Section 3.2) is quantitatively evaluated. To this end, the ratio between the number of frames in which collision is predicted and the total number of frames in a sequence has been calculated. This factor has been named "global prediction ratio" R_G. In each sequence, this ratio should be compared with the expected result ("1": whenever collision occurs in the sequence or "0": if no collision occurs in the simulation). It is also noticeable that, in some sequences, the initial size of the object in the images is very small. This fact justifies the definition of a ratio that considers only the last moments before a

Table 2: Collision prediction results as a ratio, where values close to 0 indicate a prediction of "no collision" and values close to 1 indicate a collision prediction. This evaluation has been carried out taking into account either the whole sequences or only their last 8 seconds.

Prediction of collision						
UAV category		**Cessna**		**Camcopter**		**Ground truth (0 means no collision, 1 for collision)**
Environment	**Seq**	**Global prediction ratio R_G**	**Last 8 seconds prediction ratio R_8**	**Global prediction ratio R_G**	**Last 8 seconds prediction ratio R_8**	
Crop plains	**1**	0.28	0.11	0.24	0.06	0
	2	0.45	0.41	0.34	0.30	0
	3	0.80	0.81	0.96	0.96	1
	4	0.80	0.80	0.71	0.93	1
	5	0.88	0.89	0.68	0.88	1
	6	0.03	0.05	0.12	0.00	0
	7	0.12	0.08	0.09	0.00	0
	8	0.52	0.53	0.84	1.00	1
	9	0.00	0.00	0.37	0.37	0
	10	0.40	0.40	0.03	0.03	0
Rocky mountains	**1**	0.00	0.00	0.00	0.00	0
	2	0.45	0.42	0.02	0.02	0
	3	0.96	0.97	0.79	0.79	1
	4	0.55	0.55	0.70	0.70	1
	5	0.78	0.79	0.67	0.67	1
	6	0.13	0.07	0.10	0.15	0
	7	0.08	0.00	0.09	0.00	0
	8	0.44	0.51	0.51	0.59	1
	9	0.29	0.29	0.54	0.54	0
	10	0.49	0.49	0.09	0.09	0

possible collision. In consequence, we also define the "8 seconds prediction ratio", R_8, that is computed only on the last 8 seconds of each sequence. As an example, Figure 5 shows a sequence of frames where the aircraft carrying the camera collides with a Camcopter©.

Table 2 shows the collision prediction ratio for each of the sequences that compose the proposed dataset, evaluated either considering the entire sequence or just the last 8 seconds of the simulation. Additionally, to check the effectiveness of the algorithm, the ground truth of the collision ratio is presented in the last column of the table. The results presented in this table show, in general terms, that the collision prediction is satisfactory in all sequences. It should be noted that the algorithm proposed has a good performance for both environments. In addition, the algorithm presents better results when assessed using the R_8 ratio, compared to the R_G ratio, as expected.

5 CONCLUSIONS AND FUTURE WORK

A general technique to detect UAVs that share the same airspace has been presented. The method uses the information provided by a single camera carried by a UAV. In addition, the algorithm is able to indicate if the aircrafts are in a collision path and create an alert. The method is based on the computation of a sparse optical flow that permits computing a homography matrix. The homography matrix is used to compensate the movement of the camera, thus highlighting objects that move with a different velocity. Next, collision is predicted by analysing the collinearity of the position and velocity vectors.

The results obtained with the proposed detection and collision prediction method are highly satisfactory since, in the vast majority of cases, the UAV is detected and the potential collision is accurately predicted with enough time to avoid the aerial object. The method has been tested using video sequences generated from a simulation environment.

Regarding the UAV detection and tracking method, we have found that the technique is more prone to false positives than false negatives. This fact is considered an advantageous feature, since false negatives are extremely dangerous from the perspective of drone navigation safety standards. Moreover, the number of false positives has increased especially in mountainous environments, where the aircraft flies at low altitude. In this case, not only the UAV was detected, but also the nearest scene elements that could pose a danger in navigation.

When analysing the collision prediction method, we have demonstrated that the method is not able to estimate the actual depth at which the UAV is located. However, the method provides an accurate estimation of the object direction. Thus, it is possible to estimate reliably whether both UAVs are on a collision course or not. We have shown, however, that the method is not able to provide the time of the impact, since the distance at which the UAV is located cannot be estimated with a monocular camera in absence of a reliable scale information. Finally, the ability to predict a collision was restricted to 8 seconds before the event, which permits carrying out an evasion procedure. In this situation, the method showed high success ratio in all situations.

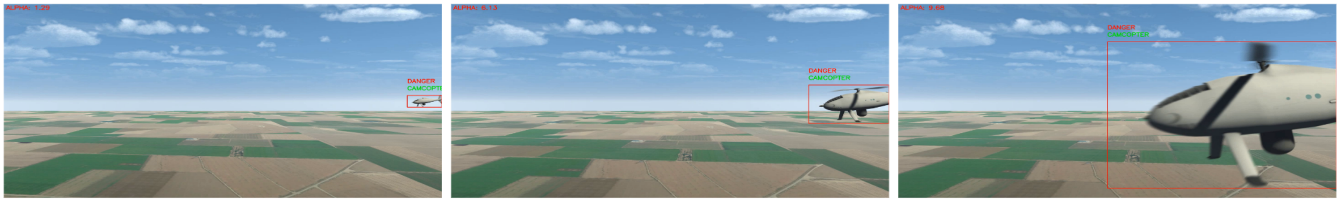

Figure 5: Example of a sequence in which the UAV equipped with the camera collides with a Camcopter. The bounding box coloured in red and the label "DANGER" indicates that the object is detected and predicted to be in collision path. Otherwise, the bounding box is green.

Regarding future work, the information obtained by the proposed method is of interest to integrate it into "sense and avoid" systems. To achieve this, a trajectory planning method should be incorporated in order to avoid the collision.

ACKNOWLEDGEMENTS

The Ministry of Science, Innovation and Universities (Spain) has supported this work through "Ayudas para la Formación de Profesorado Universitario" (FPU21/04969). This work has also been funded by the company Abionica Solutions S.L.P. under the contract: "Employment of algorithms for in-flight situational awareness using computer vision", ABIONICA1.21T. In addition, this publication is part of the project PID2020-116418RB-I00 funded by MCIN/AEI/10.13039/501100011033. The work is also part of the project TED2021-130901B-I00, funded by MCIN/AEI/10.13039/501100011033 and by the European Union "NextGenerationEU"/PRTR.

REFERENCES

[1] Yazhe Tang, Yuchao Hu, Jinqiang Cui, Fang Liao, Mingjie Lao, Feng Lin, and Rodney S. H. Teo. Vision-aided multi-uav autonomous flocking in gps-denied environment. *IEEE Transactions on Industrial Electronics*, 66(1):616–626, 2019. doi: 10.1109/TIE.2018.2824766.

[2] Csaba Benedek, Andras Majdik, Balazs Nagy, Zoltan Rozsa, and Tamas Sziranyi. Positioning and perception in lidar point clouds. *Digital Signal Processing*, 119: 103193, 2021. ISSN 1051-2004. doi: https://doi.org/10.1016/j.dsp.2021.103193. URL https://www.sciencedirect.com/science/article/pii/S1051200421002323.

[3] Frank Christnacher, Sébastien Hengy, Martin Laurenzis, Alexis Matwyschuk, Pierre Naz, Stéphane Schertzer, and Gwenaél Schmitt. Optical and acoustical uav detection. In *Security + Defence*, 2016.

[4] Jon Muhovič, Rok Mandeljc, Borja Bovcon, Matej Kristan, and Janez Perš. Obstacle tracking for unmanned surface vessels using 3-d point cloud. *IEEE Journal of Oceanic Engineering*, 45(3):786–798, 2020. doi: 10.1109/JOE.2019.2909507.

[5] M. Piccardi. Background subtraction techniques: a review. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, volume 4, pages 3099–3104 vol.4, 2004. doi: 10.1109/ICSMC.2004.1400815.

[6] Josef Maier and Martin Humenberger. Movement Detection Based on Dense Optical Flow for Unmanned Aerial Vehicles. *International Journal of Advanced Robotic Systems*, 10(2):146, February 2013. ISSN 1729-8814, 1729-8814. doi: 10.5772/52764. URL http://journals.sagepub.com/doi/10.5772/52764.

[7] Aslani Sepehr and Mahdavi Nasab Homayoun. Optical flow based moving object detection and tracking for traffic surveillance. In *International Conference on E-Business and E-Government*, 2014.

[8] Jiuchao Qian, Ling Pei, Danping Zou, and Peilin Liu. Optical flow-based gait modeling algorithm for pedestrian navigation using smartphone sensors. *IEEE Sensors Journal*, 15(12):6797–6804, 2015. doi: 10.1109/JSEN.2015.2464696.

[9] Anshuman Agarwal, Shivam Gupta, and Dushyant Kumar Singh. Review of optical flow technique for moving object detection. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pages 409–413, 2016. doi: 10.1109/IC3I.2016.7917999.

[10] Radu Timofte and Luc Van Gool. Sparse flow: Sparse matching for small to large displacement optical flow. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 1100–1106, 2015. doi: 10.1109/WACV.2015.151.

[11] Hosik Choi, Byungmun Kang, and DaeEun Kim. Moving object tracking based on sparse optical flow with moving window and target estimator. *Sensors*, 22(8), 2022. ISSN 1424-8220. doi: 10.3390/s22082878. URL https://www.mdpi.com/1424-8220/22/8/2878.

[12] Tomoyuki Mori and Sebastian Scherer. First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles. In *2013 IEEE International Conference on Robotics and Automation*, pages 1750–1757, 2013.

[13] Abdulla Al-Kaff, Fernando García, David Martín, Arturo De La Escalera, and José María Armingol. Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for uavs. *Sensors*, 17(5), 2017. ISSN 1424-8220.

[14] O. Deniz, G. Bueno, E. Bermejo, and R. Sukthankar. Fast and accurate global motion compensation. *Pattern Recognition*, 44(12):2887–2901, 2011. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2010.10.019. URL https://www.sciencedirect.com/science/article/pii/S0031320310005030.

[15] Laksono Kurnianggoro, Ajmal Shahbaz, and Kang-Hyun Jo. Dense optical flow in stabilized scenes for moving object detection from a moving camera. In *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pages 704–708, 2016.

[16] Jing Li, Dong Hye Ye, Timothy Chung, Mathias Kolsch, Juan Wachs, and Charles Bouman. Multi-target detection and tracking from a single camera in unmanned aerial vehicles (UAVs). In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4992–4997, 2016.

[17] Yuanyuan Hu, Xinjian Wu, Guangdi Zheng, and Xiaofei Liu. Object detection of UAV for Anti-UAV based on improved YOLO v3. In *2019 Chinese Control Conference (CCC)*, pages 8386–8390, 2019.

[18] Bailin Liu and Huan Luo. An improved yolov5 for multi-rotor uav detection. *Electronics*, 11(15), 2022. ISSN 2079-9292. doi: 10.3390/electronics11152330. URL https://www.mdpi.com/2079-9292/11/15/2330.

[19] Mahnoor Dil, Misha Urooj Khan, Muhammad Zeshan Alam, Farooq Alam Orakazi, Z Kaleem, and C Yuen. Safespace mfnet: precise and efficient multifeature drone detection network. *arXiv preprint arXiv:2211.16785*, pages 1–13, 2022.

[20] Zhenyu Wu, Karthik Suresh, Priya Narayanan, Hongyu Xu, Heesung Kwon, and Zhangyang Wang. Delving into robust object detection from unmanned aerial vehicles: A deep nuisance disentanglement approach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[21] Dário Pedro, André Mora, João Carvalho, Fábio Azevedo, and José Fonseca. Colanet: A uav collision avoidance dataset. In Luis M. Camarinha-Matos, Nastaran Farhadi, Fábio Lopes, and Helena Pereira, editors, *Technological Innovation for Life Improvement*, pages 53–62, Cham, 2020. Springer International Publishing. ISBN 978-3-030-45124-0.

[22] Brent A. Griffin and Jason J. Corso. Depth from camera motion and object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1397–1406, June 2021.

[23] Won Keun Youn, Seung Bum Hong, Kyung Ryoon Oh, and Oh Sung Ahn. Software certification of safety-critical avionic systems: Do-178c and its impacts. *IEEE Aerospace and Electronic Systems Magazine*, 30(4):4–13, 2015. doi: 10.1109/MAES.2014.140109.

[24] Jianbo Shi and Tomasi. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994. doi: 10.1109/CVPR.1994.323794.

[25] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981.

[26] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.