SIGS: Synthetic Imagery Generating Software for the Development and Evaluation of Vision-based Sense-And-Avoid Systems

Adrian Carrio · Changhong Fu · Jean-François Collumeau · Pascual Campoy

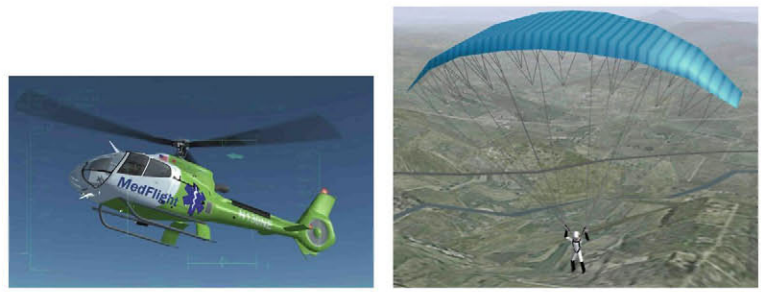
Abstract Unmanned Aerial Systems (UASs) have recently become a versatile platform for many civilian applications including inspection, surveillance and mapping. Sense-and-Avoid systems are essential for the autonomous safe operation of these systems in non-segregated airspaces. Vision-based Sense-and-Avoid systems are preferred to other alternatives as their price, physical dimensions and weight are more suitable for small and medium-sized UASs, but obtaining real flight imagery of potential collision scenarios is hard and dangerous, which complicates the development of Vision-based detection and tracking algorithms. For this purpose, user-friendly software for synthetic imagery generation has been developed, allowing to blend user-defined flight imagery of a simulated aircraft with real flight scenario images to produce realistic images with ground truth annotations. These are extremely useful for the development and benchmarking of Vision-based detection and tracking algorithms at a much lower cost and risk. An image processing algorithm has also been developed for automatic detection of the occlusions caused by certain parts of the UAV which carries the camera. The detected occlusions can later be used by our software to simulate the occlusions due to the UAV that would appear in a real flight with the same camera setup. Additionally this algorithm could be used to mask out pixels which do not contain relevant information of the scene for the visual detection, making the image search process more efficient. Finally an application example of the imagery obtained with our software for the benchmarking of a state-of-art visual tracker is presented.

Keywords Sense-And-Avoid · Collision avoidance · UAV

1 Introduction

The use of fixed-wing Unmanned Aerial Systems (UASs) for civil tasks is recently becoming more and more important. UASs can be useful for surveillance tasks, road traffic control, fire-fighting, meteorological observations, telecommunications, etc. Currently a lot of effort is being put in developing UASs that are able to coexist safely and effectively with current manned operations. UAS are expected to perform Sense and Avoid (SAA) functions at an "equivalent level of safety" (ELOS) to manned aircraft while not negatively impacting the existing infrastructure and

(a) C172P (b) Boeing 737-100



(c) Eurocopter EC130 (d) Paraglider

Fig. 1 Example of OBJ 3D models used to simulate intruder flying objects

manned Traffic Alert and Collision Avoidance System (TCAS) that ensure today's safe airspace [1, 2]. Sense-and-Avoid for UASs has been identified as one of the most significant challenges facing the integration of unmanned aircraft into the airspace [3]. Sensor information should be used to automatically detect possible aircraft conflicts ("sense") and those conflicts should be effectively avoided by taking the necessary automated control actions ("avoid").

There is a wide variety of sensing methods, which are usually divided into two categories: cooperative and non-cooperative approaches. Cooperative approaches are based in the mutual sharing of location information. An example of this approach are the Traffic-alert and Collision Avoidance System (TCAS) transponders [4]. In general cooperative approaches depend on the desire and ability of other aircraft to

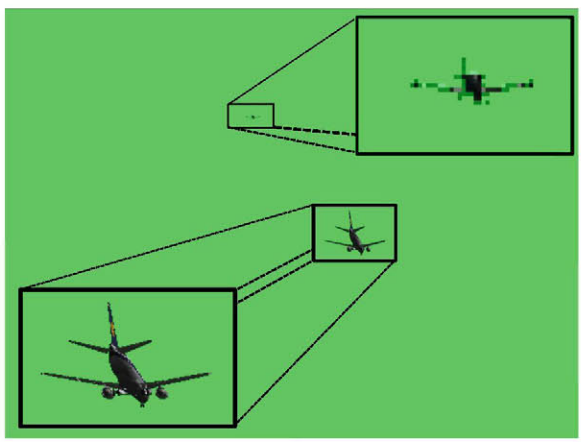


Fig. 2 Synthetic 1280x960 pixel image generated with two rendered 3D aircraft models over a chroma key green backdrop without any blending operations

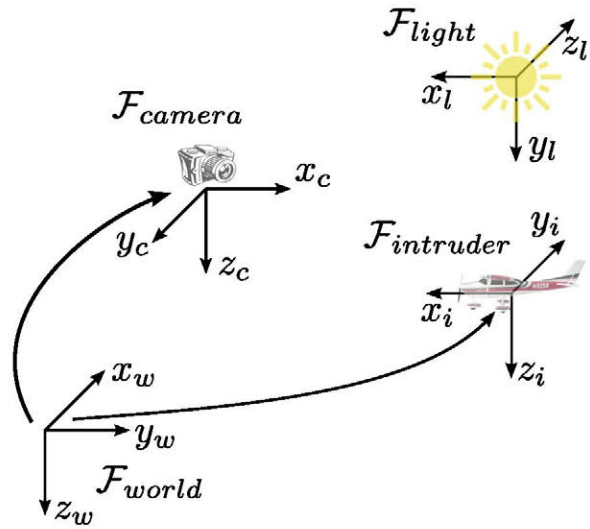


Fig. 3 Reference systems

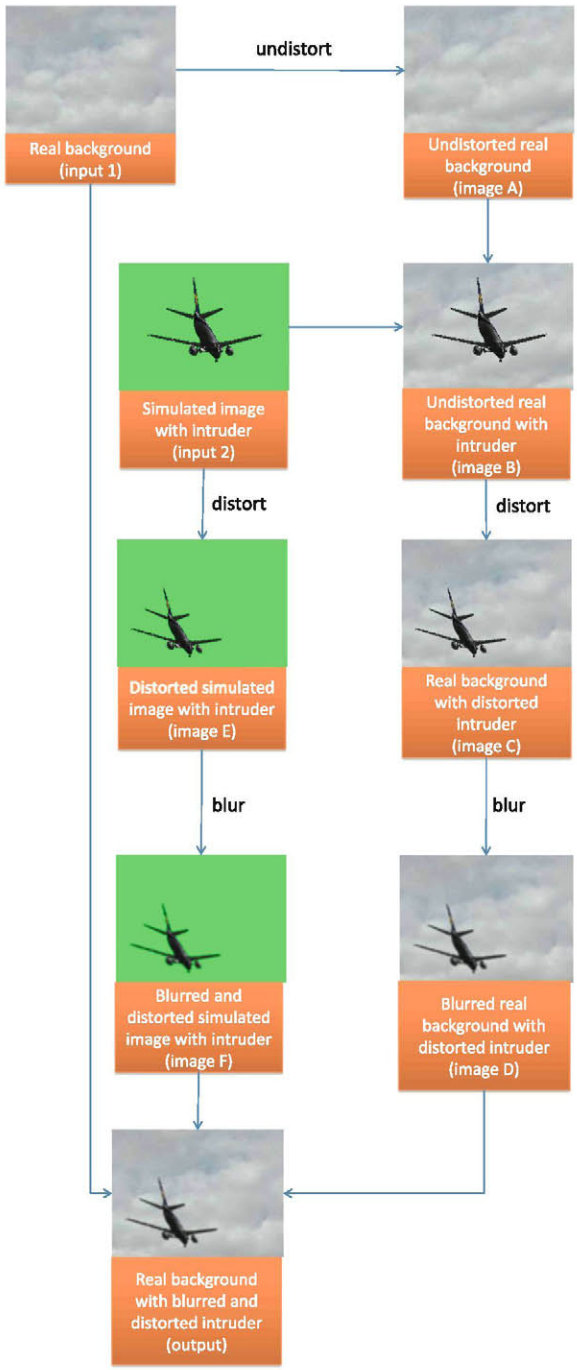


Fig. 4 Image compositing process showing results on the images

share information and they are not useful in collision situations with objects other than aircraft (e.g. flocks of birds). Moreover these systems cannot be used in

(a) Former method

(b) New method

Fig. 5 Comparison of the former and the presented image compositing algorithms. The incorporation of camera effects leads to more realistic results applied in a way that they only affect the simulated object and its boundaries

many cases because they are expensive, bulky and power-consuming (up to several hundreds of Watts).

Non-cooperative approaches are divided into two groups: active approaches and passive approaches. Active approaches are based in transmitting RF as part of the sensing. Radar is the main example of active approaches and whilst suitable for large aircraft, its use in medium-size and smaller platforms is limited due to price and to the size, weight and power restrictions onboard. In the passive approaches, promising results have been achieved by using acoustic sensors in order to identify surrounding noisy aircraft [5]. However these systems show some limitations due to the complexity of filtering the sounds generated by the UAV itself. Another passive approach with a lot of potential is passive radar, which uses the reflections of RF signals already existing in the environment to detect the presence of other flying objects [6]. Its main

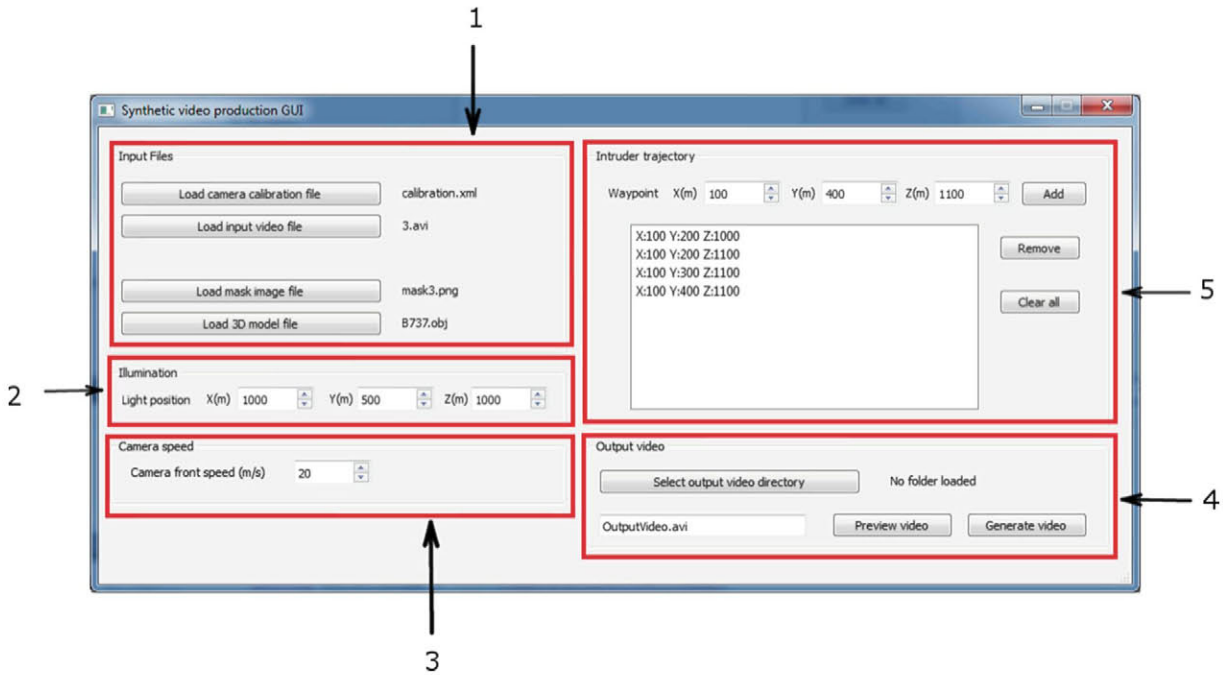


Fig. 6 Main window of the implemented graphical user interface. Block 1 refers to the input files needed for generating a synthetic video sequence: camera calibration, input video file, optional mask file and 3D model file. In block 2, the position of the omnidirectional light source point can be selected by the user. Block 3 lets the user select the front speed of the camera that captures the simulated 3D scene. Block 4 controls the preview and generation of output videos. Waypoints can be added to the trajectory of the simulated object in block 5

drawbacks are its complexity and the need to rely on third-party RF emitters.

Vision-Based approaches provide a passive, low-cost and more power-efficient solution. The availability of adequate image datasets is essential for developing and evaluating vision-based aircraft detection and tracking algorithms. However, this type of images is scarce and complicated to obtain.

Zarchan and Schneydor [7, 8] have reported that a mid-air collision between two aircraft travelling with constant speed occurs when those aircraft are converging with constant bearing. Under this collision course circumstance, aircraft appear to pilots as stationary features through the windscreen [9], so automated

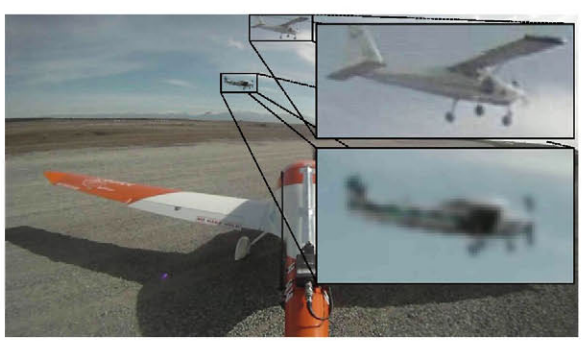


Fig. 7 Real (*above*) and simulated (*below*) intruders. Distance between camera and intruder is 58 m

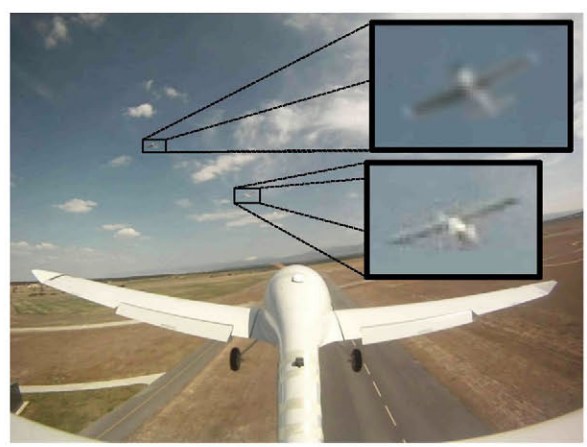


Fig. 8 Simulated (*above*) and real (*below*) intruders. Distance between camera and intruder is 156 m

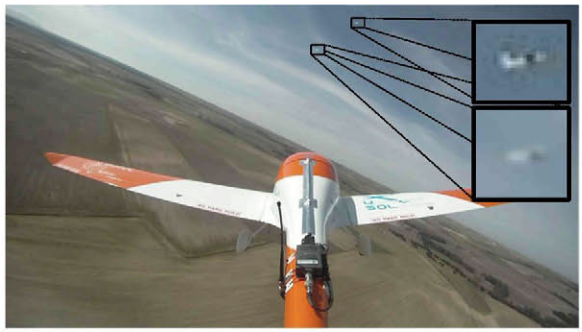


Fig. 9 Real (*above*) and simulated (*below*) intruders. Distance between camera and intruder is 333 m

Vision-based systems can exploit this property so that approached objects moving rapidly across the image plane can be rapidly discounted as genuine threats.

Therefore it is particularly interesting to use flight imagery in which collision-course encounters appear for the research and development of these systems, but obtaining this type of images is a complex and dangerous issue.

With the techniques presented in this paper, a simulated 3D aircraft is firstly positioned in a simulated 3D scene with controlled illumination and rendering parameters. After obtaining images from this scene, they are blended with images from real UASs flights using a novel image processing algorithm that takes into account the parameters of the real camera and applies them to the simulated object, obtaining realistic-looking results.

Since the intruder and camera poses are user-defined, the software provides ground truth annotations that allow to develop and quantitatively evaluate aircraft detection and tracking algorithms.

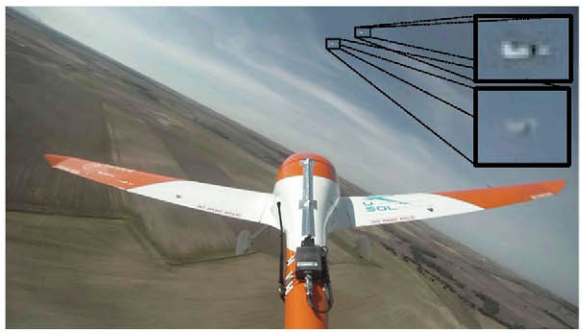


Fig. 10 Real (*above*) and simulated (*below*) intruders. Distance between camera and intruder is 334 m

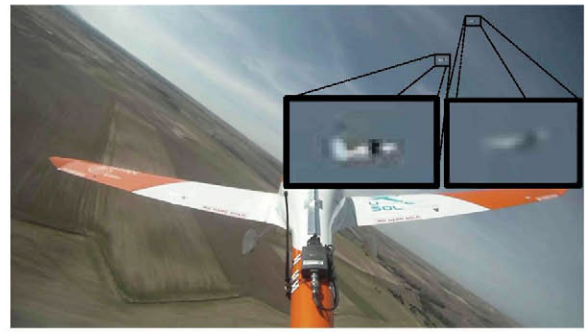


Fig. 11 Real (*above*) and simulated (*below*) intruders. Distance between camera and intruder is 391 m

This paper is organized as follows: in Section 2 the related works are presented. Section 3 introduces SIGS, our software for synthetic imagery generation. This software performs image processing to combine imagery captured from real UAV flights with simulated flight imagery. In Section 4 our algorithm for automatic occlusion detection is presented. Collected imagery is applied in order to evaluate different tracking algorithms in Section 5. Finally, Section 6 presents the conclusions and future works.

2 Related Works

Mian et al. [10] published a video database containing 23000 frames of flight sequences which were made available to the community free of cost. However, the usefulness of these images is limited since all videos were recorded from the ground.

Mejias et al. [11–13] and Dey et al. [14] describe different vision-based collision avoidance systems for which real flights were performed in order to collect suitable test data. To the author's knowledge, none of those databases have been made available for further development by the research community.

Currently some specific internet platforms can be found from which real aircraft imagery and videos can be downloaded such as AirTeamImages[1] or JetVideos.[2] However the imagery provided in these websites is not intended for research purposes and therefore many useful features are not available, e.g.

georeferenced or annotated image(s) or the possibility of selecting images by different criteria: type of background, aircraft trajectory, etc.

Recent research on the use of synthetic images for analyzing and demonstrating the functionality of vision-based systems has been performed by Forlenza et al. and Fasano et al. [15, 16], who developed a laboratory test system for Simulating a Sense and Avoid Flight System, using both real and synthetic imagery. Zsedrovits et al. [17, 18], used the flight simulation software FlightGear[3] to obtain synthetic flight imagery for which the intruder's size and 3D positions were known.

In order to improve the reality of this fully simulated approach, we have developed specific software to generate synthetic images in which the background has been replaced with real-world backgrounds recorded from UAVs, taking into account visualization parameters, such as illumination, camera lens distortions and out of focus blur. To the best of our knowledge, the development and use of this type of imagery for designing and evaluating vision-based Sense-and-Avoid systems is completely novel.

3 SIGS Software Architecture

The Synthetic Imagery Generating Software is a cross-platform graphical user interface based in Qt and written in C++. A screenshot of the user interface is shown in Fig. 6. This software takes a video file containing airborne footage, and for every frame automatically inserts a simulated aircraft in the image, according to various parameters provided by the user. This is done by blending real-world images with synthetic images which contain the simulated aircraft, using the image processing algorithm explained further on.

All the blended frames are then placed in a video sequence that realistically simulates an aircraft encounter situation.

3.1 Synthetic Images

Synthetic images were generated using GLC Lib,[4] an OpenGl-based rendering library. This library allows

[3] http://www.flightgear.org

[4] http://www.glc-lib.net/

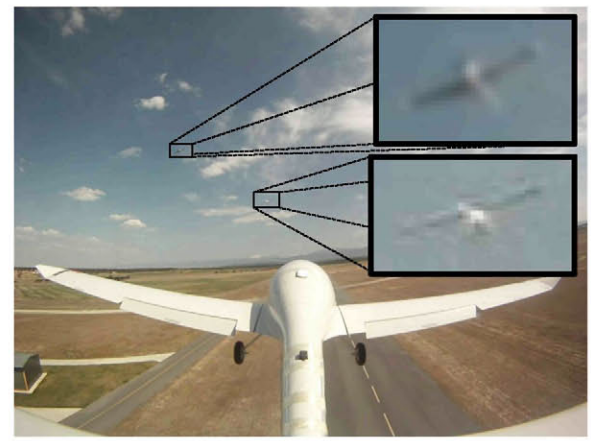


Fig. 12 Real (*above*) and simulated (*below*) intruders. Distance between camera and intruder is 175 m

to render a 3D scene with OBJ format 3D models. FlightGear offers many GPL licensed 3D models for different types of aircraft. Some examples are shown in Fig. 1.

The 3D scene is rendered with a chroma key green backdrop so that the pixels corresponding to the simulated aircraft can be easily segmented during the image processing stage. Figure 2 illustrates this.

The use of optimized rendering options is recommended for obtaining the best possible results when rendering the 3D model. In particular it is suggested the use of:

- Averaged normals for a smooth appearance
- Antialiasing to minimize the "ladder" effect
- Anisotropic filtering to improve texture sharpness

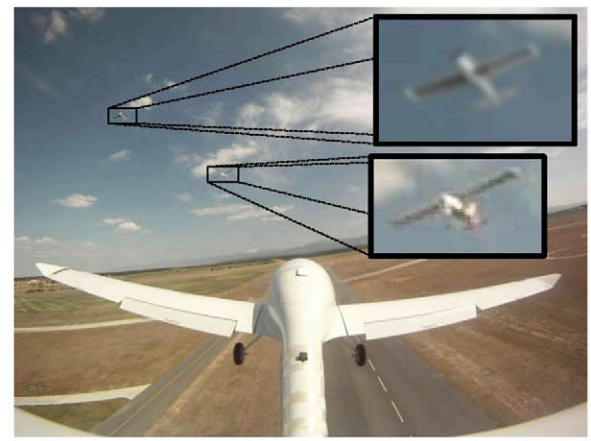


Fig. 13 Real (*above*) and simulated (*below*) intruders. Distance between camera and intruder is 135 m

3.1.1 Reference Systems

We have defined four reference systems as shown in Fig. 3. The F_{world} coordinate system is a global coordinate system used as common reference for all objects in the scene. The F_{camera} coordinate system indicates the pose of the camera that captures the synthetic image, with its optical axis pointing in the X direction. The $F_{intruder}$ coordinate system indicates the pose of the simulated aircraft and F_{light} is used to indicate the position of the omnidirectional light point that illuminates the scene.

3.1.2 Illumination

For illuminating the scene, an OpenGL point light source positionned at a 3D location has been used. This light source is omnidirectional and has white color by default.

The user should set the position of the illumination source according to the position of the sun or other light sources in the input video to get realistic illumination results.

3.1.3 Camera Speed

Since potential collisions between two aircraft travelling at constant speed will happen during stages of flight where bearing is constant [7, 8], when simulating camera movements for this type of imagery, keeping the camera direction constant should provide enough information for simulating the camera movements. Therefore, the implemented software does not depend on having geolocalized images in order to realistically simulate the movements of the intruder in the image as seen from a moving camera.

The initial position camera for the rendered 3D scene is defined at [0,0,-150] in world coordinates (altitude of 150m) and with the orientation shown in Fig. 3. The camera is assumed to have a constant front speed value (X axis, according to our convention), which should be provided by the user.

3.1.4 Trajectory Generation

In order to have smooth and realistic flight trajectories of the simulated intruder, they are generated using the basis spline model, a common method for aircraft trajectory modelling [19]. In particular we have implemented uniform splines of degree three. For this, a node vector $\mathbf{T} = \{t_0, t_1, ..., t_n\}$ is considered with $t_0 \leq t_1 \leq, ..., \leq t_n$ and n points $\mathbf{P}_i$.

When using $n+1$ control points, the interpolated points are given by:

$$\mathbf{X}_3(t) = \sum_{i=0}^{n} B_{i,3}(t).\mathbf{P}_i, 3 \leq t \leq n+1 \quad (1)$$

where $B_{i,3}(t) = 0$ if $t \leq t_i$ or $t \geq t_{i+4}$.

$$B_{i-2,3}(t) = \frac{1}{h}\begin{cases} (t-t_{i-2})^3 \text{ if } t \in [t_{i-2}, t_{i-1}] \\ h^3 + 3h^2(t-t_{i-1}) + 3h(t-t_{i-1})^2 - 3(t-t_{i-1})^3 \text{ if } t \in [t_{i-1}, t_i] \\ h^3 + 3h^2(t_{i+1}-t) + 3h(t_{i+1}-t)^2 - 3(t_{i+1}-t)^3 \text{ if } t \in [t_i, t_{i+1}] \\ (t_{i+2}-t)^3 \text{ if } t \in [t_{i+1}, t_{i+2}] \\ 0 \text{ otherwise} \end{cases} \quad (2)$$

where h is the distance between two consecutive points.

In order to generate a trajectory, the user should define a number of waypoints for the simulated aircraft to follow during the video sequence. The generated trajectory will fit these waypoints using the proposed model. The number of frames in the video sequence will determine the number of trajectory points sampled from the generated trajectory. The aircraft orientation is also computed for each trajectory point so that the aircraft follows it in a realistic way.

3.2 Image Processing Algorithm

The image processing algorithm makes possible to blend real images with synthetic ones, simulating the effects of lens radial and tangential distorsions and out of focus blur in the simulated object and adding it to the real image while smoothly interpolating the

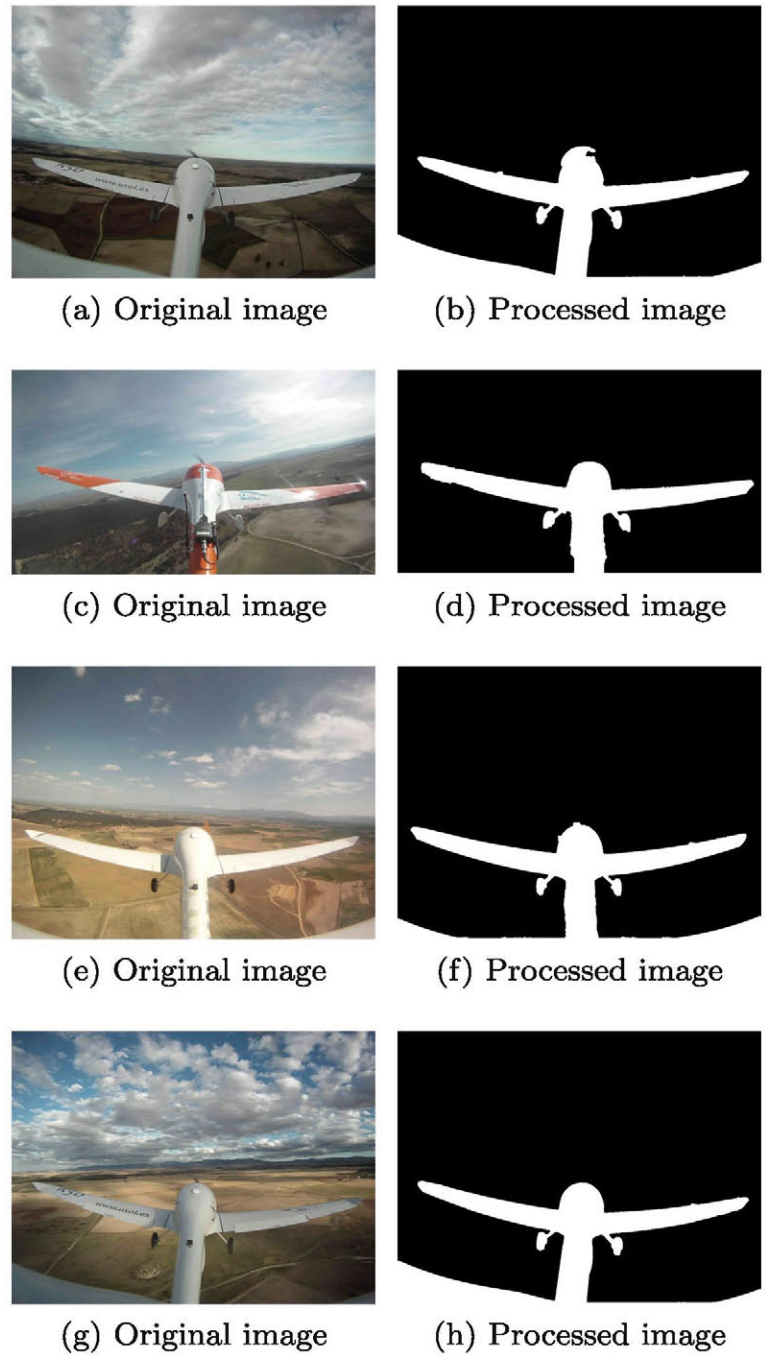


Fig. 14 Results of the automatic UAV occlusion detection algorithm

edges of the simulated aircraft. Brown's lens distortion model has been used [20, 21].

The image processing algorithm implemented has undergone several improvements with respect to the algorithm published in [22] in order to give a more realistic feel to the generated images. The OpenCV[5] library has been used for implementing the image processing stages.

[5] http://www.opencv.org

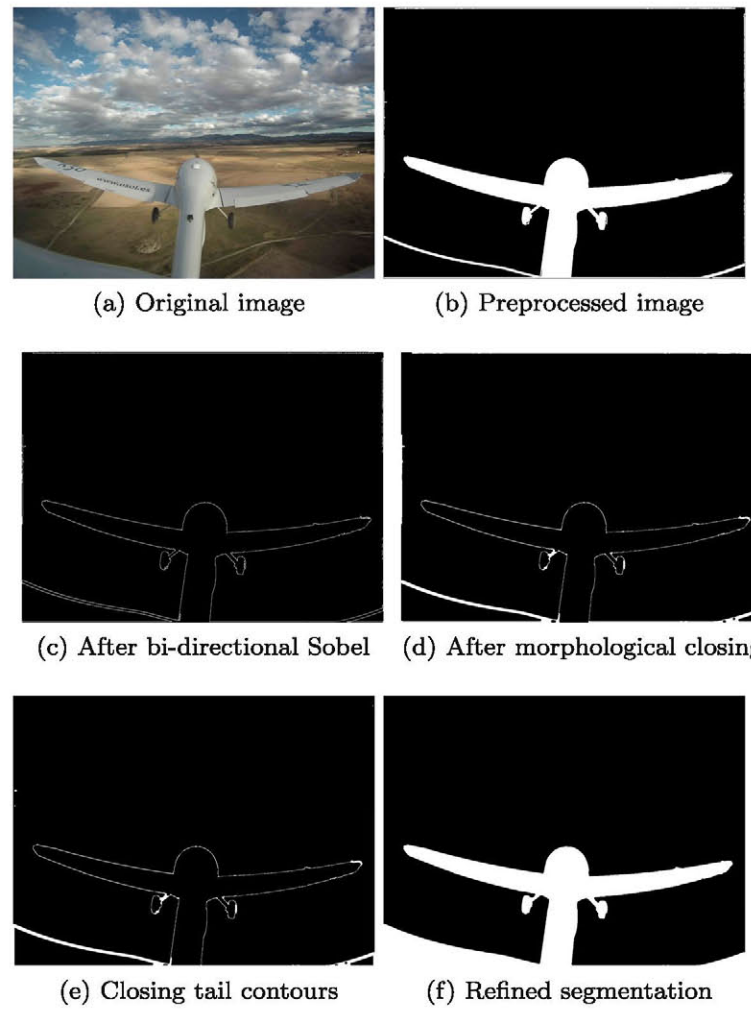
Fig. 15 Sum of normalized differences between succesive frames. The edges of the objects on the UAV causing occlusions are enhanced

The image compositing process is depicted in Fig. 4. Firstly the background image (input 1) is undistorted to obtain image A. According to the distortion model used, normalized coordinates $[x', y']$ of a generic point $[x, y]$ in the undistorted image are given by:

$$x' = x\frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} + 2p_1x'y' + p_2(r^2+2x'^2)$$
$$y' = y\frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} + p_1(r^2+2y'^2) + 2p_2x'y' \quad (3)$$

Where $k_1, k_2, ..., k_6$ are the radial distorsion coefficients, p_1, p_2 are the tangential distorsion coefficients and r is the radial location of pixel $[x, y]$. The new

Fig. 16 UAV mask refinement process



(a) Original image (b) Preprocessed image

(c) After bi-directional Sobel (d) After morphological closing

(e) Closing tail contours (f) Refined segmentation

(a) Original images

(b) Images after rough segmenetation

(c) Images after refined segmentation

Fig. 17 Examples of automatically produced UAV masks

pixel locations after the undistortion process are given by $[u, v]$:

$$\begin{aligned} u &= x' f_x + c_x \\ v &= y' f_y + c_y \end{aligned} \tag{4}$$

Where $[f_x, f_y]$ are focal lengths expressed in pixels and $[c_x, c_y]$ are the coordinates of the optical axis.

Pixels corresponding to the simulated intruder (input 2 in Fig. 4) are segmented and copied into image A to obtain image B. This image contains an undistorted background and an unaltered simulated intruder.

Then, image B is distorted using an inverse mapping procedure to the one used for undistortion and the result is stored in image C. This image is the blurred using a gaussian kernel to produce image D. The standard deviations for the gaussian blur are set proportionally to the distance between the simulated intruder and the camera, to increment the defocused effect on the simulated objects as they get farther away from the camera.

Because the whole image D has been blurred, but the blurring effect should affect only the simulated object and also because the undistortion process followed by distorsion may modify background pixels (some pixel intensities might be modified in the interpolation process), we perform a similar distortion and blur process on input 2 image, generating image F. This image allows us to identify all pixels corresponding to the simulated object and its edges after the distortion and blur process.

In image F, all pixels that have suffered some modification during the distortion and blur process are no longer green, so this image can be used as a mask to obtain the final output image. Pixels which are pure green have not been modified and therefore are copied directly from input 1 into the final image. Modified pixels, that include the simulated aircraft and the interpolated pixels around the edges are copied from image D to obtain the final output image.

Table 1 Video dataset

Filename	Intruder aircraft	No. frames	Trajectory	Main flight challenges
1LRB737.avi	Boeing 737	322	Left to right	Heavy clouds
1FNB737.avi	Boeing 737	389	Approaching	Heavy clouds
1NFB737.avi	Boeing 737	636	Moving away	Heavy clouds
1RLB737.avi	Boeing 737	275	Right to left	Heavy clouds
2LRB737.avi	Boeing 737	337	Left to right	Clear sky
2FNB737.avi	Boeing 737	318	Approaching	Clear sky
2NFB737.avi	Boeing 737	1143	Moving away	Clear sky
2RLB737.avi	Boeing 737	363	Right to left	Clear sky
3LRB737.avi	Boeing 737	353	Left to right	Vibrations
3FNB737.avi	Boeing 737	1033	Approaching	Vibrations
3NFB737.avi	Boeing 737	1033	Moving away	Vibrations
3RLB737.avi	Boeing 737	1033	Right to left	Vibrations
1LRC172.avi	Cessna C172	321	Left to right	Heavy clouds
1FNC172.avi	Cessna C172	411	Approaching	Heavy clouds
1NFC172.avi	Cessna C172	636	Moving away	Heavy clouds
1RLC172.avi	Cessna C172	274	Right to left	Heavy clouds
2LRC172.avi	Cessna C172	342	Left to right	Clear sky
2FNC172.avi	Cessna C172	287	Approaching	Clear sky
2NFC172.avi	Cessna C172	1143	Moving away	Clear sky
2RLC172.avi	Cessna C172	347	Right to left	Clear sky
3LRC172.avi	Cessna C172	337	Left to right	Vibrations
3FNC172.avi	Cessna C172	1033	Approaching	Vibrations
3NFC172.avi	Cessna C172	1033	Moving away	Vibrations
3RLC172.avi	Cessna C172	1033	Right to left	Vibrations

3.2.1 Comparison with Previous Algorithm

The incorporation of lens distortion effects and also gaussian blur to simulate the object being out of focus creates a more realistic effect. A comparison of the results using our former method and the current algorithm is shown in Fig. 5.

3.3 Graphical User Interface

A user-friendly interface has been designed in order to facilitate the generation of videos. The user interface, shown in Fig. 6, consists of five sections: input files, illumination, camera speed, intruder trajectory and output video.

In the input files section the user can select his own camera calibration file, the input video, a mask file for occluding the intruder and the 3D model file used for the intruder.

Regarding file formats, the camera calibration file should be an XML file in the standard format for calibration files used by OpenCV. AVI format is used as input video format. We have sticked to this format for its cross-platform support. The mask can be in any common image format (png, jpeg, bmp, ...). Regarding the 3D model, OBJ format is used.

3.4 Comparison with Real Imagery

In order to evaluate the similarity of the synthetic images generated with real images, a simulated intruder has been placed next to a real intruder in various video sequences using the developed software in order to compare their appareance. The results are shown in Figs. 7, 8, 9, 10, 11, 12 and 13.

Fig. 18 Real flight video captures used as background. The videos were recorded during 2011 in Marugán (Spain) from an USol K50. The images show different cloud conditions

In these sequences, a light aircraft appears in the field of view of the UAV flying different trajectories, which we have tried to imitate while keeping both real and simulated aircraft visible.

3.5 Video Annotations

Complete video annotations are included in XML format together with the video sequences, including for every frame:

- Intruder 3D coordinates with respect to the camera in meters
- Euclidean distance between camera and intruder in meters
- Intruder bounding box in the image (top-left pixel coordinates, width and height in pixels)

The camera and intruder 3D positions follow the convention for reference systems described in Fig. 3.

4 Automatic UAV Occlusion Detection

An algorithm has been implemented to automatically detect the occlusions in the image caused by the UAV body. This typically happens in fixed-wing UAVs when the camera is mounted in the tail, as in Fig. 14a, c, e and g. The detected occlusions can later be used by our software to simulate the occlusions caused by the UAV that would appear in a real flight with the same camera setup. Additionally this algorithm can be used in Vision-based See-and-Avoid systems to mask out pixels which do not contain relevant information for the visual detection or tracking tasks, making the image search process more efficient.

The image processing happens in two stages. The first one computes intra-frame differences and sums them over time in order to obtain the edges of the objects causing the occlusion. Finally during this stage a first attempt to close the found contours is performed.

Normally the segmentation of the objects on the UAV causing the occlusing is not complete for different reasons. For example, there might be holes or areas next to the image borders that do not get completely segmented. The post processing stage completes the process by filling the contours correctly.

This algorithm has been successfully tested in 4 different video sequences, recorded with different camera setups in different UAVs and with different illumination conditions.

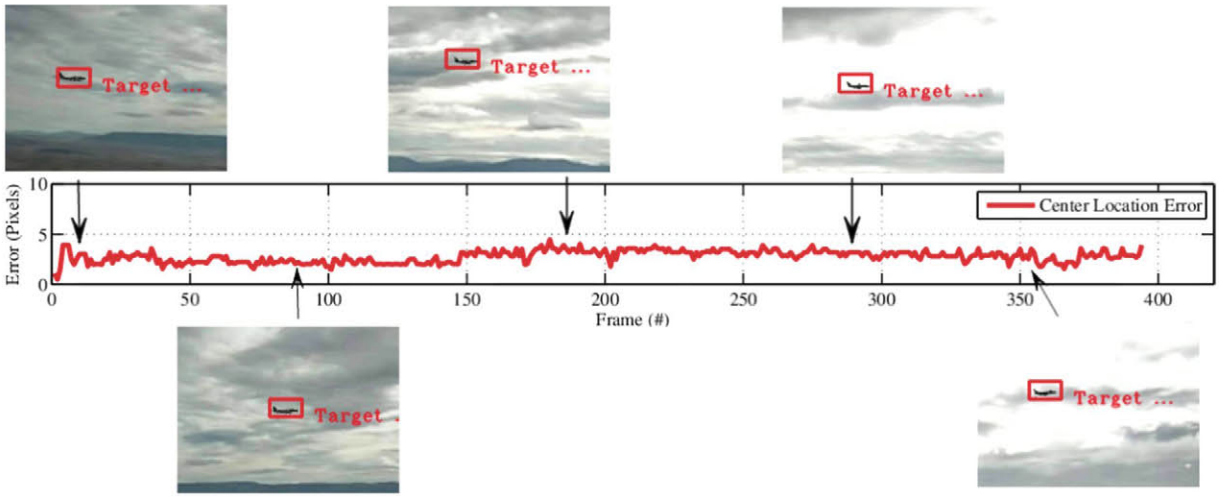


Fig. 19 Center Location Error with 1LRB737 video. We show the obtained results on five different frames extracted from this video. Mean CLE is 3 pixels in this scenario

4.1 Initial Processing Stage

During the first stage, the difference image between succesive frames is computed and normalized. Then the sum of the normalized difference images is computed for a large number of frames (i.e. 1500 frames, equivalent to a minute of footage at 25 fps).

The image obtained, shown in Fig. 15 basically contains the edges of the objects on the UAV causing the occlusion.

Once the normalized sum image is obtained, a Canny edge detector is run followed by a dilation operation. Finally a search for external contours is performed and the contours found are filled to obtain the output pre-processed images.

4.2 Post Processing Stage

As previously mentioned, the pre-processed images are incomplete masks of the UAV, as holes are present within the UAV shape. These images need to be

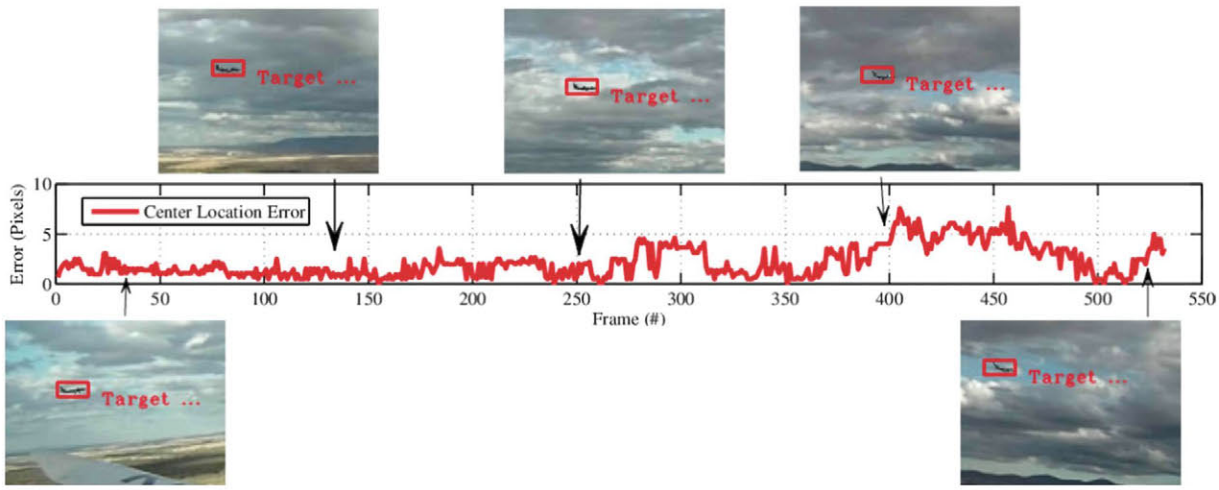


Fig. 20 Center Location Error with 3LRB737 video. Here the mean CLE is 3 pixels until frame 400 and then rises to 8 pixels due to strong vibrations in the UAV

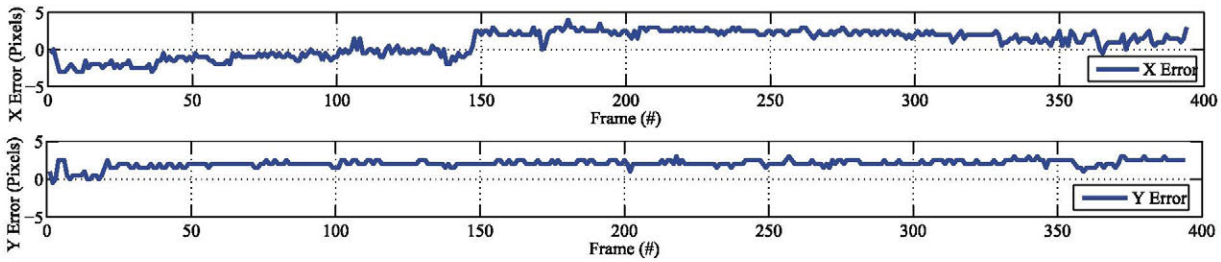


Fig. 21 Tracking errors in the 1LRB737.avi video file, computed as the difference between the coordinates of the center of the tracker's bounding box and the GT coordinates

refined in order to separate the contours of the UAV from those belonging to the background and, ultimately, obtain a more complete mask of the UAV.

The first step of this separation process consists of running a Sobel filter following both horizontal and vertical directions in order to retain only the strong edges present within the pre-processed images. The obtained contours then undergo a morphological closing operation in order to close potential holes in the obtained contours. The results of the Sobel filter and the morphological operations are illustrated in Fig. 16c and d, respectively.

The images obtained through this first step show closed contours suitable for being filled, with the exception of the tail part of the UAV. This part is in contact with the border of the image and is hence not closed. In order to overcome this problem, we select the rightmost and leftmost highest non-zero pixels and pad the sides of the image with white pixels from the bottom up to the identified pixels. The whole bottom of the image is padded in a similar manner, yielding images similar to the one shown in Fig. 16e.

The next step consists in filling the obtained closed contours. Contours which are smaller than an experimentally fixed threshold (0.5 % of the image area) are filtered out as part of the background and not belonging to the UAV. The final result of the mask refinement step can be seen in Fig. 16f. Figure 17 shows additional examples of the masks (both rough and refined) obtained by this process.

5 Results

A sample video dataset has been generated for the development and evaluation of Vision-based Sense-and-Avoid systems. The dataset consists of more than 14000 frames of flight imagery with simulated intruders and will shortly be made freely available to the research community. We have used some of these videos to show the potential of the AM^3 tracker.

5.1 Video Dataset

The video dataset currently consists of a set of 24 videos listed in Table 1. These videos offer a variety of trajectories, illumination and cloud conditions which are very challenging for detecting or tracking aircrafts, as shown in Fig. 18.

The background video imagery of real flights was recorded from an USol K50 UAV. It was provided by Unmanned Solutions (USol[6]). These videos have been recorded with a GoPro Hero 2 HD camera located in the UAV tail. The camera uses 170° fish-eye lenses, capturing frames with a 1280×960 pixels resolution.

5.2 Sample Tracker Evaluation

This section shows the evaluation results of the AM^3 visual tracker [23] on two different video files: 1LRB737.avi and 3LRB737.avi, which include challenging situations, e.g. rapid pose variation, background clutters, illumination changes, onboard mechanical vibration and wind influence. The performance of this tracker has been evaluated with the Ground Truth (GT) generated by our SIGS software on the mentioned video files. The errors in X, Y (differences in each coordinate between the tracker's bounding box center position and the intruder's GT position) and the standard Center Location Error (CLE) metric have been used for tracker evaluation. CLE is defined as the Euclidean distance from the

[6] http://www.usol.es

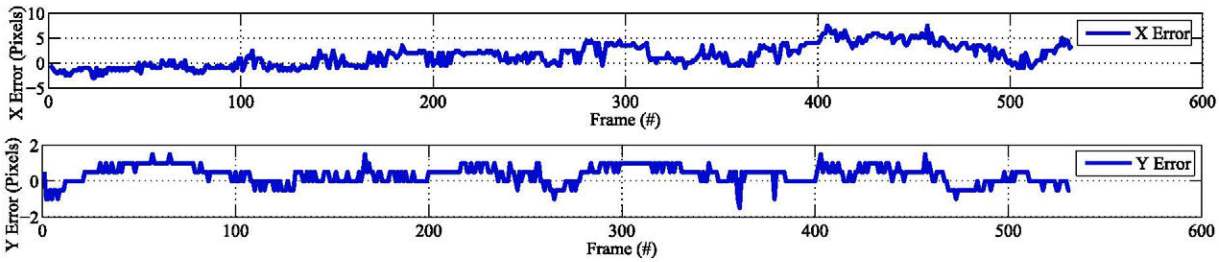


Fig. 22 Tracking errors in the 3LRB737.avi video file, computed as the difference between the coordinates of the center of the tracker's bounding box and the GT coordinates

center of the tracker's bounding box to the intruder aircraft GT position for each frame. By using these videos it is possible to tune the AM^3 tracker parameters by testing different parameter sets in order to improve the tracker's performance in various challenging conditions.

CLE results of the AM^3 tracker with the mentioned videos are shown in Figs. 19 and 20. The errors on each of the coordinates are shown as well in Figs. 21 and 22. The CLE errors stay below 5 pixels for 1LRB737.avi and for most part of 3LRB737.avi, only increasing to 8 pixels due to strong vibrations in the UAV. This proves that the tracker is robust to vibrations. Furthermore, Fig. 22 shows that image vibrations are affecting the tracker's performance mainly in the X-axis direction.

6 Conclusions and Future Works

A user-friendly graphical interface software has been developed for obtaining realistic synthetic images of simulated intruder aircraft in real flight imagery. These images are created by combining a 3D rendered intruder aircraft image with real flight images recorded from a UAV, taking into account illumination, camera lens distortions and the out of focus effect with realistic results.

An algorithm for automatic detection of the occlusions caused by the UAV body has been implemented. The segmentation of these occlusions can be used for simulating the occlusion effect in the generated imagery and also in other detection algorithms for masking out pixels which do not contain relevant information of the scene for the visual detection, making the image search process more efficient.

The images obtained with this software can be used as a powerful benchmark for the development and evaluation of vision-based Sense-and-Avoid systems, therefore representing an important improvement in the state of the art for Sense-and-Avoid. The results section shows how these images can be used to evaluate a state-of-art visual tracking algorithm, as an example application.

Regarding future works, we plan on making the software open-source and accessible to the research community in the short-term. New features will also be added such as multi-object simulation and support for geo-referenced images.

Acknowledgments The authors would like to thank the Spanish Ministry for Education, Culture and Sports for funding the international research stay of one of the authors, as well as the company USol as E-Vision's project coordinator for the valuable data and feedback provided. This project has been funded by the Spanish Ministry of Science MICYT DPI2010-20751-C02-01, the E-Vision Project (TSI-020100-2011-363) and the China Scholarship Council (CSC).

References

1. Nagy, C.J., Skoog, M.A., Somers, I.A., Cox, T.H., Warner, R.: Civil uav capability assessment. Technical report, NASA Dryden Flight Research Center (2004)
2. DeGarmo, M.T.: Issues concerning integration of unmanned aerial vehicles in civil airspace. Technical report, MITRE Center for Advanced Aviation System Development (2004)
3. Dempsey, M.: U.s army unmanned aircraft systems roadmap 2010-2035, Technical report, U.S. Army UAS Center of Excellence (2010)
4. Williamson, T., Spencer, N.A.: Development and operation of the traffic alert and collision avoidance system (tcas). In: Proceedings of the IEEE, vol. 77, pp. 1735–1744 (1989)
5. Finn, A., Franklin, S.: Acoustic sense & avoid for uav's. In: 2011 Seventh International Conference on Intelligent Sen-

sors, Sensor Networks and Information Processing (ISSNIP), pp. 586–589 (2011)
6. Barott, W.C., Coyle, E., Dabrowski, T., Hockley, C., Stansbury, R.S.: Passive multispectral sensor architecture for radar-eoir sensor fusion for low swap uas sense and avoid. In: Position, Location and Navigation Symposium - PLANS 2014, 2014 IEEE/ION, pp. 1188–1196 (2014)
7. Zarchan, P.: Tactical and Strategic Missile Guidance, American Institute of Aeronautics and Astronautics, Reston, VA, 4 edition (2002)
8. Shneydor, N.: Missile Guidance and Pursuit: Kinematics, Dinamics and Control. Horwood Publishing, UK (1998)
9. Limitations of the see-and-avoid principle, Technical report, Australian Transport Safety Bureau (1991)
10. Mian, A.S.: Realtime visual tracking of aircrafts. In: Digital Image Computing: Techniques and Applications (DICTA), p. 351 (2008)
11. Mejias, L., McNamara, S., Lai, J., Ford, J.: Vision-based detection and tracking of aerial targets for uav collision avoidance. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 87–92 (2010)
12. Lai, J., Ford, J.J., Mejias, L., O'Shea, P., Walker, R.: Detection versus false alarm characterisation of a vision-based airborne dim-target collision detection system. In: 2011 International Conference on Digital Image Computing Techniques and Applications (DICTA), pp. 448–455 (2011)
13. Lai, J., Mejias, L., Ford, J.J.: Airborne vision-based collision-detection system. J. Field Rob. **28**(2), 137–157 (2011)
14. Dey, D., Geyer, C.M., Singh, S., DiGioia, M.E.: Passive, long-range detection of aircraft: Towards a field deployable sense and avoid system. In: Proceedings Field & Service Robotics (FSR '09) (2009)
15. Forlenza, L., Fasano, G., Accardo, D., Moccia, A., Rispoli, A.: A hardware in the loop facility for testing multisensor sense and avoid systems. In: Digital Avionics Systems Conference, 2009. DASC '09. IEEE/AIAA 28th, pp. pages 5.C.4-1–5.C.4-10 (2009)
16. Fasano, G., Accardo, D., Forlenza, L., Renga, A., Rufino, G., Tancredi, U., Moccia, A.: Real-time hardware-in-the-loop laboratory testing for multisensor sense and avoid systems. Int. J. Aerosp. Eng. **2013**(748751) (2013)
17. Zsedrovits, T., Zarandy, A., Vanek, B., Peni, T., Bokor, J., Roska, T.: Collision avoidance for uav using visual detection. In: 2011 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2173–2176 (2011)
18. Zsedrovits, T., Zarandy, A., Vanek, B., Peni, T., Bokor, J., Roska, T.: Estimation of relative direction angle of distant, approaching airplane in sense-and-avoid. J. Intell. Robot. Syst. **69**(1–4), 407–415 (2013)
19. Delahaye, D., Puechmorel, S., Tsiotras, P., Feron, E.: Mathematical models for aircraft trajectory design: A survey. In: Air Traffic Management and Systems, Lecture Notes in Electrical Engineering, vol. 290, pp. 205–247. Springer, Japan (2014)
20. Brown, D.C.: Close-range camera calibration. Photogramm. Eng. **37**(8), 855–866 (1971)
21. Fryer, J.G., Brown, D.C.: Lens distortion for close-range photogrammetry. Photogramm. Eng. Remote. Sens. **52**, 51–58 (1986)
22. Carrio, A., Changhong, F., Pestana, J., Campoy, P.: A ground-truth video dataset for the development and evaluation of vision-based sense-and-avoid systems. In: 2014 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 441–446 (2014)
23. Changhong, F., Carrio, A., Olivares-Mendez, M.A., Suarez-Fernandez, R., Campoy, P.: Robust real-time vision-based aircraft tracking from unmanned aerial vehicles. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 5441–5446 (2014)

Adrian Carrio holds a degree in Industrial Engineering from the University of Oviedo, Spain. He is currently a Senior Researcher in the Computer Vision Group at the Technical University of Madrid, Spain, where he is currently working towards a PhD in Automation and Robotics. His research interests include collision avoidance systems for unmanned aerial vehicles, deep learning for pattern recognition and autonomous navigation.

Changhong Fu is a Senior Researcher and Ph.D. candidate in the Computer Vision Group at the Technical University of Madrid (CVG-UPM), Spain. He received his BEng and MEng degrees in Automation from Tianjin Polytechnic University and Xiamen University, China, in 2009 and 2011, respectively. Currently, his research areas are in the fields of vision-based tracking, odometry, simultaneous localization and mapping and control for the autonomy of unmanned aerial vehicles.

Jean-François Collumeau is a Senior Researcher in Computer Vision. His career path makes him versatile, having obtained a Ph.D. in Computer Vision and an M.Sc. in Energetics and Space at the University of Orleans (France) and has an engineering degree in Industrial Risk Management from the ENSI de Bourges (France). His main research interests include vision-based localization and recognition of objects in the context of applied Computer Vision.

Pascual Campoy is Full Professor in Automation at the Technical University of Madrid (UPM), where he received his PhD in Engineering in 1988. He presently lectures on Control, Machine Learning and Computer Vision at graduated and post-graduated level. He is leading the Research Group on Computer Vision at U.P.M., which is aimed in increasing the autonomy of the Unmanned Aerial Vehicles (UAV) by exploiting the powerful sensor of Vision. He is presently working on new learning paradigms for bio-inspired Neural Networks which aim to be used in image segmentation, detection and recognition.