Universidad Politécnica de Madrid
Escuela Técnica Superior De Ingenieros Industriales

**Vision-based Aircraft Detection Technologies
for Unmanned Aerial Vehicles**

PhD Thesis

Adrián Carrio Fernández
Ingeniero Industrial

2020

Departamento de Automática, Ingeniería Eléctrica y Electrónica
e Informática Industrial
Escuela Técnica Superior De Ingenieros Industriales
Universidad Politécnica de Madrid

Vision-based Aircraft Detection Technologies for Unmanned Aerial Vehicles

A thesis submitted for the degree of
Doctor of Philosophy in Automation and Robotics

Author:
Adrián Carrio Fernández
Ingeniero Industrial

Director:
Dr. Pascual Campoy Cervera
Full Professor

Madrid, Septiembre 2020

Título:

Vision-based Aircraft Detection Technologies for Unmanned Aerial Vehicles

Autor:

Adrián Carrio Fernández

Ingeniero Industrial

Director:

Dr. Pascual Campoy Cervera

Full Professor

Tribunal nombrado por el Mgfco. y Excmo. Sr Rector de la Universidad
Politécnica de Madrid el día de de 2020

Tribunal

Presidente : Dr. Sergio Domínguez Cabrerizo

Vocal : Dr. Luis Mejías Álvarez

Vocal : Dr. Juan López Otero

Vocal : Dr. Ignacio Álvarez García

Secretario : Dr. Pablo San Segundo Carrillo

Suplente : Dra. Carol Viviana Martínez Luna

Suplente : Dr. Bernardo Morcego Seix

Realizado el acto de lectura y defensa de la tesis el día de
..... de 2020.

Calificación de la Tesis.....

El Presidente:

Los Vocales:

El Secretario:

*A mis padres, Severino y Esther,
que me han dado tanto,
a mis queridos hermanos,
Marisa y Manuel,
y a mi amada Luisana,
por su paciencia, apoyo y
cariño incondicionales.*

Acknowledgements

I would like to express my most sincere gratitude to all the people who have contributed to the realization of this PhD thesis. Firstly, I would like to acknowledge my advisor Prof. Pascual Campoy Cervera, not only for his academic guidance and support, but also for his principles, his one-of-a-kind attitude and his tireless dedication, both as a lecturer and as head of the Computer Vision and Aerial Robotics (CVAR) research group.

Special thanks to all CVAR members for their continuous disposition to help and for all that I have learned from them. This work would have never happened without the supportive attitude of: Changhong Fu, Carol Martínez, José Luis Sánchez, Jesús Pestana, Hriday Bavle, Carlos Sampedro, Ramón Suárez, Aneesh Chauhan, Liang Lu, Javier Rodríguez and Alejandro Rodríguez. My outspoken gratitude also goes to all other students that have contributed and those who contribute today to the excellence of this great research group.

I would also like to thank some members from the Centre for Automation and Robotics (CAR) and in particular, some professors that have helped along the way: Sergio Domínguez, Paloma De La Puente, Manuel Ferrer, Martín Molina, Roque Salterén, Pablo San Segundo, other outstanding PhD students: Juan Cely, Andrés Parra and Jorge de León, also Ángel, our incredible lab technician, and the lovely current and former administrative staff: Carlos, Rosa, Teresa and Paloma. Thanks for being so helpful and supportive.

My gratitude also goes to the people I met while doing research abroad, both at ASU and at MIT. Thanks to Prof. Srikanth Saripalli and to Prof. Jonathan How for the opportunity to do research under their guidance and to all the incredible people I met along the way: Andres Mora, Sai Vemprala, Ben Stinnet, Yucong Lin, Anand Biradar, Jesús Tordesillas, Dong-Ki Kim, Michael Everett and Brett Lopez.

Special thanks to my dear friends, Miguel, Daniel and Andrea and, of course, to my parents and siblings and my beloved Luisana. This thesis would have not been possible without all of you.

Last but not least, I would also like to thank all the people who, while not specifically mentioned, have as well contributed to the realization of this PhD thesis.

Resumen

La detección de aeronaves es un problema complejo con multitud de aplicaciones en el ámbito de los vehículos aéreos no tripulados (UAVs), entre las que destacan los sistemas de detección y evasión (DAA), los sistemas de evasión de colisiones para enjambres de UAVs y los sistemas anti UAVs. La presente tesis doctoral pretende ofrecer una visión global del problema de detección visual de aeronaves y las tecnologías asociadas, presentando las contribuciones de su autor en este ámbito.

Con respecto al uso de la visión en sistemas DAA, cuyo propósito es permitir a los UAVs compartir de manera segura porciones del espacio aéreo con aeronaves tripuladas, este trabajo aborda las dificultades que implica desarrollar sistemas fiables basados en visión. Estos problemas surgen tanto de la complejidad de tratar con la gran diversidad de posibles escenarios de navegación en exteriores, con muy diferentes condiciones de iluminación, como de la gran variedad de posibles marcos en los que se puede producirse una colisión aérea. Con el objetivo de obtener datos relevantes que puedan ser utilizados para desarrollar y evaluar algoritmos de detección y seguimiento visuales, se ha propuesto un software capaz de generar imágenes sintéticas de escenarios de colisión aérea en el espectro visible, reduciendo el coste de obtener imágenes reales equivalentes y suprimiendo los riesgos de realizar vuelos que simulen escenarios de posible colisión. Estas imágenes se obtienen combinando imágenes aéreas reales con imágenes de aeronaves en vuelo generadas por ordenador. La calidad y el realismo de estas imágenes ha sido evaluado positivamente por pilotos profesionales mediante tests visuales. Además, se ha desarrollado un novedoso y eficiente algoritmo de detección empleando las mencionadas imágenes, el cual ha sido comparado con el estado del arte, concretamente con un eminente algoritmo de detección de objetos en tiempo real.

La utilización de la termografía infrarroja para la detección de aeronaves también ha sido objeto de estudio. En particular, se ha desarrollado un sistema de detección de aeronaves para pequeños UAVs, el cual integra una cámara térmica y un sistema receptor de tecnología ADS-B. Se ha desarrollado un algoritmo eficiente para la detección de aeronaves en imágenes térmicas el cual ha sido evaluado con imágenes aéreas capturadas junto a un aeropuerto. Las aeronaves detectadas visualmente se han asociado a los datos obtenidos con el receptor ADS-B para estimar la distancia a las mismas. El sistema propuesto se ha demostrado capaz de detectar aeronaves

bajo complicadas condiciones de iluminación e incluso por la noche con un coste computacional muy reducido.

Finalmente, se ha propuesto un novedoso enfoque para la detección eficiente de aeronaves a media y corta distancia utilizando mapas de profundidad: una representación alternativa a las nubes de puntos, obtenida de sensores LIDAR o cámaras de profundidad. El principio de detección de basa en la explotación de las discontinuidades de profundidad características que forman los objetos en vuelo. Adicionalmente, las detecciones sobre mapas de profundidad permiten obtener de manera sencilla la posición 3D de la aeronave detectada, lo cual puede ser utilizado a posteriori para la planificación de trayectorias libres de colisión. Se ha validado en simulación un método de detección eficiente basado en aprendizaje profundo utilizando mapas de profundidad provenientes de nubes de puntos LIDAR y en vuelos reales utilizando una cámara estereoscópica. Para su validación, el sistema se ha integrado a bordo de un pequeño UAV volando a diferentes velocidades y realizando varias trayectorias, y las estimaciones de las posiciones 3D de las aeronaves detectadas se han validado utilizando un sistema de captura del movimiento.

En resumen, esta tesis doctoral explora varias tecnologías basadas en visión aplicables al problema de detección de aeronaves y demuestra su enorme potencial, el cual destaca en aplicaciones reales por el bajo peso, tamaño y consumo energético de este tipo de sensores. Este trabajo se centra en el despliegue de estas tecnologías en pequeñas plataformas UAV, proporcionando soluciones eficientes, que están apoyadas en experimentos de vuelo real y publicaciones en revistas internacionales de impacto, y que extienden el estado del arte en este campo.

Abstract

Vision-based aircraft detection is a complex problem with multiple applications in the field of unmanned aerial vehicles (UAVs) including detect-and-avoid (DAA) systems, collision avoidance systems for UAV swarms and anti-UAV systems. The present Thesis aims at providing a global view of the problem of vision-based aircraft detection and the technologies associated and presents the author's contributions to the field.

With respect to the use of vision in DAA systems, aimed at allowing UAVs to safely share portions of the airspace with manned aircraft, this work addresses the difficulties for developing reliable vision-based systems. These problems arise both from the complexity of dealing with a great diversity of possible outdoor navigation scenarios, with very different illumination conditions, as well as with the great variety of mid-air collision settings. In order to obtain relevant data that can be used to develop and evaluate visual detection and tracking algorithms, a software framework has been proposed capable of generating synthetic visible-spectrum images of mid-air collision scenarios, thus reducing the cost of obtaining equivalent real images and eliminating the risks of performing actual flight encounters. These images are obtained by blending real aerial images with computer-generated imagery of flying aircraft. The quality and realism of these images has been positively evaluated by professional pilots in visual tests. Furthermore, a novel and efficient detection algorithm has been developed using the aforementioned images, and has been compared to a state-of-art, real-time object detection algorithm.

The use of thermal infrared (TIR) imaging for aircraft detection has also been object of study. In particular, an aircraft detection system for small UAVs has been developed, which integrates a TIR camera and an Automatic Dependent Surveillance-Broadcast (ADS-B) receiver. An efficient algorithm for the detection of aircraft in TIR images is developed and evaluated with aerial images captured near an airport. Detected aircraft have been matched with data from the ADS-B receiver in order to obtain range estimations to the visually detected aircraft. The proposed system is proved to detect aircraft at real-time frame rates under challenging illumination conditions and even at night.

Finally, a novel approach for efficient, close-range aircraft detection using depth maps has been proposed. Depth maps are an alternative representation of pointclouds, obtained from LIDAR sensors or depth cameras. The

detection principle relies on the exploitation of the characteristic discontinuities in depth caused by flying objects. Additionally, detections on depth maps seamlessly provide the 3D relative position of the detected aircraft, which can be later exploited for collision-free path planning. An efficient detection approach, based on deep learning, has been validated in simulation using depth maps from LIDAR pointclouds, and also in real flights using a stereoscopic camera. For its validation, the system has been integrated on board a small UAV flying at various speeds and performing various trajectories, and the estimations of the 3D positions of the detected aircraft have been validated using a motion capture system.

In summary, this Thesis explores various vision-based technologies applicable to the problem of aircraft detection and demonstrates their enormous potential, which is enhanced in real applications by the low weight, size and power consumption of these sensors. This work focuses on the deployment of these technologies on small UAV platforms, providing efficient solutions, which are backed by real flight experiments and publications in top international journals, and which extend the state of art in the field.

Acronyms

ACAS: Airborne Collision Avoidance System
ADS-B: Automatic Dependent Surveillance - Broadcast
CAR: Centre for Automation and Robotics
CVAR: Computer Vision and Aerial Robotics Group
DAA: Detect and Avoid
DEM: Digital Elevation Map/Model
DL: Deep Learning
EHF: Extremely High Frequency
ELOS: Equivalent Level of Safety
EO: Electro-optical (sensor)
FFC: Flat field correction
FOV: Field Of View
GAN: Generative Adversarial Networks
GPS: Global Positioning System
GPU: Graphics Processing Unit
GUI: Graphical User Interface
HMT: Hidden Markov Tree
IMC: Instrument Meteorological Conditions
IMU: Inertial Measurement Unit
INS: Inertial Navigation System
IoU: Intersection over Union
IR: Infrared (spectrum)
LIDAR: Laser Imaging, Detection and Ranging
mAP: Mean Average Precision
MTOM: Maximum Take-off Mass
NEDT: Noise equivalent differential temperature
OFDM: Orthogonal Frequency Division Multiplexing
RCD: Radar Cross Section
SAA: See and Avoid
SAR: Synthetic Aperture Radar
SHF: Super High Frequency
SIGS: Synthetic Imagery Generating Software
SNR: Signal-to-noise ratio
SSR: Secondary Surveillance Radar
SVM: Support Vector Machine
TCAS: Traffic Alert and Collision Avoidance System

TIR: Thermal Infrared (Spectrum)
UAV: Unmanned Aerial Vehicle
UHF: Ultra High Frequency
UPM: Universidad Politécnica de Madrid
UV: Ultraviolet (spectrum)
UWB: Ultra Wide Band
VFR: Visual Flight Rules
VIS: Visible (Spectrum)
VMC: Visual Meteorological Conditions

Contents

Acknowledgements	I
Resumen	III
Abstract	V
Acronyms	VII
List of Figures	XIII
List of Tables	XVII
1. Introduction	1
1.1. Motivation and overview	1
1.2. Problem statement	4
1.3. Dissertation outline and contributions	6
1.4. List of publications	6
1.4.1. Journal Publications	7
1.4.2. Publications in International Conferences	7
2. State of the art	9
2.1. Introduction	9
2.2. Cooperative systems	10
2.3. Non-cooperative systems	11
2.3.1. Active sensing technologies	11
2.3.2. Passive sensing technologies	14
2.4. Hybrid systems	23
2.5. Specific areas of research in Computer Vision	24
2.5.1. Synthetic image generation	24
2.5.2. Sky segmentation and horizon line detection	26
2.5.3. Real-time object detection	30
2.6. Conclusions	33

3. Aircraft detection in the visible spectrum	37
3.1. Introduction	37
3.2. Synthetic image generation	38
3.2.1. Methodology	38
3.2.2. Experiments and results	48
3.2.3. Conclusions and future works	54
3.3. Sky segmentation and horizon detection	57
3.3.1. Proposed approach	58
3.3.2. Evaluation methodology	60
3.3.3. Results and discussion	63
3.3.4. Conclusion and future works	67
3.4. Aircraft detection algorithm	69
3.4.1. Proposed approach	69
3.4.2. Evaluation methodology	70
3.4.3. Experiments and results	73
3.4.4. Conclusions and future works	74
3.5. Conclusions	75
4. Aircraft detection in the thermal infrared spectrum	79
4.1. Introduction	79
4.2. On-board detection system	80
4.2.1. System description	80
4.2.2. Data collection	87
4.2.3. Experiments and results	87
4.2.4. Conclusions and future work	94
4.3. Additional methods for sky segmentation	95
4.3.1. Infinite horizon method	96
4.3.2. ConvNet-based method	98
4.3.3. Data acquisition	99
4.3.4. Evaluation methodology	101
4.3.5. Experiments and results	103
4.3.6. Conclusions	106
4.4. Conclusions	106
5. Aircraft detection using depth maps	109
5.1. Introduction	109
5.2. Depth maps using stereoscopic vision	110
5.2.1. Detection and localization method	111
5.2.2. Dataset	113
5.2.3. Experimental setup	115
5.2.4. Experiments and Results	116

5.2.5.	Conclusions and future work	119
5.3.	Depth maps using LIDAR	119
5.3.1.	Dataset generation	120
5.3.2.	Detection models	122
5.3.3.	Evaluation methodology	125
5.3.4.	Experiments and Results	125
5.3.5.	Conclusions and future work	127
5.4.	On-board detection and localization	130
5.4.1.	Proposed approach	130
5.4.2.	Hardware setup	136
5.4.3.	Evaluation methodology	138
5.4.4.	Experiments	140
5.4.5.	Results	143
5.4.6.	Conclusions and future work	145
5.5.	Conclusions	146
6.	Conclusions and future work	149
6.1.	Conclusions	149
6.2.	Future work	151
A.	Scientific Dissemination	153
A.1.	Publications	153
A.1.1.	Journals	153
A.1.2.	Conferences	155
A.2.	Research stays / exchange	157
A.3.	Technology transfer projects	157

List of Figures

1.1. Applications of vision-based aircraft detection	3
3.1. 3D models for simulating flying obstacles.	39
3.2. Synthetic image of aircraft over a chroma key green backdrop	40
3.3. Reference systems for synthetic image generation.	41
3.4. Comparison of results over two image compositing algorithms.	42
3.5. Image blending process with sample images.	43
3.6. Results of the automatic UAV occlusion detection algorithm.	46
3.7. Sum of normalized differences between successive frames. . .	47
3.8. UAV mask refinement process.	47
3.9. Examples of automatically produced UAV masks.	48
3.10. Main window of the implemented graphical user interface. .	49
3.11. Captures from real flight videos used as backgrounds. . . .	51
3.12. Center Location Error over 1LRB737 video.	52
3.13. Center Location Error over 3LRB737 video.	52
3.14. Sample images shown to pilots.	55
3.15. Pilots guesses about the number of aircraft present in the image.	56
3.16. Pilots guesses on the type of aircraft for a single aircraft. . .	56
3.17. Pilots guesses on the type of aircraft for two aircraft. . . .	57
3.18. Pictures extracted from the sky segmentation dataset. . . .	61
3.19. Normalized correlation coefficient between the average sky and ground histograms.	64
3.20. Examples of the proposed post-processing strategy for remo- ving misclassified patches based on patch connectivity. . . .	66
3.21. Final sky segmentation results.	67
3.22. Automatic horizon line detection	68
3.23. Intersection over union.	71
4.1. Main components of the proposed aircraft detection system.	81
4.2. Stationary ground system with both VIS and TIR cameras and an ADS-B receiver.	81

4.3. UAV prototype with VIS and TIR cameras and an ADS-B receiver.	84
4.4. Screenshot of the application developed for visualizing the positions of aircraft broadcasting ADS-B messages and their timestamps.	87
4.5. Examples of VIS and TIR images.	88
4.6. Automatic horizon line detections.	89
4.8. Aircraft detections in a frame sequence captured with the ground-based prototype.	91
4.7. Automatic aircraft detections in a video sequence captured on board.	92
4.9. Automatic aircraft detections under different lighting and thermal radiation settings.	93
4.10. Automatic long-range detection (10.17Km).	94
4.11. Horizon line detection using the infinite horizon method. . .	97
4.12. Examples of VIS and TIR images from the three video sequences captured.	102
4.13. Attitude estimations for roll and pitch angles.	105
5.1. Visual aspect of a flying object in a depth map.	110
5.2. Speed and accuracy of various object detection models. . . .	112
5.3. Three UAV models used for building the training dataset. . .	114
5.4. Environments created within Unreal Engine.	114
5.5. Sample images from the depth dataset.	115
5.6. Obtaining depth maps.	122
5.7. Models used as target UAVs in the simulation.	123
5.8. Role of the collision model in the depth maps.	123
5.9. SSD model.	124
5.10. LIDAR detection examples.	127
5.11. Precision-recall curves for each of the four models tested. . .	128
5.12. Detection system using depth maps.	132
5.13. CNN architecture for UAV detection.	132
5.14. Sample images from the depth dataset.	135
5.15. UAV platform used as host in the experiments.	136
5.16. Three target UAVs used in the experiments.	137
5.17. Graphical explanation of the performed experiments. . . .	139
5.18. Results for experiment 3.	141
5.19. Results for experiment 4.	142
5.20. Results for experiment 5.	142
5.21. Results for experiment 6.	142
5.22. Results for experiment 7.	142

List of Figures

5.23. Results for experiment 8.	142
5.24. Precision-recall curve of the UAV detection model.	143

List of Tables

1.1. Relative importances of different requirements in four relevant vision-based UAV detection applications.	5
2.1. Comparison of sensing technologies for aircraft detection . .	35
3.1. Video dataset used in the development of the AM ³ tracker. .	51
3.2. Details about the pilots participating in the test.	54
3.3. Sky segmentation dataset.	62
3.4. Confusion matrices and F1 score indicating the performance of the trained classifier.	65
3.5. Sky segmentation performance estimation.	66
3.6. Average processing time per frame for sky segmentation. . .	68
3.7. mAP for the evaluated detection algorithms	74
3.8. Average processing times of detection algorithms.	74
4.1. Visual detection performance using TIR images.	91
4.2. Largest ranges from visual detections, estimated using ADS-B data.	93
4.3. CNN model architecture proposed.	99
4.4. Train and validation errors obtained during the training process of the ConvNet models.	103
4.5. RMS pitch and roll errors.	104
4.6. Comparison of vision-based attitude estimation methods. . .	104
5.1. Precision and recall in online detection	117
5.2. Comparison of depth estimation methods	118
5.3. mAP as a function of the type of LIDAR.	128
5.4. mAP as a function of the type of the type of UAV.	128
5.5. mAP as a function of the distance.	129
5.6. Detection range using LIDAR (all dimensions in meters). . .	129
5.7. Inference time of the tested models (seconds).	129
5.8. Dimensions of the target UAVs used in the experiments. . .	137
5.9. Execution times	145

Chapter 1

Introduction

*It is of the highest importance
in the art of detection to be able to
recognise, out of a number of
facts, which are incidental and
which are vital.*

Arthur Conan Doyle

1.1. Motivation and overview

Unmanned Aerial Vehicles (UAVs) offer a number of operational benefits over manned aircraft. Because there is no on-board pilot, UAVs can be used to execute dangerous missions or be operated in harmful environments without putting the lives of pilots at risk. Also, their operational costs are, in general, much lower than those of manned aircraft and the human resources involved are generally lower. Additionally, whenever the pilot is on board the aircraft, pilot fatigue is a relevant factor to consider, which may limit the duration of the flight or the characteristics of the mission. In the case of UAVs, human pilots are not always required for operating the aircraft, but even when they are, their fatigue is no longer a concern, as the piloting happens off board and pilots can be relieved as many times as necessary without the aircraft having to land. These benefits have encouraged the use of UAVs in countless applications, typically in dull, dirty and/or dangerous missions and have also opened up the possibility of new applications, which cannot be performed by manned aircraft.

In order to operate safely in airspaces shared with manned aircraft, UAVs must follow the existing practices and procedures so that current air management systems do not need to be drastically modified nor their safety degraded. In this sense, one of the methods used for collision avoidance in manned aviation is the “See and Avoid” (SAA) principle (Regulation 14 CFR, 2002), which is required in order to avoid collisions with conflicting traffic. According to this principle, pilots operating manned aircraft must regularly scan across their field of view from the cockpit and look for other aircraft, make judgments about the trajectories of those aircraft and determine whether maneuvering is required to avoid the risk of collision. Besides visibility, this capability is highly dependent on the ability and behavior of the pilots.

When applied to UAVs, SAA has evolved into the augmented concepts of “Sense and Avoid” (Angelov, 2012) and the more generic “Detect and Avoid” (Annex 2 ICAO, 2005). These terms refer to the expected capabilities of UAVs aimed at avoiding collisions not only with other traffic but also with terrain and obstacles, and not only by means of visual perception, but also using other sensing technologies or even mitigating collision risks with operational procedures. In particular, DAA is defined as “the capability to see, sense or detect conflicting traffic or other hazards and take the appropriate action” (Annex 2 ICAO, 2005).

DAA is a capability not only required for the safe integration of UAVs in general aviation, but also so that multiple UAVs can fly safely in airspaces that are not shared with manned aircraft using distributed decision making. This is the case for several UAV applications, mainly those taking place in indoor environments, and particularly those involving swarms of UAVs.

Architecting DAA systems has been thoroughly studied (Hutchings et al., 2007), and three stages have been identified in the collision avoidance process: detection, avoidance and resolution (Mcfadyen and Mejias, 2016). A great part of this Thesis focuses in the first of them, which involves the perception of potential collision threats, and more particularly the visual perception of aircraft.

Being the sense used by human pilots, vision-based detection (i.e. using electro-optical sensors) seems the most natural approach to implement DAA systems. Furthermore, the properties of electro-optical sensors: low weight, power consumption and cost, make them particularly appropriate for UAVs, with two main drawbacks: the dependency on the visibility conditions - which also applies to human pilots anyway- and the relatively high computational cost of the processing algorithms.

Additionally, and even beyond specific solutions for UAVs, aircraft detec-

Chapter 1. Introduction

(a) DAA systems

(b) UAV swarms

(c) Anti-UAV systems

(d) Remote sensing applications

Figure 1.1: Applications of vision-based aircraft detection

tion has other relevant applications in safety, security and privacy. Today's ease of access and popularity of UAVs has raised concerns about malicious uses including terrorist and electronic attacks, smuggling, reconnaissance and distraction using UAVs. Vision-based aircraft detection technologies can definitely be a part of the solution to these potential threats. Integrated in anti-UAV systems and deployed in critical infrastructures (e.g. government buildings, nuclear plants) or wherever they are needed, they can be used to anticipate UAV attacks in order to neutralize them. The most relevant applications of vision-based aircraft detection are depicted in Fig. 1.1.

In summary, aircraft detection is a key technology to safely deploy UAVs in airspaces shared with other aircraft. Visual perception seems an appropriate approach, given the way human pilots resolve potential collision conflicts and the adequate properties of electro-optical sensors. Furthermore, aircraft detection not only can provide UAVs with collision avoidance capabilities, but is also a powerful technology for applications in which safety, security and privacy may be at risk.

1.2. Problem statement

Object recognition, which aims at finding and identifying objects in an image or video sequence, is one of the most complex tasks in Computer Vision. When the object being searched for is known in advance, the problem is known as *object detection*. It involves quickly scanning an image to determine where a match may occur (Szeliski, 2010).

In the case of visual detection of aircraft, there are several aspects to consider, including:

- Visibility requirements.
- Detection of small (few-pixels) aircraft.
- Integrability on board (small) UAVs.
- Real-time performance.
- Need for estimating the range and/or relative velocity of the aircraft.
- Tolerance to false positives from other flying objects (e.g. birds).
- Requirement for continuous detection.

Due to the many factors involved, the problem of vision-based aircraft detection may have various requirements depending on the specific application. Furthermore, when developing DAA systems the requirements are not easy to define, as the reference standard -human performance in real SAA situations- is hard and risky to evaluate.

Among the various applications presented, DAA systems have the highest requirements for several reasons: they are critical real-time systems, which should operate robustly in as many meteorological conditions as possible and they are typically integrated on board UAVs that fly at high velocity, which additionally demands a higher detection range and field of view (FOV), in order to timely detect potential aircraft in collision course. In DAA systems, the crossing velocities of aircraft are in the range of several meters per seconds and the detection ranges in the range of kilometers. Additionally, when dealing with fully-automatic collision avoidance systems, estimating the range and/or relative velocity of the detected aircraft may as well be crucial to guarantee a safe avoidance maneuver that will resolve the aircraft encounter. Given the complexity of the problem, research continues today in order to come up with standard requirements, procedures and equipment for DAA systems.

Chapter 1. Introduction

Decentralized UAV swarms typically have looser requirements, as the distance between UAVs in a swarm is typically shorter (tens of meters) and the velocities are generally lower (few meters per second). Also because these systems are often operated in indoor environments, requirements for meteorological conditions do not apply.

In the case of anti-UAV systems, the main difference is that they are not integrated on board UAVs, so they have less constrained weight, size and power consumption requirements. With virtually unlimited computing power, the need for powerful processing resources -one of the key challenges in Computer Vision- is less of a problem. Additionally, and in contrast with DAA systems, where false positives from the detection of birds are not a concern as birds are just another airborne object to avoid, alarms in anti-UAV systems caused by birds are false alarms and therefore not desirable.

Finally, remote sensing applications usually have looser requirements than all other aircraft detection applications. While the required range of detection can be high, the aircraft being detected are static and therefore a small FOV can be used, which makes aircraft typically appear with a relatively large size in the image. Another relevant simplification is that the processing can be done off board and off line.

The relative importances of all the aforementioned factors according to the application are summarized in Table 1.1.

	DAA systems	UAV swarms	Anti-UAV systems	Remote sensing applications
Visibility requirements	High	Low	High	High
Detection of small (few-pixels) aircraft	High	Low	High	Low
Integrability on board UAVs	High	High	Low	Low
Real-time performance	High	High	High	Low
Range and/or relative velocity estimation	High	High	High	Low
Aircraft-only detection (i.e. not birds)	Low	Low	High	Low
Continuous detection	High	High	High	Low

Table 1.1: Relative importances of different requirements in four relevant vision-based UAV detection applications.

The goal of this research is to contribute to the development of vision-based aircraft detection systems, particularly those suitable for small UAVs, through the generation of relevant datasets, the exploitation of depth maps -among other image representations- and the application of deep learning techniques.

1.3. Dissertation outline and contributions

From all of the aforementioned applications, this Thesis focuses mainly in aircraft detection in multiple application scenarios, using visible and thermal infrared cameras and additionally exploring the applicability of a novel approach for aircraft detection based on depth maps.

Chapter 2 covers the state of art in aircraft detection, providing an overview of all the existing technologies, with a focus in vision-based techniques, and further discussing the main challenges of the aircraft detection problem with vision.

In Chapter 3 the topic of aircraft detection using visible spectrum cameras is discussed. In this chapter, the following contributions of this Thesis are presented:

- The development of a software for synthetic image generation, a key tool for the research and improvement of DAA systems.
- A novel algorithm for the detection of self-occlusions onboard UAVs.
- A novel machine learning-based sky segmentation algorithm.
- A novel aircraft detection algorithm.

Chapter 4 discusses aircraft detection using thermal infrared images. In this chapter, a novel, computationally-efficient approach for aircraft detection in thermal images is presented and validated on board a small UAV and two novel algorithms for horizon detection in thermal images are presented.

In Chapter 5 a novel approach for aircraft detection using depth maps is covered. This detection method is another contribution of the present Thesis.

Finally, Chapter 6 includes the conclusions and future work.

1.4. List of publications

The work presented in this Thesis is based on the following publications:

1.4.1. Journal Publications

- Carrio, A., Fu, C., Collumeau, J. F., & Campoy, P. (2016). SIGS: Synthetic imagery generating software for the development and evaluation of vision-based sense-and-avoid systems. *Journal of Intelligent & Robotic Systems*, 84(1-4), 559-574. (Chapter 3)
- Carrio, A., Sampedro, C., Rodriguez-Ramos, A., & Campoy, P. (2017). A review of deep learning methods and applications for unmanned aerial vehicles. *Journal of Sensors*, 2017. JCR 2018: 2.024 (Chapter 2)
- Carrio, A., Lin, Y., Saripalli, S., & Campoy, P. (2017). Obstacle detection system for small UAVs using ADS-B and thermal imaging. *Journal of Intelligent & Robotic Systems*, 88(2-4), 583-595. (Chapter 4)
- Carrio, A., Bavle, H., & Campoy, P. (2018). Attitude estimation using horizon detection in thermal images. *International Journal of Micro Air Vehicles*, 10(4), 352-361. (Chapter 4)
- Carrio, A., Tordesillas, J., Vemprala, S., Saripalli, S., Campoy, P., & How, J. P. (2020). Onboard Detection and Localization of Drones Using Depth Maps. *IEEE Access*, 8, 30480-30490. (Chapter 5)

1.4.2. Publications in International Conferences

- Carrio, A., Fu, C., Pestana, J., & Campoy, P. (2014, May). 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). IEEE. (Chapter 3)
- Carrio, A., Sampedro, C., Fu, C., Collumeau, J. F., & Campoy, P. (2016, June). A real-time supervised learning approach for sky segmentation on board unmanned aerial vehicles. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 8-14). IEEE. (Chapter 3)
- Carrio, A., Vemprala, S., Ripoll, A., Saripalli, S., & Campoy, P. (2018, October). Drone detection using depth maps. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1034-1037). IEEE. (Chapter 5)

The rest of the publications that have contributed indirectly to the achievement of this PhD Thesis or in which the author of this Thesis has participated as co-author are shown in Appendix A.

Chapter 2

State of the art

2.1. Introduction

There is a wide variety of applicable technologies for aircraft sensing, which can be divided into two main categories: cooperative and non-cooperative systems (Angelov, 2012). While this categorization is most commonly used in the context of DAA systems, it can also be used to classify aircraft detection systems in general. Cooperative systems rely on standardized and mature technologies and are based in the mutual sharing of location information. This is usually done by integrating systems on board the aircraft which transmit its location information. However, these systems are not useful in collision situations with aircraft which do not integrate them or to avoid other potential collision threats, such as birds. Moreover these systems have other disadvantages related mainly to their cost, weight and power-consumption.

On the other hand, non-cooperative systems do not require other aircraft to equip any devices in order to avoid collisions and they can detect and avoid other flying objects besides cooperative aircraft. In general, non-cooperative systems suffer detection and tracking difficulties at long distances but become accurate and increasingly reliable at close distances (Moore, 2019).

This chapter will provide an overview of the most common aircraft sensing technologies, focusing on those relevant to the field of UAVs, either because they have suitable properties for being integrated on board UAVs or because they provide UAV detection capabilities. The state of the art of certain specific areas of research relevant to the topic of aircraft detection

using Computer Vision which deserve special attention have a dedicated section at the end of the chapter.

2.2. Cooperative systems

Today, cooperative systems are best represented by two main technology standards, which are both currently in use for aircraft-to-aircraft collision avoidance in manned aviation: Traffic Alert and Collision Avoidance Systems (TCAS) and Automatic Dependent Surveillance - Broadcast (ADS-B).

The Traffic Alert and Collision Avoidance System (TCAS), very common in manned aircraft, is a mature cooperative system (Williamson and Spencer, 1989; Billingsley et al., 2012). It is an implementation of the Airborne Collision Avoidance System (ACAS) standard, mandated by the International Civil Aviation Organization to be fitted to all aircraft with a maximum take-off mass (MTOM) of over 5,700 kg (12,600 lb) or authorized to carry more than 19 passengers. The transponder is a piece of onboard equipment designed to respond to various types of requests using radio signals. These requests, known as interrogations, can be made from both ground SSR (Secondary Surveillance Radar) systems and other aircraft equipped with TCAS transponders -an specific class of transponders-. TCAS-equipped aircraft will therefore receive the identification and ranges of all nearby aircraft equipped with a functional transponder, warning the pilot when separation is too small. Furthermore, if both aircraft are equipped with modern TCAS transponders, the pilot will also receive indications for resolution maneuvers to prevent a collision. The latest versions of TCAS have detection ranges of up to 160 Km. This sort of technology has been considered for potential use in UAVs (Zeitlin and McLaughlin, 2007; Lee, 2006). However, TCAS may not directly be applied for small-to-medium UAVs due to payload limitations and cost. The communication between manned and unmanned aircraft is another challenging issue. Additionally, TCAS has limited ability to handle the case of multiple aircraft avoidance.

Automatic Dependent Surveillance - Broadcast (ADS-B), is based on a periodic radio broadcast of the identification, position, velocity, and intent of the carrying aircraft and has gained considerable interest in the aviation industry as the next generation of surveillance technology (Samuelson et al., 2006). Global Positioning System (GPS) is integrated in ADS-B to obtain the information of the position, altitude, and velocity of aircraft. Ground stations and other aircraft within a roughly 240 km radius can receive the data-link broadcasted by the aircraft. This technology is aimed at reducing the reliance on high-cost older SSRs. The introduction of ADS-B also

enables advanced operational procedures like self-spacing. With these types of procedures and reduced separation standards, airspace capacity can be increased and safety can be improved. ADS-B is mandatory in Europe since 2018 and in the US since Jan. 1, 2020.

ADS-B is a suitable technology for DAA as it can provide accurate and reliable information of navigation variables. It is also based on a well-proven communication technology, plus it has a flexible structure, which allows for an easy implementation and incorporation of future techniques. The implementation of ADS-B for collision avoidance on board UAVs has already been considered (Lin and Saripalli, 2015, Martel et al., 2009, Martel et al., 2011, Foerster et al., 2012). In these approaches, ADS-B transceivers are not carried onboard small UAVs, mainly due to the power limitations onboard and ADS-B receivers are used instead. In these systems ADS-B data may not always be available, depending on the antenna carried. Furthermore, as a cooperative approach, ADS-B is ineffective for the detection of certain flying objects (e.g. non-equipped aircraft and birds) and ground-based obstacles, such as terrain, towers, power lines or people (Barnhart et al., 2012).

2.3. Non-cooperative systems

Non-cooperative systems are divided into two groups, according to the nature of the sensors they rely on: active or passive. Active sensors are based in the transmission of electromagnetic signals as part of the sensing process, whereas passive sensors rely on signals already generated by the target, other systems or on signals already present in the environment in order to detect aircraft. Detection with passive sensors generally relies on certain a priori knowledge of the environment. Active sensors generally use echolocation or reflectometry principles to measure the distance to other aircraft. Additionally, velocity can be measured by means of the Doppler effect (Moore, 2019). Active sensing technologies include active radar systems and Laser Imaging, Detection and Ranging (LIDAR) systems, while the most relevant passive sensing technologies are electro-optical (EO) systems, acoustic systems, and passive radar systems.

2.3.1. Active sensing technologies

Active radar systems

Active radar is considered the best sensing solution for airborne surveillance (Chen et al., 2011) and is currently the main active sensing technology used for aircraft detection. It typically relies on microwave signals and is

commonly applied for detecting motion and determining location, velocity, and size of ground targets and whilst suitable for large aircraft, its use on board medium-size and smaller platforms is limited due to the onboard size, weight and power restrictions and to the cost of the equipment.

Microwave bands used in radar systems cover a spectrum that ranges between 300 MHz and 300 GHz can be grouped into three main groups:

- Ultra high frequency (UHF) bands (300 MHz - 1 GHz)
- Super high frequency (SHF) bands (1 GHz - 30 GHz)
- Extremely high frequency (EHF) bands (30 GHz - 300 GHz)

UHF bands are typically used for the transmission of telephony, radio and television signals, so these bands are more commonly used for passive radar systems. SHF radars rely on the L, S, C, X and K bands. This frequency range also contains some commonly used bands, such as WiFi (2.4 GHz - 5 GHz) which can be used for passive radars as well. EHF radiation corresponds to wavelengths that range from 1 to 10 mm and therefore this type of radiation commonly receives the name of millimeter-waves.

Most radar sensors based on SHF bands have good range and range-rate features, but are heavy and bulky. These sensors are insensitive to weather conditions but have a relatively low frequency band and therefore do not provide a good angular resolution. On the other hand, radars based on EHF bands are smaller and have a finer angular resolution but have a shorter range than SHF and are more sensitive to weather conditions (Kwag and Kang, 2004).

Radars are appropriate for outdoor applications due to their relative immunity against environmental conditions such as the ability to operate irrespective of lighting conditions or overcast weather, and offer a wide range coverage. However, only obstacle detection can be done but exact reconstruction of the dimensions of the obstacle is not possible with radars due to their low output resolution.

A major challenge with radar-based detection of small UAVs is that they have a very small radar cross section (RCS). Additionally, the fact that they fly at lower altitudes and at lower speeds compared to larger aircraft also complicates its use (Fioranelli et al., 2015).

Multiple micro-Doppler effect radar systems have been proposed for UAV detection, classification and tracking in SHF bands (Molchanov et al., 2014, Fioranelli et al., 2015, Ritchie et al., 2015, Troy et al., 2015, Hoffmann et al., 2016, Nakamura and Hadama, 2017). The principle behind these radars is the characterization of the rotor blade scattering signature of the

UAV. Most of the small systems developed to be used onboard UAV showed limited detection ranges. A lightweight Doppler Radar system weighting only 230g was developed for detection and identification and integrated on a miniature UAVs (Moses et al., 2011). This radar had a detection range of 66 meters for a target with a 1 m^2 section. Research on lightweight radars with shorter ranges has been published by Fontana et al., who developed an Ultra-Wide-Band (UWB) radar in the C-band with a range of 21 meters and a weight of only 42.5 grams (Fontana et al., 2002) and by Viquerat et al., who developed a collision avoidance system in the K-band with a 10-15 meter range weighting 304 grams (Viquerat et al., 2008).

The General Atomics Predator B radar prototype (Owen et al., 2014) is another example of an active radar system on board a UAV, with good detection capabilities up to 5 kilometers for a Cessna Stationair 206 target. While having a longer range, this system had a weight over 7 kilograms and consumed 160 watts of power.

EHF or millimeter-wave radars with ranges up to a few kilometers have also been proposed for UAV detection (Kwag and Chung, 2007, Caris et al., 2015). However, these systems have been considered for detection from the ground, mainly as part of Anti-UAV systems, and may not be suitable to integrate onboard UAVs.

While active radar is one of the most appropriate technologies for aircraft detection, its main limitations come in general from its limited angular resolution and its weight and size requirements to achieve large detection ranges, which limits its applicability mostly to large aircraft. Its reliability is also limited when flying at low altitude, as echoes from other aircraft must be properly separated from ground clutter (Accardo et al., 2013). For further references on radar-based solutions, Güvenç et al. published a complete review about radar-based detection, localization and tracking methods (Güvenç et al., 2017) and more recently, Moore published a thesis about the use of radar for detection, tracking and identification in DAA applications (Moore, 2019).

LIDAR systems

Laser/Light Detection and Ranging (LIDAR) systems measure distance by illuminating a target with a laser and analyzing the reflected light. The detection ranges of LIDAR systems usually vary from a few meters to 3 km. The benefit of LIDAR lies in that it is capable of detecting non-perpendicular surfaces at high resolution and identifying objects from 5 mm in diameter to buildings. Likewise it is highly configurable and appropriate for different atmospheric conditions. The main disadvantages are the size,

weight and power requirements of these systems for long range detection.

As an example, LIDAR-based DAA systems developed at Carnegie Mellon University were able to reduce the false positive detection rate and provide accurate distance estimations of intruding aircraft (Geyer et al., 2009) up to 900 meters. Also, an interesting work by De Haag et al. (de Haag et al., 2016) explored the use of a LIDAR sensor (900 grams) and compared its performance with that of a radar sensor (854 grams). LIDAR provided a high update rate and high ranging accuracy with a detection range of 15 meters. While radar had problems at close range, it provided a longer range of 25 meters.

The main limitation from LIDAR for on board operation comes from the fact that high power consumption and weight are required for long range detection. However, recent advances in LIDAR technology have contributed to an important weight reduction in the sensors (Xie et al., 2019), which seems very promising for its use as sensing technology for near-range obstacle avoidance on board UAVs.

2.3.2. Passive sensing technologies

Since no signal transmission is required, passive sensing systems have looser requirements in terms of weight, size and power consumption, which makes them particularly interesting for applications on board small aircraft. The most relevant passive sensing technologies include: electro-optical (EO), acoustic and passive radar systems.

EO systems convert light or changes in illumination into electronic signals. While in general these sensors are sensitive to light with wavelengths between the infrared (IR) and the ultraviolet (UV) spectra, aircraft detection applications typically operate in the visible spectrum (emulating the human eye) and/or the IR spectrum. In general, EO systems alone do not provide depth information, but depth data can be obtained mainly by means of stereoscopic vision or structured light cameras, with the latter requiring additional illumination of the scene. Compared to microwaves in radar, the smaller wavelengths of visible and IR light increase the bearing precision of the detection but are less robust to obstacles or weather.

Solutions for aircraft detection in the visible, IR spectra and other vision-based approaches will be covered next, followed by a short literature review of acoustic and passive radar systems for aircraft detection.

EO sensors in the visible spectrum

Visible spectrum imaging is one of the technologies most commonly proposed for aircraft detection, mainly because of the benefits it brings by being passive (particularly for applications on board UAVs) and also due to its low cost and high availability. Additionally, in the case of DAA systems, it seems the most straightforward way to mimic the aircraft spotting capability of human pilots.

As mentioned in Section 1.2, the requirements of an aircraft detection system may vary a lot depending on the detection application. Due to its greater relevance and potential impact, the use of visual sensing for aircraft detection in DAA systems has been thoroughly researched and several attempts have been made to come up with specifications for vision-based DAA systems.

In 2007, a study by Thales UK tried to translate the concept of equivalent level of safety (ELOS) to human performance in aircraft visual detection into specific requirements (Hutchings et al., 2007). These requirements arose from the study of human pilots' vision and all the factors involved in the human task of DAA. It concluded that the minimum human reaction time of 12.5 seconds implied a visual resolution of 0.5 degrees for timely detection, which is indeed quite challenging for the human eye. The requirement of camera resolution for the timely detection of general aviation has been estimated in $125 \mu\text{rad}$ (Geyer et al., 2008). The need for a wide field of view limits image magnification, therefore requiring aircraft to be detected when they still have a very small size (subpixel to a few pixels) and no spatial features that allow for discrimination of the object. The FOV and detection range trade-offs based on the probability of an intruder aircraft being detected for various sensor configurations has also been studied in detail (Griffith et al., 2008). In general, the conclusion of these studies is that for DAA systems, visible spectrum cameras have to operate close to the sensing limits for these systems to be minimally reliable.

A complete book covering the DAA problem for UAVs was published in 2012 and included human factors, regulatory issues and navigation, control, aerodynamics and physical aspects of the problem (Angelov, 2012). A whole chapter, written by researchers from the Queensland University of Technology, was devoted to the use of on board vision for DAA. Several requirements for the DAA capability are discussed in this book, from which two have become accepted as general rules for DAA systems:

- A field of regard with an azimuth angle of ± 110 degrees and an elevation angle of ± 15 degrees (similar to that available in manned aircraft cockpits).

- A minimum horizontal miss distance of 500 ft.

Additionally, and in the same University, McFadyen et al. (McFadyen and Mejias, 2016) wrote a survey on the topic and estimated that a generally-accepted time to collision of 15 seconds requires a detection range between 1500 and 3000 m for the typical airspeeds in flights under Visual Flight Rules (VFR). This survey, based on the experience acquired over the years by the same research group (Mejias et al., 2016), also covers other interesting discussions, such as how the performance of visual detection and tracking algorithms can impact subsequent avoidance and resolution functions in DAA.

Fasano et al. wrote a very exhaustive article on the topic of DAA (Fasano et al., 2016a). In particular, they analyzed the impact of the obstacle azimuth angle and its speed in the minimum sensing range required. He concluded that visible spectrum cameras can provide a good DAA solution for small UAVs flying in non-segregated or uncontrolled airspaces and also as a complement to radar systems, in order to increase the overall angular resolution and the data rate.

According to a study on DAA systems, “human vision relies on either contrast or movement of an object in order to detect it effectively” (Hutchings et al., 2007). In terms of Computer Vision techniques, most approaches in the literature can be classified as intra-frame, inter-frame techniques or a combination of both. Intra-frame techniques are applied to individual frames, exploiting the size, shape or brightness of the target, while inter-frame techniques are applied to frame sequences, exploiting temporal or dynamic features, such as temporal changes in size, shape, position or brightness. A relevant fact to highlight is that a vision-based warning system perceives objects on collision course as relatively stationary features on the image plane (Angelov, 2012).

In terms of airborne object detection techniques, there are abundant publications in the literature (Gandhi et al., 2003). Published intra-frame techniques include: basic edge detection (Forlenza et al., 2012), max-mean and max-median subtraction filters (Deshpande et al., 1999), analysis of the signal-to-noise ratio (Minwalla et al., 2012), morphological operations (Lai et al., 2013) and cascaded learning (Dey et al., 2011). Inter-frame methods proposed include: optical flow (Recchia et al., 2007), dynamic programming (Carnie et al., 2006), particle filters (Gustafsson et al., 2002), hidden markov model filtering approaches (Lai et al., 2008), recursive ad-hoc viterbi approaches (Lai et al., 2011b) and bayesian approaches (Bruno, 2004).

Most research, particularly in the development of DAA systems, has focused on aircraft detection scenarios in which aircraft fly at altitudes at

which the sky fills the image background. This generally produces homogeneous backgrounds from which aircraft are easier to pick out. However, small UAVs typically operate at lower altitudes due to limitations in service ceilings and wireless communications, making the detection of airborne obstacles much more difficult as the aircraft gets blended with the terrain clutter. In this sense, efforts have been made in developing sky segmentation algorithms which allow the use of independent strategies above and below the horizon line. The use of this approach has led to good detection results also for aircraft detection below the horizon line (Cho et al., 2013, Molloy et al., 2017, James et al., 2019, Opromolla et al., 2019). Related to the topic and also thoroughly researched, but outside the scope of this thesis, is the aerial detection of aircraft on the ground for surveillance purposes, which is most commonly conducted using visible spectrum satellite images (Zhang et al., 2016).

Another topic closely related to visual detection is visual tracking. Trackers have been proven useful not only for predicting the position of an object in the next frame, but also for confirming or discarding potential detections. For this reason, the boundary between obstacle detection and tracking may sometimes be blurred in the literature in the case of aircraft detection, as many vision-based DAA approaches rely on techniques that try to track aircraft before detecting them. However, there is specific research in the literature devoted only to aircraft tracking (Mian, 2008, Ali et al., 2013, Sandmann et al., 2014). This thesis focuses on aircraft detection using one or more frames rather than in specific tracking methods which deserve their own area of research.

In summary, besides the aforementioned advantages of being a passive sensing technology, the use of visible spectrum cameras provides good range and angular resolution and represents the most straightforward adaptation from the human DAA capability. The main drawbacks of using visible spectrum sensors for aircraft detection are that: (1) sensors may not be effective in the case of weather uncertainty (e.g., smoke, fog, dust) or for the detection of aircraft flying under the horizon line because of the cluttered backgrounds; (2) arrays of sensors are required to achieve a wide field of view with adequate angular resolution; (3) for the application in DAA systems, information about the target, such as range or target size cannot be obtained directly; and (4) the high resolution generally required for aircraft detection comes with a high computational cost, which might be hard to deploy in small UAV systems.

EO sensors in the thermal infrared spectrum

Sensitive to heat, radiated at wavelengths between 3 and 15 μm , thermal infrared cameras are another common technology for aircraft detection. Flying objects typically emit a stronger heat signal than their surroundings, facilitating their detection in this portion of the spectrum. While thermal infrared imaging sensors have typically lower resolutions compared to visible spectrum sensors, they provide better visibility under light rain, snow or fog, under extreme lighting conditions and at nighttime. The size and cost of these sensors has experienced a great reduction in the last years, allowing them to be carried on board small UAVs.

Research for the detection of aircraft in the thermal infrared has been active for decades, due to its interest for missile guidance. One of the main challenges is to characterize the image background in order to filter it out and the use of morphological operations has been the main approach used. A morphology-based algorithm for low signal-to-noise ratio (SNR) target detection based on prior knowledge of target size demonstrated real-time detection capabilities and the ability to deal with background clutter and with the noise typically present in thermal images (JiCheng et al., 1996). Another study applied morphological applications in both vertical and horizontal direction, with good detection results on real IR images with clouds and ground objects as backgrounds (Zhu et al., 2000). A genetic algorithm was proposed to learn the shape and structural characteristics of morphological filters for target detection in IR images (Yu et al., 2003). Both neural networks and a genetic algorithm were used to obtain optimal top-hat morphological filters for infrared target detection, providing robust adaptability in low SNR images (Zeng et al., 2006). Wavelets have also been proposed to detect moving single-pixel targets in a sequence of infrared images. An algorithm using wavelets focused on the detection of targets in airborne applications, with a large movement from one frame to the next (20 to 30 pixels per frame with a frame rate of 1 frame/s) and with clutter and occlusion due to moving clouds (Zaveri et al., 2007). The nighttime detection of UAVs has also been the object of specific research (Andraši et al., 2017).

In another published work, targets initially detected using window intensity variance with a sliding window approach were tracked and their angular velocity was estimated using an adaptive noise variance estimation algorithm and Kalman filters. The proposed collision warning system was based on low-resolution infrared images (320×240 pixels) captured from cameras mounted on board an aircraft. It also proposed the use of an IMU for image stabilization. The system triggered an alarm whenever angular velocity of tracked targets was small enough ($< 3\sigma$) (Osborne et al., 2011).

Chapter 2. State of the art

This is because, as in the case of visible spectrum images, objects on collision course are perceived in infrared images as nearly stationary features on the image plane.

Horizon detection has also been applied to thermal infrared images, but to the author's knowledge not as an intermediate step for aircraft detection, but mainly for attitude estimation (Bazin et al., 2009). This work explored the use of a dynamic programming algorithm on images captured using catadioptric mirrors and presented horizon estimation results based on synthetic images. The use of thermopiles has also been proposed for attitude estimation (Taylor et al., 2003). With respect to thermal imaging datasets, there are few public IR datasets for aircraft detection and those available also include visible spectrum images (Wang et al., 2019, CVPR, 2020).

Several works have focused on tracking methods, but mainly for airborne infrared target tracking in combat environments and for military purposes (Wu et al., 2019). As in the case of research for aircraft detection in the visible spectrum, tracking methods deserve their own specific research and fall out of the scope of this thesis.

Again, as with visible spectrum imagery, Deep learning methods applied to thermal infrared images have raised a lot of research interest in the last few years. Generative Adversarial Networks (GANs) have been proposed for synthetic image generation, Convolutional Neural Networks (CNNs) have been used for object detection and Siamese Networks have been proposed for aircraft detection and tracking (Wang et al., 2019, Kim and Ko, 2020).

In summary, thermal infrared cameras do not require visible light for correct operation, making them the best choice for nighttime and/or for operation under certain meteorological conditions. However, they have lower resolutions than visible spectrum cameras, which limits their effective range for the same FOV. Their cost is also quite higher and measurements are noisier.

Other vision-based technologies

Other relevant vision-based technologies which require separate mention are depth cameras and event-based cameras. Although they are mentioned together here, these technologies are very different from each other. Depth cameras are cameras which, besides visible or infrared pixel intensity values, can also provide depth information for each pixel. Depth is generally obtained using structured light projectors (turning the camera into an active sensing device) or additional cameras, typically through stereoscopic vision. Event cameras are a totally different type of device with quite uni-

que properties. While visible or infrared cameras generally have latencies in the order of tens of milliseconds, event-based cameras output pixel-level brightness changes at the time they occur, with theoretical latencies of micro to nanoseconds. Instead of streaming frames at constant time intervals, each pixel operates at its own frequency, firing an event every time there is a brightness change in it. Since event cameras are in a sense motion activated, they are particularly well-suited for obstacle detection.

Event-based cameras have been proposed for dynamic obstacle detection and the role of perception latency in high-speed DAA has been analyzed in a recent publication (Falanga et al., 2019). The maximum latency an aerial robot can tolerate to guarantee safety has been proven to be related to the desired speed, the maximum acceleration of the vehicle and the sensing range. The authors of this work ran several experiments in which a UAV was able to avoid a ball thrown to it as speeds up to 9 ms^{-1} .

3D point clouds have been recently proposed for having UAVs autonomously execute obstacle avoidance maneuvers using a depth camera (Lopez and How, 2017). In this work, point clouds are sampled and converted into octomaps and a very fast planning algorithm is used to plan collision-free trajectories. However, this work has focused on the detection of static obstacles only.

While the research of event-based cameras for collision avoidance is still emerging and their cost is still quite high (2000-5000 €), they are a promising approach for high-speed, short-range collision avoidance. Depth cameras are definitely more affordable for the same purpose, but their use for aircraft detection has not been much explored in the literature.

Acoustic sensing techniques

The acoustic pressure level emitted by engine-powered aircraft generally comprises a strong narrow-band tone imposed onto a broadband random component. Based on this, surveillance systems can rely on microphones for UAV detection and UAVs equipped with acoustic sensors can detect intruding aircraft.

Several forms of microphones are available. In order to provide information about the bearing of an intruder aircraft, directional microphones are needed. Two basic types of microphones are available, pressure-operated microphones, with omni-directional detection and pressure-gradient microphones, which have a different sensitivity depending on the incoming sound direction. The best angular resolution that can be obtained by acoustic sensors is around 15 degrees, with theoretical maximum distances ranging from 900 m for the detection of a piston-engine aircraft to 9000 m for a jet-engine

aircraft (Fasano et al., 2016a). Nevertheless, accurately estimating range is extremely challenging due to its dependence on the thermodynamic properties of the air and also to the potential presence of several unwanted effects such as echoes, distortions and non-linearities.

In the case of DAA systems, a proper installation of the sensors on board the aircraft must be considered in order to minimize the disturbance generated by the own aircraft or other unwanted sources. Moreover, the installation should protect the microphones from rapid degradation from the engine exhaust or from dirt collected during landing and takeoff.

Various sets of microphones have been integrated in acoustic arrays on different systems to locate and track aircraft. A low-cost ground-based system consisting on 24 microphones was tested with real flights and demonstrated the capability to capture the path followed by a remotely controlled UAV. No details about the type of UAV or the detection range were reported (Case et al., 2008). Another ground-system based on four microphones was tested by placing various fixed and rotary wing model aircraft engines between the microphones to simulate the acoustic signature of the host UAV while letting various “intruder” manned aircraft fly above the system. Authors reported their system was able to track the aircraft up to 1500 m (Finn and Franklin, 2011). More recent research has demonstrated that a gasoline-powered UAV could be detected from another UAV at distances up to 678 m using an specific harmonic spectra transform. According to the authors, the detection range of their system was more than double the minimum required to avoid a head-on collision (Harvey and O’Young, 2018).

The sound emitted from a flying aircraft is received in a stationary microphone through multiple paths (e.g. directly from the source and reflected from the ground or other objects). The exploitation of this effect to obtain instantaneous estimates of various flight parameters of the aircraft emitting the sound has also been object of research. In particular, a technique called Cepstrum tracks has been proposed to estimate aircraft speed along with the time, horizontal distance and height of the closest point of approach (Gao et al., 2002, Lo et al., 2003). This technique has also been proposed for aircraft classification (Sendt et al., 2002). However, this technique shows limitations in the presence of air turbulence or with certain types of grounds, such as sand.

There is also specific published research about UAV surveillance systems. For example a distributed system to identify the appearance and the approximate position of UAVs using a wireless acoustic sensor network and machine learning algorithms has been recently published (Yue et al., 2018).

In summary, acoustic sensors are a highly cost-effective solution capable of determining rough estimates of aircraft speed and bearing and in some cases even approximate position information if a multi-microphone array system is properly deployed. In general, the detection performance of acoustic sensing systems can be poor depending on the type of aircraft and their performance can be greatly deteriorated by atmospheric conditions, such as severe wind. However, while these systems can operate day and night, they do not have the long-range capabilities of radar or EO systems. Furthermore, for DAA applications, acoustic sensing-based systems show some limitations due to the complexity of filtering the sounds generated by the UAV itself and because they cannot be used for collision avoidance of silent aircraft such as gliders or balloons. For these and other reasons, while they can be deployed as part of anti-UAV systems, acoustic systems are not considered a primary sensing device for DAA systems but as an auxiliary low-cost solution.

Passive radar systems

An alternative to active radar with a lot of potential is passive radar, which uses the reflections of RF signals already existing in the environment (e.g. radio stations, mobile telephony networks or digital TV) to detect the presence of other flying objects. Usually, the ultra high frequency (UHF) band, with frequencies between 300 MHz and 1 GHz, is used. Its main drawbacks are its high complexity and the need to rely on third-party emitters.

Orthogonal Frequency Division Multiplexing (OFDM) is a technique that allows to increase the spectrum efficiency and reduce the multi-path effect in wireless signals. It has been recognized as the core of the 4th generation of wireless communication. Due to the availability of these signals, they have been proposed for aircraft detection by means of an array of ground-based antennas (Yang et al., 2016).

Digital audio broadcast (DAB) signals have also been proposed and researchers have demonstrated that it is possible to detect micro-UAVs up to a distance of 1.2km from the receiver with a single antenna and a two-channel receiver (Schüpbach et al., 2017). Another solution based on a single receiver antenna and relying on digital television signals has been proposed in order to detect aircraft or birds flying at low altitudes and slow speeds. Multiple antennas can be used to locate and track targets (Fang et al., 2018).

Passive radar systems have desirable characteristics for on board aircraft detection (low weight and power consumption) and have demonstrated potential for ground-based, stationary systems. However, the technology is still in its infancy and a lot of research is still required to improve its re-

liability, to come up with better ground clutter cancellation techniques as well as ways to reduce its computational complexity. Its application using moving antennas, such as on board aircraft is also quite limited, although some preliminary results have already been published (Dawidowicz et al., 2012).

2.4. Hybrid systems

Unfortunately, no single sensing technology has suitable properties for all the different aircraft detection applications and requirements. Detection range, day/night availability, meteorological conditions required, integrability on board small UAVs and velocity estimation capabilities are some of the factors to take into account when choosing the right sensing approach for aircraft detection. For this reason, some authors in the literature have explored the integration of various technologies to complement each other and extend systems' capabilities.

One of the first theoretical approaches to the combined use of various sensing techniques was proposed for DAA and relied on radar and both visible and thermal infrared cameras (Fasano et al., 2008). Another theoretical system combining collaborative ADS-B sensing with both visible and thermal infrared images was validated through simulation. The system used EO sensors to detect potential intruders and ADS-B when available to obtain information about them (Ramasamy et al., 2014). Other theoretical approaches have also been published (Jackson et al., 2015, Ramasamy et al., 2016). Theoretical sensor fusion approaches for anti-UAV systems have also been proposed (Sturdivant and Chong, 2017).

A ground-based system has been proposed as a first approach to a low size, weight and power DAA system with aircraft classification capabilities (Barott et al., 2014). This system used passive radar to determine relative distance and orientation of the aircraft and EO sensors to confirm and classify the aircraft: visible spectrum images were processed to extract aircraft size and shape while thermal images were used to determine the thermal characteristics of the propulsion system of the detected aircraft. Features from both EO sensors were used to build multi-spectral features, which were classified in real-time with a support vector machine(SVM).

Fasano et al. proposed the use of data fusion to complement the fine angular accuracy of EO sensors with active radar's range and range rate estimates, thus providing a statistically significant increase of the level of situational awareness, compared with standalone radar systems. The performance of EO systems alone has great sensitivity to environmental condi-

tions and it cannot provide reliable range or range rate estimates, requiring the use of over-conservative approaches for conflict detection (Fasano et al., 2016b).

A stationary anti-UAV system combining optical and acoustic sensors demonstrated UAV detection capabilities with a maximum detection range of 300 m (Christnacher et al., 2016). A similar approach using 30 visible spectrum cameras and 3 microphones demonstrated good performance, detecting even occluded UAVs (Liu et al., 2017). Another publication combined visible and thermal infrared images for the same purpose, using a visible spectrum camera for locating potential threats and then tracking and classifying them using a zoomed thermal camera mounted on a gimbal (Unlu et al., 2019).

While complementing the advantages of various sensing technologies might be a good solution to improve performance in anti-UAV systems and even in DAA systems for large UAVs, the weight and power consumption of hybrid solutions are still serious limitations for their use on board small UAVs.

2.5. Specific areas of research in Computer Vision

This section covers in higher detail the state of the art of specific research areas in Computer Vision relevant to the topic of aircraft detection. The first is synthetic image generation, a key topic today in object detection and particularly in deep learning-based approaches. Then a review of the state of the art of horizon line detection and sky segmentation techniques is presented. Finally, the most relevant learning-based approaches for real-time object detection are covered.

2.5.1. Synthetic image generation

The availability of adequate image datasets is essential for developing and evaluating vision-based aircraft detection systems. Particularly in the case of DAA systems, obtaining images of an aircraft captured from another aircraft is a task that involves certain complexity. Furthermore, when dealing with collision course scenarios, obtaining this type of images becomes a risky challenge. This is not only a difficulty for the development of vision-based aircraft detection systems but also for the standardization of system requirements, as no common benchmarks are available to objectively evaluate and compare these systems.

One of the first video datasets containing 23000 frames of flight sequences was made available to the community free of cost back in 2008. However, the usefulness of these images for DAA systems is limited since all videos were recorded from the ground (Mian, 2008). Since then, various authors have described different vision-based collision avoidance systems for which real flights were performed in order to collect useful data. However, none of those datasets has been made available for further development by the research community (Mejias et al., 2010, Lai et al., 2011b, Lai et al., 2011a, Dey et al., 2009). Currently some specific internet platforms can be found from which real aircraft imagery and videos can be downloaded such as AirTeamImages (AirTeamImages, 2003) or even Youtube (Youtube, 2005). However, most the imagery provided in these websites is not intended for research purposes and therefore many useful features are not available, such as geo-referenced or annotated image(s) or the possibility of searching for images using relevant criteria: type of background, aircraft trajectory, etc.

One of the first attempts to try to standardize datasets was the work of Rozantsev et al., who built and published a dataset of aircraft images captured from a moving camera and including both manned and unmanned vehicles. They also proposed an algorithm for aircraft detection based on appearance and motion cues (Rozantsev et al., 2016). The *Drone-vs-Bird Detection Challenge* started in 2017 as part of the IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS). The challenge was aimed at detecting and discriminating birds and drones in videos captured from static, visible-spectrum cameras placed on the ground, with a focus on border security. In the second edition, the dataset from the first edition was extended with additional footage (Coluccia et al., 2019). A similar dataset was recently published by researchers from the University of Southern California (Wang et al., 2019). Another challenge specifically targeted to UAV detection both in the visible and thermal spectra was the Anti-UAV Challenge organized around the Computer Vision and Pattern Recognition Conference (CVPR, 2020). Other similar datasets have been recently published in Kaggle (Aksoy, 2019, Ozel, 2019).

The lack of adequate images has also forced researchers to explore the use of simulated/synthetic images and data augmentation (Forlenza et al., 2009, Fasano et al., 2013, Zsedrovits, 2014, Kang et al., 2017, Wang et al., 2019). Some authors have used the well-established flight simulation software FlightGear (FlightGear, 1997) to obtain synthetic flight imagery for collision avoidance and estimation of the attitude of the intruder aircraft (Zsedrovits et al., 2011, Zsedrovits et al., 2013). Today, powerful options to create highly-realistic images can be found in the market, such as Microsoft's

AirSim (Shah et al., 2017) and Unreal Engine (Epic Games., 1998).

The use of rendering engines, which can synthesize high quality images, together with data augmentation techniques, are capable of easily producing valuable datasets with thousands of samples. This has nudged the use of Deep Learning approaches, which have demonstrated great results and are currently a very active area of research, with new algorithms being continuously published and more and more powerful parallel computing devices being produced every year. Several publications can be found which propose the use of deep neural networks for aircraft (Hwang et al., 2018) and UAV detection (Saqib et al., 2017, Aker and Kalkan, 2017, Chen et al., 2017, Schumann et al., 2017, Wu et al., 2018, Opromolla et al., 2019). Semantic segmentation has also been proposed to differentiate manned aircraft (James et al., 2018) and UAVs (Craye and Ardjoune, 2019) from the background.

The lack of standardized, high-quality imagery of collision scenarios for the development of DAA systems has been one of the problems addressed in this doctoral research and is covered in Chapter 3.

2.5.2. Sky segmentation and horizon line detection

Many research efforts have focused on approaches for sky segmentation and horizon line detection, due to their relevance for visual geo-localization and port security, among other applications. In the field of UAVs, these techniques have been mainly proposed for attitude estimation purposes. In the case of aircraft detection, they can be used to filter out potentially undesired ground clutter and also to establish specialized algorithms, strategies or models for the detection above and below the horizon line. While not directly applied as an intermediate step for aircraft detection, a wide range of approaches for sky segmentation and horizon line detection using vision have been published, mainly based on three different methods: (1) the classification of sky and non-sky image regions using machine learning techniques (Todorovic et al., 2003, McGee et al., 2005, Fefilatyev et al., 2006, de Croon et al., 2011, Boroujeni et al., 2012, Yazdanpanah et al., 2013, Verbickas and Whitehead, 2014, Yazdanpanah et al., 2015), (2) edge detection techniques, (Lie et al., 2005, Zafarifar et al., 2008, Kim et al., 2011, Ahmad et al., 2013), and (3) classification of pixels belonging to the horizon line (Hung et al., 2013, Ahmad et al., 2015a, Porzi et al., 2016). A doctoral thesis on the combined use of omnidirectional vision and polarization vision was published which includes a comprehensive review of the available algorithms for vision-based horizon line detection (Shabayek, 2012). Also, a comparison of semantic segmentation approaches for horizon/sky line detection has been published (Ahmad et al., 2017) and can be looked up for further references.

In 2013, Todorovic et al. proposed a Hidden Markov Tree (HMT) model trained with an expectation maximization algorithm using color and texture features, obtaining good results in terms of processing speed (120 ms per 640×480 frame on an AMD Athlon processor at 1.8 GHz). Color features were based on H and I channels from the HSI color space, while texture features were extracted using a complex wavelet transform (Todorovic et al., 2003). McGee et al. proposed a method combining machine learning techniques with morphological operations. The machine learning method was based on an SVM classifier trained using YCrCb color space features. After the division of the image into sky and non-sky regions, morphological operations and the Hough Transform were utilized for horizon line detection (McGee et al., 2005). Fefilatyev et al. compared the performance of three machine learning classifiers (SVM, J48, and Naïve Bayes) that were trained and tested using 10 images for the purpose of classifying pixels as sky or ground, assuming a linear horizon boundary. Two experiments were conducted. The first used similar images for training and testing, and the second one used images from different cameras, lighting conditions and time of day. Results showed that the SVM classifier performed most effectively (Fefilatyev et al., 2006). Croon et al. extended the features extracted in this work by including cornerness, grayness and Fisher discriminant features to train a J48 decision tree classifier for obstacle avoidance applications (de Croon et al., 2011). Boroujeni et al. presented a pre-processing stage using a Gaussian filter previous to a clustering stage. Two clustering algorithms were tested, one based on an intensity-based technique and another one using k-means. Results showed processing times of 1.5 seconds per frame for the intensity-based technique and 10 seconds per frame using k-means in MATLAB (Boroujeni et al., 2012). More recently, Verbickas et al. proposed the use of a Convolutional Neural Network trained over a dataset of 13,687 images to perform the semantic segmentation of the sky (Verbickas and Whitehead, 2014). The approach of Yazdanpanah et al. was based on a real-time horizon line detection method which combined clustering with k-means and classification using a 3-layered neural network trained with 9×9 non-overlapping blocks of the image. For evaluating the proposed algorithm, 1038×1388 resolution images were used, showing that the fastest implementation using a parallel algorithm on a GPU could reach 1014 ms per frame (Yazdanpanah et al., 2015).

Zafarifar et al. proposed a horizon line detection method based on the combination of color and edge-based algorithms. The edge-based algorithm computed the most prominent line in the image using a canny edge detection and the hough transform, whereas the information of color, texture,

position and gradient using the YUV color space was taken into account in the color-based algorithm (Zafarifar et al., 2008). A similar approach using machine learning techniques was published a few years later. It consisted on extracting Canny edges, computing various descriptors based on color, illumination and location, and classifying them in order to detect those belonging to the horizon line (Hung et al., 2013).

The strategy proposed by Ahmad et al. consisted of a horizon line detection approach based on a classification map, which classified 16×16 pixel normalized image patches as belonging or not to the horizon line using several classifiers, such as SVMs or Convolutional Neural Networks. A training set consisting of 9 pictures was used, from which 343 positive and negative examples were extracted for each picture (Ahmad et al., 2015a). In another work, the same author presented a detailed comparison between two machine learning-based approaches for horizon line detection, edge and edgeless-based (Ahmad et al., 2015b). Additionally, the authors propose a fusion strategy, combining both approaches for better results. Deeply-Supervised Deconvolutional Networks have also been proposed for horizon detection in mountain pictures with an improvement with respect to CNNs. In this work, real-time performance is not regarded and the approach is evaluated in terms of accuracy only (Porzi et al., 2016).

In terms of solutions published specifically for attitude estimation, Ettlinger et al. implemented one of the earliest horizon line detection algorithms, using differences in the color distribution between the ground and sky regions and optimized it for real-time performance (Ettlinger, 2001). In their work, Bao et al. also proposed the use of classical computer vision-based methods for horizon line detection by means of a Laplacian of Gaussian to enhance edges and suppress noise, followed by a binary thresholding operation to extract the main lines from the image. Finally a orientation projection algorithm was used to determine the horizon position using a technique similar to a Hough transform. Information from prior frames was used to reduce the search range over the image and improve the algorithm's speed (Bao et al., 2003). A similar approach was followed in a posterior work by other authors, but using independent morphological filters for each image channel (Dusha et al., 2007). Another work proposed a color space transformation in order to enhance the differences between the sky and ground regions, then applied a binary thresholding operation using the Otsu method and finally computed the ground line based on the position of the centroids of the sky and ground regions (Cornall et al., 2006). In the work published by Thurrowgood et al., three steps were followed for extracting the horizon: enhancing sky/ground contrast, determining the optimum threshold

for sky/ground segmentation and then fitting the horizon to the boundary between both areas (Thurrowgood et al., 2009). Another approach relied on optical flow and used a basic image processing approach based on edge detection techniques to detect horizon line candidates and optical flow to reject false detections (Dusha et al., 2011).

Several authors have proposed the use of catadioptric or panoramic cameras for attitude estimation based on horizon detection. These cameras provide wide-angle views, in which the horizon does not appear as a straight line but with an elliptical curve shape. Natraj et al. published a review of several methods for horizon detection and other applications of catadioptric cameras (Natraj et al., 2013). One of the first works using these cameras for horizon detection used a color space transformation followed by a pyramid segmentation and a gaussian adaptive thresholding operation in order to obtain a sky-ground binary image. Finally the contour from this binary image was extracted and an ellipse was fitted to obtain the horizon (Mondragón et al., 2010). An adaptive approach was proposed by Moore et al., in which a classifier maintained a record of the spectral and intensity properties of the sky and ground regions, which was continuously up-dated online (Moore et al., 2011a). In another work, the horizon line was extracted using a UV-filtered panoramic vision sensor. This method required a sophisticated camera system including a combination of a CCD image sensor, an ultraviolet filter and panoramic mirror lens (Tehrani et al., 2016).

RANSAC-based line clustering has also been proposed to find the outliers among all possible horizon edges in an image. This approach did not rely on the detection of the horizon directly, but rather on its estimation from building regularities in urban scenes (Hwangbo and Kanade, 2011). Grelsson et al. proposed a horizon estimation method which incorporated a probabilistic Hough voting scheme, in which edge pixels on the image were weighted in the voting, based on how likely they were to be a horizon edge pixel given the probability distributions for the camera altitude and the pitch and roll angles. Prior to the Hough voting, the image was pre-processed using conventional methods (Grelsson and Felsberg, 2013, Grelsson et al., 2013). Other authors have proposed color channel-wise smoothing, horizon edge extraction using gradients followed by a refinement stage and the use of an m-estimator to fit a line to the extracted profile by using iterated re-weighted least squares minimization (Dumble and Gibbens, 2012).

Several other researches for horizon detection have also been carried using digital elevation models. The main disadvantage of this approach is that these methods require a pre-generated terrain model in order to detect the horizon and estimate the attitude and typically fail to provide real-

time performance. An early example of this approach was a method that estimated the position and orientation of a camera using a single image of a mountain, by aligning the terrain skyline with a synthetic skyline generated from a digital elevation model (DEM) (Naval Jr et al., 1997). In the work by Gupta et al., a very simple algorithm sought to find the horizon by searching down each pixel column of the image for a certain intensity change. The edge so extracted was smoothed using a second order butterworth low-pass spatial filter to remove noise. The main focus of this work was not the horizon detection itself, but how it matched a rendered DEM in order to estimate position and attitude (Gupta and Brennan, 2008). In another approach, a Canny edge detector and a probabilistic Hough voting scheme were used to estimate the camera attitude and the corresponding horizon line in the image. The attitude estimates were refined through registration of the extracted horizon pixels with the geometrical horizon from a DEM. The approach was tested with a fish-eye lens camera with high resolution images (30MP) (Grelsson et al., 2015).

Less literature is available about horizon detection in infrared images. Some research was published by Bazin et al. on the detection of the horizon using catadioptric mirrors, however the authors did not use real images but manually-synthesized catadioptric images (Bazin et al., 2009). Another work relied on infrared images, digital elevation maps and an altimeter, however it did not extract the horizon from the images, but only mountain peaks, which were later matched with the elevation map and allowed to estimate the position and attitude of the camera (Woo et al., 2007).

In general, the main drawback of the reviewed techniques is the high computational cost. Few of the aforementioned authors have paid attention to real-time performance, not even in the works for attitude estimation. A trade-off between quality and performance has been considered in the sky segmentation and horizon detection algorithms proposed in this thesis work.

2.5.3. Real-time object detection

Object detection is a common problem in Computer Vision, aimed at detecting instances of semantic objects of a certain class in an image. Classical methods have generally tackled this problem through one or more of the following stages:

- Pre-processing, which consists on transforming an image with the purpose of improving its features, facilitating posterior stages.
- Segmentation, which aims at finding meaningful regions in the image.

- Description, through which regions of interest are parameterized.
- Recognition, which compares the extracted values with a knowledge base.

In general, classical methods perform well when very characteristic features can be extracted for the object of interest. However, these methods are complex to implement and fail to provide robust detections in unstructured scenarios due to the great domain variability. In contrast, learning-based approaches trained on meaningful data can provide solutions straightforwardly and with greater robustness to changes in illumination, background, type of object, etc. As a drawback, learning methods and especially deep learning ones, usually require large amounts of data to perform well.

In the following paragraphs, some of the most relevant learning-based algorithms for real-time object detection are briefly reviewed.

The Viola-Jones algorithm

One of the first learning-based frameworks for real-time object detection was the Viola-Jones algorithm, published in 2001 (Viola and Jones, 2001). It was applied with great success to the problem of face detection and is still commonly used today. The approach is based on training a cascaded classifier, which once trained, can be used to detect objects in unseen images.

The classifier is “cascaded” as it consists on multiple stages of simpler classifiers or stumps, which are applied successively to each region of interest until some stage rejects it or all stages are passed. The object is detected only if the outcome is positive for all the classifiers applied to a given region of interest. The algorithm relies on Adaboost, which determines the most relevant features and builds an optimized cascade of decision stumps (decision trees with a single level). Haar-like features are used in these weak classifiers due to their calculation speed, achieved with the use of integral images. In order to deal with the variability in size of objects, the scanning of the image is done at different image scales.

R-CNN, Fast R-CNN and Faster R-CNN

A method that initiated an entire family of object detectors is R-CNN (Girshick et al., 2014). This algorithm operates in three stages: first, region proposals are generated using the selective search algorithm (Uijlings et al., 2013), then, a convolutional neural network (CNN) extracts features from each region independently, and finally an SVM with a linear kernel is used to classify them.

Beyond, Fast R-CNN proposed in Girshick, 2015 was conceived for addressing the low speed of R-CNN. In Fast R-CNN, the network first processes the whole image with several convolutional and max pooling layers to produce a convolutional feature map. Then, for each region proposal, another max pooling layer extracts a fixed-size feature map which is then mapped to a feature vector by fully connected layers. The network has two output vectors per region of interest: softmax probabilities, which indicate the most likely class and per-class bounding-box regression offsets, used for refining the detection.

Built on top of Fast R-CNN, Faster R-CNN (Ren et al., 2015) proposed to replace the selective search algorithm by a Region Proposal Network (RPN). RPN is a fast fully-convolutional network, used for efficiently predicting region proposals with a wide range of scales and aspect ratios. For this, bounding boxes called “anchor boxes” that serve as references at multiple scales and aspect ratios were used. The region proposals predicted by the RPN were later processed in a similar way to previous approaches.

Single-shot detector (SSD)

The single-shot detector model (Liu et al., 2016) further increased the speed of Faster R-CNN. While the family of R-CNN models perform the detection process in two stages: region proposal and region classification, SSD tries to predict bounding boxes and their corresponding classes in a single stage.

In order to do this, the image is fed to a series of convolutional layers, producing several sets of feature maps at various scales. Then, a small set of anchor boxes with different aspect ratios and obtained from the training data is evaluated at each location and for each of the feature maps. For each anchor box, both shape offsets and confidences for all object categories are predicted. The model loss is a weighted sum between localization loss and confidence loss, to try to optimize both classification and localization accuracies.

You only look once (YOLO)

As in the case of SSD, YOLO tries to perform object detection and classification in a single stage. YOLO is one of the most popular and currently one of the fastest and most accurate real-time object detection methods (Bochkovskiy et al., 2020).

In its original implementation (Redmon et al., 2016), the input image was fed through a CNN, partially based on the popular architecture Goo-

gLeNet, which predicted a tensor encoding multiple bounding boxes and their corresponding confidences. The image was divided into regular grid cells, and a fixed number of bounding boxes and confidences was predicted for each cell. A non-maxima suppression algorithm was used to filter undesired bounding boxes.

Several evolutions of the model have been proposed: YOLOv2 (Redmon and Farhadi, 2016), YOLOv3 (Redmon and Farhadi, 2018) and YOLOv4 (Bochkovskiy et al., 2020). These models have incorporated several features, such as the ability of the model to adapt to inputs of different sizes by resizing the model during training (YOLOv2), the ability to predict bounding boxes at multiple scales (YOLOv3) and the use of data augmentation based on adversarial attacks (YOLOv4). The incorporation of these features has kept YOLO at the top of the real-time object detection benchmarks, systematically providing unprecedented levels of speed and accuracy.

2.6. Conclusions

As shown in Table 1.1, the various applications of aircraft detection have heterogeneous requirements and it is hard to find a single sensing technology or a combination of them which perfectly fits each of them.

The existing sensing technologies have various advantages and drawbacks, which are briefly indicated in Table 2.1. VMC refers to visual meteorological conditions, a term used in aviation which basically refers to good visibility conditions for the human eye. When VMC are not present, then instrument meteorological conditions (IMC) apply. Cooperative sensing systems (TCAS and ADS-B) can be used under all-weather conditions. However most of them typically have size, weight, power and cost limitations. Active radar can be used under most meteorological conditions, but it has limited angular resolution and has size, weight, power and cost limitations. As for LIDAR, it is an easy technology to configure, which can work at under VMC and IMC, but it has size, weight, power and cost limitations and limited range.

The main advantages of non-cooperative passive technologies (acoustic, visible and infrared EO sensors) are the low cost, size, weight and power and ease of integration on board small UAVs and the ability to detect traffic not equipped with transponders, including birds and silent aircraft (except for acoustic sensors). Prime disadvantages include the inability to measure direct range and poor performance in IMC. EO sensors decrease their performance when visibility conditions are not good and also in the presence of cluttered environments, such as below the horizon line. Thermal infrared

sensors have little resolution and therefore have notable range limitations as compared to visible spectrum ones. Also, there is still little research on aircraft detection using depth and event-based cameras.

According to the literature and although it is still an unsolved problem conditioned by multiple factors, the best general choice for DAA systems appears to be the combination of radar, complemented with visible spectrum cameras, which provide the angular resolution that radar lacks. For UAV swarms, depth cameras and small LIDARs seem to be the best solution according to the solutions proposed in the literature. A lot of sensing options are available for anti-UAV systems, depending on the reliability requirements: LIDAR, active and passive radar, thermal EO sensors and acoustic sensors. Commercial systems typically rely on more than one sensing technology. Finally, aircraft detection for remote sensing purposes seems to be best solved using visible spectrum cameras.

In aircraft detection, as in general object detection, Computer Vision has been generally applied in the research community with specific objectives in each publication and only recently have research goals started to become put in common and standardized. In general, each group of researchers has had to collect their own dataset in order to validate their approach, not allowing others to continue the research from the same endpoint. Only recently with the rise of deep learning have image datasets been commonly shared in the community, in order to be used as common standards and allowing researchers to systematically improve the performance of detection algorithms.

This thesis tackles some of the problems which arise from the application of vision in aircraft detection for UAVs: the need for standardized datasets and benchmarks and the advantages of simulation, the applicability of vision-based technologies according to the scenario, particularly on board small UAVs, and the use of novel deep learning approaches to the problem of aircraft detection.

	Information provided	VMC	IMC	Size, weight, power, cost	Max range	Others
ADS-B	Position Altitude Velocity	✓	✓	×	240 Km	Well proven
TCAS	Range Altitude	✓	✓	×	160 Km	Well proven Widely used
Active Radar	Range Bearing Speed Azimuth	✓	✓	×	35 Km	Limited angular resolution
Acoustic sensors	Elevation Velocity Range	✓	✓	✓	9 Km	Lack of direct range Delay
Passive radar	Bearing Speed Azimuth	✓	✓	✓	5 Km	Complex Limited applicability
Visible EO	Elevation Azimuth	✓	×	✓	5 Km	Lack of direct range
Thermal Infrared EO	Elevation Azimuth	✓	✓	✓	3 Km	Lack of direct range
LIDAR	Position Velocity	✓	✓	×	1 Km	Easy configuration

Table 2.1: Comparison of sensing technologies for aircraft detection, sorted according to their approximate maximum range. VMC and IMC refers to the applicability under visual and instrument meteorological conditions, respectively.

Chapter 3

Aircraft detection in the visible spectrum

3.1. Introduction

As covered in Chapter 2, there are two aspects relevant to the use of visible spectrum cameras for aircraft detection in the context of DAA systems which have not received enough attention by the research community. The first one is the lack of image datasets with collision course scenarios. This chapter aims to contribute to the research in this field with a software for the generation of synthetic images of aircraft encounter scenarios which has been proved useful for the development of DAA systems. The second one is an approach for sky segmentation in aerial images with a focus on real-time performance. Since aircraft detection over cluttered ground backgrounds is harder than over the sky, segmentation of the sky region offers the possibility of having dedicated strategies for aircraft detection above and below the horizon line.

Additionally, this chapter aims at coming up with a very efficient approach for aircraft detection in visible spectrum images based on contrast feature extraction, which is validated by comparing it with some of the fastest state-of-the-art object detection algorithms.

3.2. Synthetic image generation

As already mentioned in the previous chapter, the availability of adequate image datasets is essential for developing and evaluating vision-based aircraft detection and tracking algorithms. However, these images are in general scarce and complicated to obtain.

In mid-air collisions, aircraft appear to pilots as stationary features through the windscreen (ATSB, 1991). Automated 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 for the research and development of DAA systems to use flight imagery in which collision-course encounters are simulated, since obtaining real equivalent images is a complex and dangerous issue.

Computer simulated imagery has only recently been able to provide levels of realism equivalent to those of real images. Alternatively, blending real imagery with rendered objects of interest provides an interesting middle-ground solution. However, in order to have a realistic appearance, rendered objects must be correctly accommodated to the real scene they will be blended to. The challenge is therefore to take into account optics, illumination and camera motion to produce a realistic output.

With the techniques presented in this section, a 3D simulated aircraft is firstly positioned in a 3D simulated scene with controlled illumination and rendering parameters. Images captured in this scene are then blended with images from real UAV flights using an 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.

3.2.1. Methodology

The Synthetic Imagery Generating Software (SIGS) is a cross-platform graphical user interface software based in Qt and written in C++. This software takes a video with aerial images as an input, and for every frame automatically inserts a simulated aircraft in the image, according to the requirements provided by the user. This is done by blending the real-world images with synthetic images which contain the simulated aircraft, using an image processing algorithm. All the blended frames are then placed in a video sequence that realistically simulates an aircraft encounter situation.

SIGS relies on GLC (Lib, 2005), an OpenGL-based rendering library.

Chapter 3. Aircraft detection in the visible spectrum

(a) Cessna C172P

(b) Boeing B737-100

(c) EC130

(d) Paraglider

Figure 3.1: Example of 3D models in OBJ format that can be used to simulate flying obstacles.

This library allows to render 3D scenes using 3D models in OBJ format. FlightGear offers many 3D models for different types of aircraft in this format. Some examples are shown in Fig. 3.1. 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. Fig. 3.2 illustrates this.

Optimized rendering options are used for obtaining the best possible results when rendering the 3D model. For the images presented in this thesis, the software SIGS relied on a Nvidia GeForce GTX 670MX graphics processing unit (GPU), which allowed for the following rendering options:

- Averaged normals for a smooth appearance;
- Antialiasing to minimize the “ladder” effect;
- Anisotropic filtering to improve texture sharpness.

Four reference systems have been defined for controlling the objects and illumination of the scene, as shown in figure 3.3. 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 images, with its optical axis pointing in the X direction. The camera movement has been simplified so that the camera moves with its optical axis parallel to the ground plane at a constant

Figure 3.2: Synthetic 1280×960 pixel image generated with two rendered 3D aircraft models over a chroma key green backdrop. Since edges look pixelated, directly copying the rendered models into a frame does not provide realistic appearance and therefore blending operations are desirable to improve the quality of the result.

speed defined by the user. 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.

For illuminating the scene, an OpenGL point light source positioned at a 3D location has been used. This point light is omnidirectional and has white color as default option. The user should set the position of the illumination source according to the position of the sun in the input video to get realistic illumination results on the simulated aircraft.

Trajectory generation

In order to have smooth and realistic flight trajectories of the simulated intruder, these are generated using the basis spline model, a common method for aircraft trajectory modeling Delahaye et al., 2014. In particular, uniform splines of degree three have been implemented. For this, a node vector $\mathbf{T} = \{t_0, t_1, \dots, t_n\}$ is considered with $t_0 \leq t_1 \leq \dots \leq t_n$ and n points $\mathbf{P}_i$.

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

$$\mathbf{X}_3(t) = \sum_{i=0}^n B_{i,3}(t) \cdot \mathbf{P}_i, \quad 3 \leq t \leq n + 1 \quad (3.1)$$

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

Figure 3.3: Reference systems for synthetic image generation.

$$B_{i-2,3}(t) = \frac{1}{h} \begin{cases} (t - t_{i-2})^3 & \text{if } t \in [t_{i-2}, t_{i-1}] \\ h^3 + 3h^2(t - t_{i-1}) + 3h(t - t_{i-1})^2 - 3(t - t_{i-1})^3 & \text{if } t \in [t_{i-1}, t_i] \\ h^3 + 3h^2(t_{i+1} - t) + 3h(t_{i+1} - t)^2 - 3(t_{i+1} - t)^3 & \text{if } t \in [t_i, t_{i+1}] \\ (t_{i+2} - t)^3 & \text{if } t \in [t_{i+1}, t_{i+2}] \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

where h is the distance between two consecutive points.

The number of frames in the video determines the number of trajectory points sampled from the trajectory. This trajectory fits the waypoints provided by the user in the graphical user interface, computing the 3D position of the simulated aircraft for each frame of the input video. The aircraft orientation is also computed for each node in the computed trajectory so that the aircraft follows it in a realistic way in terms of pitch and yaw angles.

Image blending algorithm

The proposed image blending algorithm blends real images with synthetic ones, simulating the effects of lens radial and tangential distortions and out-of-focus blur in the simulated object and blending it with the real image while smoothly interpolating object edges. Brown's lens distortion model has been used (Brown, 1971, Fryer and Brown, 1986).

This image processing algorithm implemented has undergone several improvements with respect to the original algorithm published in Carrio et al., 2014 in order to give a more realistic feel to the generated pictures. This is achieved thanks to the incorporation of lens distortion effects and also blur to simulate the object being out of focus. A comparison of the results

Figure 3.4: Comparison of the former and the new image compositing algorithms. The incorporation of camera effects leads to more realistic results affecting the simulated object only and leaving the background unaltered.

using the original method and the current algorithm is shown in Fig. 3.4. The OpenCV library (Bradski, 2000) has been used for implementing the image blending algorithm.

The image compositing process is shown in Fig. 3.5. 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:

$$\begin{aligned} x' &= x \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2) \\ y' &= y \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1 (r^2 + 2y'^2) + 2p_2 x' y' \end{aligned} \quad (3.3)$$

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

$$\begin{aligned} u &= x' f_x + c_x \\ v &= y' f_y + c_y \end{aligned} \quad (3.4)$$

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. 3.5) are copied into image A to obtain image B. This image contains an undistorted background while keeping the simulated intruder unaltered. 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 then blurred by convolving with a kernel to produce image D. For the development of this software, the out-of-focus blur or bokeh effect has been simulated in a simplified way by convolving the image with a gaussian kernel trying to obtain the most realistic results possible. The gaussian kernel is built setting the

Figure 3.5: Image blending process with sample images.

standard deviation proportionally to the square root of the distance between the simulated intruder and the camera, to increment the out-of-focus effect on the simulated objects as they get farther away from the camera. An exhaustive implementation of the bokeh effect exceeds the object of this simulation, as it would require a much more complete knowledge about the nature of the lenses used to capture the real images and the geometry of the scene (Conrad, 2009).

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 distortion alters background pixels (information is lost 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 no longer have the color of the backdrop so this image can be used as a mask to obtain the final output image. Backdrop (green) pixels in image F have not been modified and therefore are copied from input 1 into the final image. Modified pixels, which include the simulated aircraft and the pixels interpolated with the background around the edges are copied from image D into the output to obtain the final blended image.

Automatic detection of occluded areas

The UAV frame may occlude regions within the field of view of the camera. The detected areas of occlusion can be used to obtain a mask and be later exploited by the software to correctly simulate the occlusions caused by the UAV frame that would appear in a real flight with the same camera setup. Additionally, the obtained mask can be used in vision-based detection algorithms to mask out pixels which do not contain relevant information for the visual detection task, discarding false positives or making the image search process more efficient by removing those blind areas. Based on the vibration of the UAV with respect to a stabilized camera, an algorithm has been implemented in order to automatically detect those occluded areas. Image samples, recorded on board a fixed-wing UAV, are shown in Figs. 3.6a, 3.6c, 3.6e and 3.6g.

The detection of these occluded areas happens in two stages. In the first one, inter-frame differences are computed and accumulated over several frames in a video sequence in order to obtain the edges of the vibrating elements which cause the occlusion. Afterwards, an attempt to close the

Chapter 3. Aircraft detection in the visible spectrum

found contours is performed. Normally the segmentation of the occluding objects causing the occlusion is not complete for different reasons. For example, there might be holes or areas next to the image borders which cannot be completely segmented. A post-processing stage completes the process by correctly closing and filling all the contours of the areas causing occlusion.

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

During the first stage, the difference image between successive frames is computed and normalized. Then, the sum of the normalized difference images is computed for 1500 frames (1 minute of video at 25 fps). The obtained image, shown in Fig. 3.7 basically enhances the edges of the UAV. This is due to the fact that, since the UAV vibrates while the camera does not, the areas corresponding to pixels with the maximum accumulated change across consecutive frames become enhanced. In the background, differences between frames are also observed, but these variations are not systematic, so their sum has a relatively lower intensity when measured over a large number of frames. 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 pre-processed images.

As previously mentioned, the pre-processed images are incomplete masks of the occluded areas, in this case the UAV frame. These images need to be further 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 actual occluded areas. 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 figures 3.8c and 3.8d respectively.

The images obtained through this first step show closed contours suitable for being filled, with the exception of the tail 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 figure 3.8e.

The next step consists in filling the obtained closed contours. Contours

(a) Original image.

(b) Processed image.

(c) Original image.

(d) Processed image.

(e) Original image.

(f) Processed image.

(g) Original image.

(h) Processed image.

Figure 3.6: Results of the automatic UAV occlusion detection algorithm.

Figure 3.7: Sum of normalized differences between successive frames. The edges of the objects vibrating, in this case the contours of the UAV carrying the camera, are enhanced.

(a) Original image.

(b) Pre-processed image.

(c) After bi-directional Sobel.

(d) After morphological closing.

(e) Closing tail contours.

(f) Final segmentation.

Figure 3.8: UAV mask refinement process.

Figure 3.9: Examples of automatically produced UAV masks.

which are smaller than an experimentally fixed threshold (0.5 % of the area of the image) 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 figure 3.8f. Fig. 3.9 shows additional examples of the masks obtained through this process.

3.2.2. Experiments and results

Graphical User Interface (GUI)

A user-friendly GUI has been designed in order to facilitate the generation of labeled videos. The GUI includes five configuration blocks: input files, illumination, camera speed, intruder trajectory and output video.

In the input files block the user can select his/her own camera calibration file, the input video, a mask file for occluding the intruder and the 3D model

Figure 3.10: Main window of the implemented graphical user interface. Block 1 is used to select 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.

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. The input video file should be in AVI format. The mask can be in any of the image formats supported by OpenCV. Regarding the 3D model, only OBJ files, one of the most standard formats, is supported. These formats have been chosen for their cross-platform support. The software generates complete video annotations 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. 3.3.

Sample application for aircraft tracking

The SIGS software has already contributed to the development of the AM³, a learning-based visual tracker that has achieved some popularity (Fu et al., 2014a, Fu et al., 2014b). For its development and evaluation, a video dataset consisting of more than 14k frames of flight imagery with simulated intruders was generated using the SIGS software.

The video dataset consisted of a set of 24 videos whose details are summarized in table 3.1. These videos offered a variety of trajectories, illumination and cloud conditions which were very challenging for detecting or tracking aircraft, as shown in Fig. 3.11. This dataset has also been used for the evaluation of the aircraft detection algorithm presented at the end of this chapter. The images from real flights used as backgrounds were captured from USol K50 UAVs. These images were kindly provided by the Spanish UAV manufacturer Unmanned Solutions (USol). 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.

Some evaluation results of the AM³ visual tracker with two different videos are shown next. These videos include challenging situations, e.g. rapid pose variation, background clutters, illumination changes, onboard mechanical vibration and wind influence. The performance of this tracker was evaluated with the ground truth data generated by the SIGS software, and

Chapter 3. Aircraft detection in the visible spectrum

Figure 3.11: Captures from real flight videos used as backgrounds. The videos were recorded in 2011 in Marugán (Spain) from an USol K50. The images show different cloud conditions.

Filename	Intruder aircraft	No. frames	Trajectory	Comments
1LRB737.avi	Boeing 737	322	Left to right	Heavy clouds
1FNB737.avi	Boeing 737	389	Head on	Heavy clouds
1NFB737.avi	Boeing 737	636	Tail chase	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	Head on	Clear sky
2NFB737.avi	Boeing 737	1143	Tail chase	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	Head on	Vibrations
3NFB737.avi	Boeing 737	1033	Tail chase	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	Head on	Heavy clouds
1NFC172.avi	Cessna C172	636	Tail chase	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	Head on	Clear sky
2NFC172.avi	Cessna C172	1143	Tail chase	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	Head on	Vibrations
3NFC172.avi	Cessna C172	1033	Tail chase	Vibrations
3RLC172.avi	Cessna C172	1033	Right to left	Vibrations

Table 3.1: Video dataset used in the development of the AM³ tracker.

Figure 3.12: Center Location Error over 1LRB737 video. The results obtained on five different frames extracted from this video are shown. Mean CLE is 3 pixels in this scenario.

Figure 3.13: Center Location Error over 3LRB737 video. Here the mean CLE is 3 pixels until frame 400 and then rises to 5 pixels due to stronger vibrations of the UAV.

Chapter 3. Aircraft detection in the visible spectrum

the Center Location Error (CLE) was applied as evaluation metric. CLE is defined as the Euclidean distance from the detected intruder aircraft center to its ground truth for each frame.

Taking 1LRB737 and 3LRB737 videos as examples, Fig. 3.12 and Fig. 3.13 show the center location error results of AM³ tracker with these videos. Although both videos show similar trajectories, the performance obtained is different because of the different level of turbulence experienced by the UAV in these videos. These videos have allowed to tune the parameters and optimize the AM³ tracker for a significant variety of challenging conditions.

Evaluation by human pilots

In order to evaluate the realism of the generated images, a test was made to seven professional pilots using images containing real aircraft and aircraft simulated using the SIGS software. 24 images showing a total of 33 aircraft were used in total: 9 showing a real aircraft, 6 showing a simulated aircraft and 9 showing a simulated aircraft next to a real aircraft within the same image. Some examples of the images showing two aircraft are shown in Fig. 3.14. Both the real and the synthetic aircraft had similar sizes, between 10 and 65 pixels in their maximum dimension. There were no previous indications on the location or number of aircraft visible in the image and the same order was randomly chosen to show the images to the pilots. Images had a variety of backgrounds and illumination conditions, always with daylight, and were shown to the pilots allowing them to zoom in as much as desired into the scene.

The pilots taking part in the test were relatively young, between 22 and 46 years old. Between them there were both civil and half military pilots. All military pilots except for one were helicopter pilots, while all the rest were trained to fly airplanes. The details about each of the pilots can be found in Table 3.2.

The conclusions that can be extracted out of the answers of the test are quite interesting. To the question of how many aircraft the pilots could see in each of the pictures, zero, one or two, a majority always agreed on the correct answer. The results are summarized in Fig. 3.15. This indicates that the aircraft had enough size and had an appearance that allowed them to be detected, regardless of the aircraft being real or simulated.

When looking at images with a single aircraft, the results, shown in Fig. 3.16, are also very interesting. For the images showing real aircraft, in all nine pictures except for one, a majority of pilots again gave the right answer, correctly identifying the real aircraft as real. However, consensus drops and in half of the images, the simulated images appear to pilots are

Age	Type of pilot	Type of aircraft	Flight hours
34	Civil	Airplane	600
46	Civil	Airplane	12000
43	Military	Helicopter	2600
36	Military	Helicopter	1800
44	Military	Airplane	5000
41	Military	Helicopter	2000
22	Civil	Airplane	210

Table 3.2: Details about the pilots participating in the test.

real ones. Actually, all pilots surprisingly agreed that the simulated aircraft in image no. 1 looked real to them. This indicates that simulated images look very realistic as long as they are observed individually.

When looking at images containing two aircraft, a real and a simulated one, again there was not much agreement between pilots and the observations of the majority were correct only in two out of nine images (22.2 %). Only in three out of nine images (33.3 %) a majority of pilots correctly stated that one of the images was real and the other simulated. On most images (55.5 %) a majority of pilots considered both images real. Those pilots who correctly identified that one of the aircraft was real while the other was not -something that happened with seven of the images only- were asked to identify which of the two was the real aircraft. In this case, in five out of seven pictures (71.4 %) the majority could tell which of them was real one. A summary of the answers are plotted in Fig. 3.17. From these questions, we can infer that, while it is still not an easy task (a majority of pilots did right in only 33 % of the images), it is easier to pick out a simulated aircraft when it is next to a simulated one.

In conclusion, the generated images unquestionably have a realistic appearance. This guarantees that the images do not have any visual features or artifacts perceptible to an expert human eye.

3.2.3. Conclusions and future works

A methodology has been developed for obtaining realistic images of synthetic aircraft blended with real aerial images. These images have been created by blending a 3D-rendered aircraft image with real flight images recorded from a UAV, taking into account illumination, camera lens distortions and the out-of-focus effect achieving realistic results, which have been validated by professional pilots.

(a) Real (above) and simulated aircraft (below).

(b) Real (above) and simulated aircraft (below).

(c) Real (right) and simulated aircraft (left).

(d) Real (below) and simulated aircraft (above).

Figure 3.14: Sample images shown to professional pilots to evaluate the realism of the aircraft blending.

Figure 3.15: Pilots guesses about the number of aircraft present in the image. Images contained a maximum of two aircraft, either real or synthetic. Results indicate the majority of pilots always agreed on the right answer for each image.

Figure 3.16: Pilots guesses on the type of aircraft seen (real or synthetic) when seeing a single aircraft in the image. When the image is real, most pilots agree in most pictures. However, if the aircraft is synthetic, they do not generally agree.

Figure 3.17: Pilots guesses on the type of aircraft seen (real or synthetic) when seeing two aircraft in the image. On most of the images a majority of the pilots believe both images appear to be real. The majority realizes that one is real and the other synthetic only in 3 out of 9 pictures. If they guess right that one is real and the other synthetic, they generally can tell which of the two is the real one.

Additionally, an algorithm for the automatic detection of the objects vibrating in the scene has also been implemented. Since these objects create occlusions in the images used in this research, the segmentation of the areas where these occlusions happen can be used for simulating the occlusion effect in the generated imagery and could also be potentially used in detection algorithms for masking out pixels which do not contain relevant information for the visual detection, making the image search process more efficient.

The images obtained with this software can be used for the development and evaluation of vision-based DAA systems and have already proved to be a powerful benchmark, therefore representing a significant improvement in the state of the art for vision-based DAA systems.

Regarding future works, new features will be added such as multi-object simulation and support for geo-referenced images. The work presented in this section has been originally published in Carrio et al., 2014 and Carrio et al., 2016a.

3.3. Sky segmentation and horizon detection

Vision-based sky segmentation and horizon line detection are two intimately related problems whose solutions can contribute to some fundamental problems of UAV systems. Such is the case of attitude estimation, in which an accurate estimation of the horizon line by means of Computer Vision can

be used to accurately determine UAV orientation for autonomous navigation. Nowadays, vision-based approaches are being used to improve attitude estimations on board UAVs based on horizon line detection in the image plane, which can be used to estimate the orientation of the UAV. These approaches are based on the extraction of image features with the purpose of detecting the horizon line. The biggest constraint with these approaches is the high computational cost of the Computer Vision and/or machine learning algorithms involved. The main challenge in real-world flights is to automatically interpret images robustly enough to be used in various environments and fast enough to achieve real-time performance.

Sky segmentation also has applications in collision avoidance systems for UAVs. Vision-based aircraft detectors for UAVs may improve their performance using different techniques or parameter sets for the detection of aircraft in ground and sky backgrounds.

Taking into account the need for real-time performance in the aforementioned horizon detection applications, a novel strategy with low computational cost for sky segmentation is proposed and used as a preliminary step for horizon line detection. In particular, an approach for semantic segmentation (pixelwise classification) of the sky has been designed to minimize the computational cost while processing at real-time frame rates images captured on board UAVs, obtaining processing times below 100 ms per frame in 1280×960 resolution images, outperforming other existing approaches in terms of computational cost while providing accurate segmentation results.

3.3.1. Proposed approach

The general strategy for the proposed sky segmentation method is as follows: first an input picture is divided into equally sized subregions or patches. The UAV carrying the camera is visible in the images used for the development of this sky segmentation algorithm. A mask of the pixels corresponding to the UAV frame has been obtained in advance and patches which contain parts of the UAV frame are not considered in the approach. A logistic regression classifier is then used on each of the patches (which might contain sky pixels only, ground pixels only or a mixture of both), outputting probabilities of belonging to two different classes, namely *sky* and *ground*. Finally, a post-processing stage is applied in order to correct the eventual misclassification of sky and ground patches from the segmented images and to obtain the final result. Additionally, a method for automatic horizon line detection based on the segmented images is proposed. In the next paragraphs, the construction of the descriptors used to train the classifiers is discussed. The classification strategy as well as the post-processing stage

Chapter 3. Aircraft detection in the visible spectrum

for removing potential misclassified sky/ground patches is also presented. Finally, the proposed method for horizon line detection is presented.

Descriptor design and classification

A descriptor containing local color information for every patch has been built combining 256-bin normalized histograms for each of the three color channels in the patch. The descriptor for a given patch is built in the following way:

$d = \{H_1^0, H_2^0, H_3^0, H_1^1, H_2^1, H_3^1, \dots, H_1^{255}, H_2^{255}, H_3^{255}\}$, where H_j^i is the i -th bin of the j -th channel normalized histogram. The total length of the resulting descriptors is 768 bins.

A logistic regression classifier trained with the aforementioned descriptor was used to estimate the probability of a given image patch of belonging to the class *sky* or *ground*. This is done using a logistic regression classifier with weights W and biases b . Given j classes or possible values of a stochastic variable Y , the probability of an input vector x belonging to class i is given by:

$$P(Y = i|x, W, b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}} \quad (3.5)$$

The classification of the patch is performed by choosing the class with the highest *a posteriori* probability:

$$y_{\text{pred}} = \underset{i}{\operatorname{argmax}} P(Y = i|x, W, b) \quad (3.6)$$

In order to correct potentially misclassified patches, a simple strategy based on checking the 4-connected connectivity between patches for discordances is used. If the patches immediately above, below, to the right and to the left of a given patch P share the same predicted class and this class is opposite to that of P , then the predicted class of P is changed to that of its neighbors. In the case of patches located on the image corners and borders, an analogue strategy is used, using 2-connected and 3-connected pixels, respectively.

Horizon line detection

Our method for automatically detecting the horizon line starts by performing a Canny edge detection over the segmented image. Those extracted edges falling within the UAV frame mask and therefore corresponding to the UAV frame's edges are removed. Finally, a line is fitted over the remaining

detected edges. The line is fitted by minimizing $\sum_i \rho(r_i)$, where r_i is the distance between the i -th point and the line, and ρ_i is the Welsch distance function Dennis Jr and Welsch, 1978, given by:

$$\rho(r) = \frac{C^2}{2} \left[1 - \exp \left(- \left(\frac{r}{C} \right)^2 \right) \right] \quad (3.7)$$

where $C = 2,9846$. This fitting method was chosen empirically based on the observed performance.

3.3.2. Evaluation methodology

Our approach has been evaluated in a dataset with real-world images captured on board UAVs. A statistical analysis was performed in order to select a color space that best enhanced the differences between the histograms of ground and sky patches. A k-fold cross-validation strategy was used for validation of a logistic regression classifier and for optimization of a regularization parameter. The performance of the classifier was measured using the mean zero-one error and with the help of confusion matrices. Since this work focuses on the sky segmentation, the horizon line is simply extracted and evaluated in terms of its computational cost while its accuracy has only been evaluated in a qualitative way. The residual from the line fitting algorithm could be used for further improvements or quantitative comparisons with other methods and is left for future work.

The dataset, whose details are summarized in Table 3.3, was created with the purpose of training the proposed system and evaluating its performance. This dataset is composed of approximately 3000 images extracted from videos taken from 6 different flight scenarios. The videos were acquired with a GoPro Hero 2 equipped with 170 ° fisheye lenses and fixed to the tail of an USol K50 UAV. Camera defaults for exposure, gain and white balance were used (auto mode). The weather conditions in the images vary from a clear to an overcast sky with various changes in illumination. The ground appearance has great variety, ranging from clear tones (e.g. wheat crops) to darker tones (e.g. forests).

The labeling process consisted of a human expert drawing the horizon profile for each of the pictures. The picture is hence separated into two regions, namely *sky* and *ground with aircraft*. The pixels corresponding to the UAV frame are labeled using a separate mask generated automatically using the method explained in Sec. 3.2.1. Pictures from the dataset as well as UAV frame masks and ground truth data are shown in Fig. 3.18. All the videos, masks and ground truth data are publicly available for download at <https://www.kaggle.com/adriancarrio/aircraft-detection-and-sky-segmentation-dataset>.

Figure 3.18: Pictures extracted from the dataset with their corresponding UAV frame masks and ground truth sky segmentations.

The images in the dataset were split into 1024 patches of 40×30 pixels. Each patch may fall into one of the following categories:

- Sky patches. Any patch containing only pixels of sky areas.
- Ground patches. Any patch containing only pixels of ground areas.
- Aircraft patches. Any patch containing pixels belonging to the aircraft.
- Mixture patches. The remaining patches.

Aircraft patches were not considered throughout the sky segmentation approach since their position in the image is fixed and known in advance

3.3. Sky segmentation and horizon detection

Video sequence	Image resolution	Number of frames	Sky patches	Ground patches	Aircraft patches	Mixture patches	Sky condition
1	1280×960	282	134070	69702	76140	8856	Cloudy
2	1280×960	940	484563	222885	226540	28572	Scattered
3	1280×960	825	366828	216730	237600	23642	Clear
4	1280×960	193	91612	49447	49987	6586	Scattered
5	1280×960	167	75232	45619	45090	5067	Stormy
Evaluation	1280 × 720	542	149336	149168	101896	15856	Clear

Table 3.3: Details of the image dataset used. Video sequences #1 to #5 were used for color space analysis and for training the classifier. The evaluation video sequence was used for testing purposes only. The number of patches is given for patches of size 40×30 pixels.

(given by the pre-generated masks). The number of 40×30 patches belonging to each category available in the dataset are shown in Table 3.3.

One of the videos was arbitrarily selected for evaluation purposes only and its contents were excluded from the classifier training and optimal color space selection processes. This specific video sequence shows different illumination conditions, with partially cloudy skies and direct sun exposure in certain frames. Its resolution, 1280×720, is also different from that of the rest of images (1280×960). This was done on purpose to evaluate the impact of image resolution variations. In this case, the resulting number of 40×30 patches per frame is 768.

Color space selection

A statistical analysis has been proposed in order to select a suitable color space that better enhances the differences between sky and ground patches. For this purpose, the average ground histogram and an average sky histogram were computed for sky and ground patches from all the images available in the dataset using various color spaces and excluding the video sequence considered only for the final evaluation of the proposed approach.

A normalized correlation coefficient (Eqs. 3.8 and 3.9) was used in order to compare the average sky and ground histograms for each of the color channels. The influence of histogram equalization in the value channel for illumination normalization was considered in all cases and the histogram equalization was performed before converting to the desired color space and computing the average histograms.

$$\rho(H_1, H_2) = \frac{\sum_i (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_i (H_1(I) - \bar{H}_1)^2 \sum_i (H_2(I) - \bar{H}_2)^2}} \quad (3.8)$$

$$\bar{H}_k = \frac{1}{N} \sum_j H_k(j) \quad (3.9)$$

Classifier training and validation procedure

Our training set contained 40k randomly chosen descriptors, half of them corresponding to sky patches and the other half corresponding to ground patches. Additionally, we trained a logistic regression-based classifier with three output classes: *ground*, *sky* and *mixture* by adding 20k descriptor samples of mixture patches to the training dataset. Training and validation of the classifiers was implemented using the Theano library (Theano Development Team, 2016).

The negative log-likelihood function was used as loss function. Even though the loss is formally defined as the sum over the individual error terms for all the samples in the training set, we have instead used the mean, allowing for the learning rate choice to be less dependent on the mini-batch size. For the i -th sample in a training set $\mathcal{D}$, the loss function ℓ , under the model parametrized by θ , can be written as:

$$\ell(\theta = \{W, b\}, \mathcal{D}) = - \frac{\sum_{i=0}^{|\mathcal{D}|} \log(P(Y = y^{(i)} | x^{(i)}, W, b))}{|\mathcal{D}|} \quad (3.10)$$

All available samples in the training set were used for a 5-fold cross-validation strategy. Early-stopping was implemented for optimal selection of the L2 regularized logistic regression term and for validation. Our early stopping implementation limited the number of optimization iterations to 100, doubling the number of iterations available after each iteration (up to a maximum of 100) if the training error dropped by more than 0.5%. Stochastic gradient descent with momentum was used for training, and a mini-batch size of 100 was used. The learning rate was set to 1.0 and the momentum parameter to 0.9.

3.3.3. Results and discussion

Color space selection

The results of the color space analysis performed, shown in Fig. 3.19, indicated that the blue channel has the minimum correlation between histograms of opposite classes. Based on this fact, BGR was chosen as the color space to compute the color descriptors.

Figure 3.19: Normalized correlation coefficient between the average sky and ground histograms in each color space with and without prior value channel equalization of all averaged histograms.

Performance of the classifier

The suitability of value channel histogram equalization for enhancing sky/ground patch differences could not be clearly concluded by simply looking at the results of the color space analysis. Therefore, we separately trained our classifier with 40k descriptors of sky/ground patches of images with and without prior value channel histogram equalization. A mean zero-one validation error of 1.030 % was obtained when training with histograms with prior value channel equalization. For the descriptors extracted from images without value histogram equalization, a mean zero-one validation error of 0.775 % was obtained.

20K additional samples of descriptors extracted from *mixture* patches were added to the training set and the classifier was trained to now output the probabilities of belonging to each of the 3 output classes: *ground*, *sky* and *mixture*. Validation errors increased substantially (26.863 % and 23.545 % mean zero-one validation errors, with and without prior value channel histogram equalization, respectively). This was probably due to the closer similarity between classes in this case. Mixture patches were therefore not used for training the final classifier. As for the two-class classifier approach, it was concluded that histogram equalization of the value channel did not improve the classification results and was dismissed from the proposed approach.

A classifier predicting the two output classes *ground* and *sky* was used in the final implementation. The trained classifier was tested on all the available images in the dataset, which contain ca. 2M patches. The confusion

Chapter 3. Aircraft detection in the visible spectrum

matrices and F1 scores shown in Table 3.4 indicate good classification results for the trained classes, with no remarkable performance differences in the video sequence exclusively designated for evaluation.

Video sequence	Confusion matrices	F1 score
#1	$\begin{bmatrix} 69442 & 260 \\ 8 & 134062 \end{bmatrix}$	0.998
#2	$\begin{bmatrix} 222043 & 842 \\ 1208 & 483355 \end{bmatrix}$	0.995
#3	$\begin{bmatrix} 212232 & 4498 \\ 50 & 366778 \end{bmatrix}$	0.990
#4	$\begin{bmatrix} 49365 & 82 \\ 2935 & 88677 \end{bmatrix}$	0.970
#5	$\begin{bmatrix} 44953 & 666 \\ 2977 & 72255 \end{bmatrix}$	0.961
Evaluation	$\begin{bmatrix} 144672 & 4496 \\ 62 & 149274 \end{bmatrix}$	0.984

Table 3.4: Confusion matrices and F1 score indicating the performance of the trained classifier for all the ground patches and the sky patches in the dataset. In the confusion matrices, columns represent the predicted classes while rows represent the actual classes. First row and column correspond to the category *ground* and second row and column correspond to *sky*. Mixture patches are not included here as they were not considered as output classes of the classifier.

Sky segmentation performance

Qualitative results of the post-processing stage and final sky segmentation results are shown in Figs. 3.20 and 3.21, respectively. Since the trained classifier does not predict probabilities of belonging to class *mixture*, obtaining a quantitative performance analysis that included these patches using our classifier was not possible. However, the performance of the full sky segmentation approach was estimated by assuming a pessimistic scenario,

Figure 3.20: Examples of the proposed post-processing strategy for removing misclassified patches based on patch connectivity.

in which all mixture patches would be misclassified. Results are shown in Table 3.5.

Video sequence	Mean zero-one error before post-processing (%)	Mean zero-one error after post-processing (%)
#1	4.29	4.28
#2	4.16	4.11
#3	4.64	4.61
#4	6.50	6.26
#5	6.92	6.74
Evaluation	6.49	6.39

Table 3.5: Sky segmentation performance estimation, assuming a pessimistic scenario, in which all mixture patches would be misclassified.

Horizon line detection

Although it assumes a linear horizon, the proposed method provides qualitatively good detections of the horizon line in the evaluated videos, showing robustness to partial occlusions. A coarse segmentation using 40×30 pixel patches is sufficient to provide good results, even in the cases when the

Figure 3.21: Final sky segmentation results.

horizon line is partially occluded by the aircraft carrying the camera. Two horizon line detection examples are shown in Fig. 3.22.

Processing time

Our C++ implementation, with no code optimizations, performs sky segmentation at real-time frame rates on an quad-core Intel i7 CPU running at 3.4 GHz. Tests were run off-line and the processing time was averaged for all frames available in each resolution. The results are shown in Table 3.6. Our approach allows for image segmentation at 34 Hz for 960p images using 40×30 patches and at 10 Hz using 20×15 patches. If required, horizon line detection adds an average of 6.4 ms of computation time per frame and still allows for 10 Hz frame rates with 20×15 patches.

3.3.4. Conclusion and future works

A real-time supervised learning approach for sky segmentation has been implemented and tested in real UAV flight footage. A search for a color space that is suitable to differentiate sky and ground regions was conducted. A novel strategy was implemented, consisting of the segmentation of the image into equally sized patches, extracting color descriptors for each patch,

Figure 3.22: Horizon line can be automatically detected using the images segmented with the proposed method. The left image shows a horizon line (red) detection example under challenging illumination and cluttered sky conditions. The right image shows the horizon line detection with a partial occlusion. A coarse segmentation using 40×30 pixel patches is enough to provide visually accurate results. The method is robust to partial occlusions.

Resolution & patch size	Classification time [ms]	Post-processing time [ms]
1280 × 960 40 × 30 patches	26.5	2.8
1280 × 720 40 × 30 patches	19.7	2.1
1280 × 960 20 × 15 patches	82.9	10.7
1280 × 720 20 × 15 patches	62.4	7.8

Table 3.6: Average processing time per frame for sky segmentation. Times were measured for different image resolutions and patch sizes. Detection of the horizon line required additional 6.4ms.

classifying color descriptors using a logistic regression classifier and post-processing the segmented image to improve the quality of the results. The proposed approach allows for real-time segmentation at frame rates varying between 10 and 34 Hz depending on the image resolution and the desired segmentation fineness, achieving great accuracy.

In future works, this dataset will be extended to cover an even more diverse set of outdoor scenarios, the performance of different supervised learning classifiers will be compared and superpixels will be used for prior segmentation instead of fixed-size patches. As already mentioned, a quantitative evaluation of the goodness of the detected horizon line is also left for future work. The work presented in this section has been originally published in Carrio et al., 2016b.

3.4. Aircraft detection algorithm

In this section, a novel efficient algorithm for aircraft detection based on contrast features extracted from visible spectrum images has been proposed and evaluated. The algorithm is based on a track-before-detect strategy and aims at providing a highly efficient approach for aircraft detection when they have relatively small sizes in the image while being able to deal with changes in scale. The algorithm is specifically designed for detecting aircraft in collision course or close to it, based on the fact that these aircraft should appear as stationary features in the image plane (Angelov, 2012).

3.4.1. Proposed approach

The proposed algorithm (Alg. 1) operates by extracting high-contrast points using morphological operations. The sky region is segmented (lines 9-22 in the algorithm), following the method described in Section 3.3. Then, the positions of the high-contrast points found in the sky region are tracked, so that points that are systematically detected across multiple frames and whose speed -measured as displacement in the image plane- is below a threshold are declared targets.

In order to detect potential aircraft in collision course, the proposed algorithm first performs a top-hat morphological operation with a fixed-size kernel in order to enhance small high-contrast areas in the image. The resulting image is thresholded using a fixed threshold. In order to allow aircraft with large image sizes to be detected, whenever the width or height of the last declared threat is larger than a fixed threshold, the resulting image is smoothed using a normalized box filter so that high contrast points

in aircraft with large sizes in the image are blended together as much as possible. The size of the normalized box filter is chosen proportionally to the width of the last declared threat.

Then, sky is segmented and all high-contrast areas which fall outside the sky region are discarded as potential threats. A contour search is performed to localize each of the remaining high-contrast areas or blobs and each of them is compared against a list of potential targets. If a blob is matched to a potential target because of its location (being within a given radius of the potential target), the target is updated with the position and bounding rectangle of the blob. If not, the blob is added as a new potential target. The radius allowed for a match to be considered is initialized with a fixed value but may be increased if the size of the declared threats exceeds a given threshold.

A variable named *patience* has been defined, which stores for how many consecutive frames can a potential target be maintained in the list of potential targets without a blob match being found for it. It is initialized with a fixed value that decreases with each new frame if the blob is not matched to a potential target, or stays the same if a match occurs. Additionally, the variable *total_detections* stores the number of times a certain potential target has been detected (matched to a blob) and is used as signal to declare threats. If a certain potential target has been detected (matched to a blob) for a certain number of times without it being discarded because of patience being over, then it is declared as a genuine threat and will continue to be declared for every new frame until patience is over.

3.4.2. Evaluation methodology

For its evaluation, the proposed method will be compared with some of the most recent evolutions of the YOLO algorithm (Redmon and Farhadi, 2018, Bochkovskiy et al., 2020). YOLO is a state-of-the-art real-time object detection algorithm, currently considered one of the fastest detectors in the literature, and it has already been considered for aircraft detection in the visible spectrum (Opromolla et al., 2019). In particular the standard YOLOv3 and YOLOv4 and the theoretically faster but less accurate versions YOLOv3-tiny and YOLOv4-tiny have been evaluated together with the proposed approach. These faster versions achieve more speed by reducing the number of layers of the model and sometimes also the input image size. The standardized mean average precision (mAP) metric based on the 11-point interpolation method (Zhang and Zhang, 2009) has been used for this purpose.

In order to understand the aforementioned metrics, the concepts of IoU,