



Autonomous Decision-making for UAVs Operating under Environmental and Object Detection Uncertainty

Juan David Sandino Mora
B.Eng. (Mechatronics)

Submitted in fulfilment of the requirement for the degree of
Doctor of Philosophy

School of Electrical Engineering and Robotics
Faculty of Engineering
Queensland University of Technology

2022

Keywords: drones, unmanned aerial vehicle (UAV), unmanned aerial system (UAS), aerial robotics, computer vision, embedded systems, motion planning, partially observable Markov decision process (POMDP), search and rescue (SAR), planetary exploration, object detection uncertainty, vision processing unit (VPU).

*To my parents, Dora and Rafael, my family, and my friends; who always supported and trusted in me,
on what I can achieve and how far I can go.*

*A mis padres, Dora y Rafael, mi familia, y a mis amigos; los cuales siempre me apoyaron y confiaron
en mí, en lo que puedo conseguir y que tan lejos puedo ir.*

Abstract

THE increasing number of automated capabilities now available to better operate unmanned aerial vehicles (UAVs)—also known as drones—has resulted in their exponential adoption in civilian applications such as biosecurity, precision agriculture, mapping and surveying. Examples of these automated capabilities include automated ‘home returning’ in compromised mid-flight events (*e.g.*, low battery power, strong wind gusts, or loss of communication), reactive collision avoidance, autonomous take-off and landing, active object tracking and waypoint-based navigation. However, operational hardware and software limitations have restricted a broader use of *small UAVs*, those with a maximum take-off weight (MTOW) less than 13.5 kg, for applications that require onboard data processing and rapid decision-making. These time-critical applications are necessary in real-world emergency environments such as search and rescue (SAR) and disaster monitoring, where the identification, localisation, and quantification of objects (*i.e.*, victims and structures) is vital for first responders to coordinate and prioritise response efforts in affected, or concerned zones.

In real-world scenarios, collected data, or observations, by small UAVs about the environment they are surveying, are limited by imperfections from their sensors. Partial observability is caused by physical limitations on their sensor systems, the complex nature of the surveyed environment, object detection uncertainty from predictions of object detector models, and the intractability of predicting unlimited sequences of future action commands to interact in the world. Although existing research in autonomous navigation, utilising small UAVs already proposes approaches to survey partially observable environments, little analytic attention has addressed how to model and reduce object detection uncertainty from vision-based models during mid-flight events. Limitations of previous UAV framework implementations infer the need to integrate computationally expensive algorithms for autonomous decision-making and object detection onboard resource-constrained hardware systems contained in small UAVs.

This research therefore presents a methodology and framework to increase the cognitive levels of small UAVs navigating in challenging environments, and under object detection uncertainty using a sequential decision process (SDP), vision-based sensors and onboard inference of convolutional neural network (CNN) models. This research investigated how to formulate the navigation problem using partially observable Markov decision processes (POMDPs), and design a modular and scalable framework to compute a motion policy and integrate CNN models for object detection in resource-constrained hardware onboard small UAVs. The complete UAV system was tested using several SAR case studies to locate victims last seen inside surveyed indoor and outdoor environments. A scalability evaluation of the complete system in the field of planetary exploration was examined by autonomously segmenting and mapping desiccation cracks of dry lake beds, which may contain records of past or present life forms shaped as fossils. Experimental results from simulation and real flight tests indicate improved cognitive levels of small UAVs when operating in complex environments, as the aircraft computed and recorded traversed paths in real-time that allowed a better visualisation of partially occluded targets, and increased the confidence level of detected victims and desiccation

cracks from CNN models. Framework implementations of the UAV system for SAR and planetary exploration are validated with sub-2 kg quadrotor UAVs by using three key components: 1) a model-based POMDP solver to compute the motion policy; 2) a vision processing unit for onboard inference of CNN models; and 3) a modular design for payload flexibility in diverse autonomous navigation and object finding applications.

This research is significant for six key reasons: 1) it presents new mathematical methods to model and attenuate object detection uncertainty from modern computer vision detectors; 2) the research is validated by implementing a POMDP-based framework for autonomous decision-making in partially observable environments from imperfect sensor data; 3) such decision-making means UAVs are enabled to navigate autonomously in time-critical applications such as SAR, disaster management, surveillance, and planetary exploration; 4) enabling this higher cognitive power in small UAVs also improves the UAV's decision-making capabilities to collect more accurate statistics of detected objects, and increases robustness against noisy detection of potential objects from computer vision algorithms; 5) the robust validation process achieved with the designed UAV system to confirm detected objects in real-time when data is complex to interpret for UAV pilots, which reduces human bias on scouting strategies; and 6) this research and its outcomes can be extrapolated to any other fields and problems that demand real-time decision-making under uncertainty and partial observability of studied state variables.

Table of Contents

Keywords	ii
Abstract	iv
List of Figures	x
List of Tables	xiii
List of Publications	xiv
List of Acronyms	xv
Statement of Original Authorship	xvii
Acknowledgements	xviii
1 Introduction	1
1.1 Background	1
1.1.1 Time-critical Applications	3
1.1.2 Types of Object Detection Uncertainty	4
1.2 Research Problem	6
1.2.1 Research Questions	6
1.2.2 Aims of the Study	7
1.2.3 Scope	8
1.3 Research Significance	8
1.4 Thesis Outline	9
2 Literature Review	11
2.1 Autonomous Navigation Methods in UAVs	13
2.2 Autonomous Navigation for Small UAVs in Uncertain Environments	15
2.3 Sequential Decision Processes (SDPs)	16
2.3.1 Markov Decision Processes (MDPs)	16
2.4 Decision-making under Uncertainty and Partial Observability	18
2.4.1 Partially Observable Markov Decision Processes (POMDPs)	18
2.4.2 Autonomous Decision-making for Small UAVs using POMDPs	20
2.4.3 Adaptive Belief Tree (ABT)	21
2.5 Small UAV Frameworks for Decision-making in Time-critical Applications	23
2.6 Summary	24

3 Autonomous UAV Navigation for Active Perception of Targets in Uncertain and Cluttered Environments	26
3.1 Introduction	28
3.2 Related Work	29
3.3 Background	30
3.4 Problem Formulation	30
3.4.1 Partially Observable Markov Decision Process	30
3.4.2 Adaptive Belief Tree	31
3.4.3 Small UAV Navigation Task	31
3.4.4 State Space	31
3.4.5 Actions	32
3.4.6 Transition Function	32
3.4.7 Rewards	32
3.4.8 Observation Space	32
3.4.9 Observation Model	32
3.5 Implementation	33
3.5.1 System Architecture	33
3.5.2 Experimentation Setup	34
3.6 Results	35
3.6.1 Location Uncertainty	35
3.6.2 Orientation Uncertainty	36
3.7 Conclusions	36
4 UAV Framework for Autonomous Onboard Navigation and People/Object Detection in Cluttered Indoor Environments	40
4.1 Introduction	42
4.2 Background	45
4.2.1 Partially Observable Markov Decision Processes	45
4.2.2 Augmented Belief Trees	45
4.3 System Architecture	46
4.4 Framework Implementation	47
4.4.1 UAV Frame and Drivers	47
4.4.2 Operating Systems and Middleware	48
4.4.3 Computer Vision Module	48
4.4.4 Decision-making Module	49
4.5 Experiments	55
4.5.1 Environment Setup	55
4.5.2 POMDP Problem Formulation	57
4.6 Results	59
4.7 Discussion	65
4.8 Conclusions and Future Work	66
5 Drone-based Autonomous Motion Planning System for Outdoor Environments under Object Detection Uncertainty	73
5.1 Introduction	75
5.2 System Architecture	77
5.3 Motion Planner Design	79
5.3.1 Assumptions	80

5.3.2	Action Commands	81
5.3.3	States	81
5.3.4	UAV Motion Model	82
5.3.5	Reward Function	82
5.3.6	Observations	85
5.3.7	Observation Model	85
5.4	Experiments	87
5.4.1	Environment Setup	87
5.4.2	Victim Locations	88
5.4.3	Hardware	89
5.4.4	Software and Communications	89
5.4.5	UAV Flight Modes	90
5.5	Results	93
5.6	Discussion	98
5.7	Conclusions	100
6	Reducing Object Detection Uncertainty from RGB and Thermal Data for UAV Outdoor Surveillance	105
6.1	Introduction	107
6.2	Framework Design	108
6.2.1	UAV Airframe and Payloads	108
6.2.2	Vision Module	109
6.2.3	Mapping Module	110
6.2.4	Planner Module	110
6.2.5	Communication Interface	110
6.3	Planner Design	110
6.3.1	Assumptions	110
6.3.2	Actions	110
6.3.3	States	111
6.3.4	Transition Function	111
6.3.5	Reward Function	111
6.3.6	Observations	111
6.3.7	Observation Model	112
6.4	Experiments	112
6.4.1	Location and Environment Setup	112
6.4.2	Flight Modes	113
6.4.3	POMDP solver	114
6.5	Results and Discussion	114
6.6	Conclusions	118
7	Autonomous Mapping of Desiccation Cracks via a Probabilistic-based Motion Planner On-board UAVs	123
7.1	Introduction	125
7.2	Framework Design	126
7.2.1	UAV and Sensors	127
7.2.2	Semantic Segmentation Model	127
7.2.3	Communication Interface	128
7.3	POMDP Motion Planner Design	128

7.3.1 Assumptions	128
7.3.2 Actions	129
7.3.3 States	129
7.3.4 Transition Function	129
7.3.5 Reward Function	129
7.3.6 Observations	129
7.3.7 Observation Model	130
7.4 Experiments	130
7.4.1 Environment Setup	130
7.4.2 Flight Modes	130
7.4.3 POMDP solver	130
7.5 Results and Discussion	131
7.6 Conclusions	136
8 Conclusions	139
8.1 Research Findings	140
8.2 Recommendations	144
8.3 Future Research Avenues	145
References	147

List of Figures

1.1	Autonomous navigation capabilities onboard small unmanned aerial vehicles (UAVs) for obstacle avoidance, exploration, and object finding in partially observable environments.	2
1.2	Factors that can induce object detection uncertainty during UAV surveys in outdoor environments.	4
2.1	Small UAVs establishing situational awareness of challenging and dangerous environments.	12
2.2	Chain diagram of actions, states, and rewards between the agent and the environment in MDPs.	17
2.3	Chain diagram of actions, observations, and belief states between the agent and the environment in POMDPs.	19
2.4	Belief tree using POMDPs.	21
3.1	Illustration of the small UAV navigation task to find people in a cluttered environment.	31
3.2	FOV and 2D image representation from a vision-based camera pointing to the ground with an angle α from the vertical.	34
3.3	Proposed system architecture under a simulation environment.	34
3.4	Detection example of a child dummy from a downward-looking camera under Gazebo SIL.	35
3.5	SIL environment illustration that overlays an occupancy map (in green) and the belief map (in red).	35
3.6	Time steps of successful runs under location uncertainty.	36
3.7	Dummy child at different orientation configurations.	36
3.8	Time steps of successful runs under orientation uncertainty.	36
4.1	Proposed modular system architecture for an autonomous onboard navigation in Global Navigation Satellite System (GNSS)-denied environments in small Unmanned Aerial Vehicles (UAVs).	46
4.2	Proposed UAV frame with mounted drivers, companion computer and payload.	48
4.3	Victim detection trial using the Google MobileNet Single-Shot Detector (SSD) architecture.	49
4.4	Example of a cluttered indoor environment for UAV Search and Rescue mission.	50
4.5	Example of a recorded trajectory.	52
4.6	Two-dimensional (2D) projection of vision-based sensors pointing to the ground.	54
4.7	Environment setup for autonomous UAV victim finding in HIL simulations and real-flight tests.	56
4.8	Environment setup for hardware experiments.	56
4.9	Tested case studies of search and rescue situational awareness regarding the victim location.	58

4.10 Flight mission examples of most frequent types of UAV trajectories under a single belief cluster.	60
4.11 Flight mission example trajectory under two defined belief clusters.	61
4.12 Example global trajectories followed by the UAV under a uniform belief cluster in flight tests.	61
4.13 Example traversed path in a real flight test if a victim is undetected under a uniform belief cluster.	62
4.14 Stability anomalies on the traced UAV trajectory during real flight tests.	63
4.15 Data distribution of executed time steps by the POMDP solver.	64
4.16 Anomalies on the amount of executed time steps in simulation.	65
5.1 Factors that can cause object detection uncertainty and partial observability during UAV surveys.	77
5.2 System architecture for applications with rich global navigation satellite system (GNSS) coverage.	78
5.3 Interaction between the UAV (or agent) and the environment under the framework of Markov decision processes (MDPs).	79
5.4 Belief tree and motion policy representation in partially observable MDPs (POMDPs).	80
5.5 Example of a traversed path by the UAV and its corresponding footprint map.	85
5.6 Field of view (FOV) projection and footprint extent of a vision-based sensor.	86
5.7 QUT Samford Ecological Research Facility (SERF) virtual environment setup.	88
5.8 Top-down visualisation and detection challenges of the victim placed at location L2.	89
5.9 Three-dimensional (3D) occupancy map from the Hovermap LiDAR data of The Samford Ecological Research Facility (SERF).	90
5.10 Designed flight plan that the UAV follows in mission mode.	91
5.11 Probabilistic distribution of initial UAV and victim position belief across the survey extent.	92
5.12 Probabilistic distribution of initial UAV and victim position belief under hybrid flight mode.	93
5.13 Heatmaps from recorded GNSS coordinate points from detections in mission mode.	94
5.14 GNSS coordinates' heatmaps of confirmed victim detections by the POMDP-based motion planner in offboard flight mode.	94
5.15 GNSS coordinates' heatmaps of confirmed victim detections by the POMDP-based motion planner in hybrid flight mode.	94
5.16 Most common object visualisations that triggered false positive readings of people in mission mode.	95
5.17 Distribution frequency of the elapsed time until the POMDP-based motion planner confirmed a detected victim in a trivial (L1) and complex (L2) locations.	96
5.18 Illustration of two traversed paths using the proposed POMDP-based motion planner.	97
5.19 Demonstration of the presented UAV system in hardware with real flight tests at QUT SERF.	98
6.1 UAV navigating in environments under uncertainty and partial observability.	107
6.2 Modular system architecture for autonomous navigation onboard UAVs in uncertain outdoor environments.	109
6.3 Framework implementation in a sub 2 kg quadrotor UAV.	109
6.4 GoPro Hero 9 mounted onto an anti-vibration bracket, pointing to the ground and in parallel to Earth's nadir.	109
6.5 Field of View (FOV) projection and footprint extent of a vision-based sensor.	112

6.6	Location of conducted flight tests at the Samford Ecological Research Facility (SERF), QLD, Australia.	113
6.7	Adult mannequin placed in the surveyed area as the victim to be found.	113
6.8	Executed flight modes for exploration and object detection in outdoor environments.	114
6.9	Strong winds distorting RGB streaming in offboard mode.	115
6.10	Heatmaps of recorded GNSS coordinates in a trivial victim location (L1).	116
6.11	Heatmaps of recorded GNSS coordinates in a complex victim location (L2).	116
6.12	Reduction of object detection uncertainty from RGB camera after executing the motion policy.	117
6.13	Example traversed path in offboard mode while no victims are found.	117
6.14	Reduction of object detection uncertainty from thermal camera after executing the motion policy.	118
6.15	Flight plan following a lawnmower pattern using the RGB camera properties from Table 4.	119
7.1	Environments with indicators of water-related life signatures (or biosignatures).	125
7.2	Framework architecture for autonomous navigation onboard UAVs in uncertain outdoor environments.	127
7.3	Sub 2 kg UAV framework implementation for autonomous detection and mapping of desiccation cracks.	127
7.4	OAK-D camera mounted to an anti-vibration mount, down-facing and in parallel to Earth's nadir.	127
7.5	Inference preview of ResNet18 CNN model for real-time semantic segmentation of desiccation cracks.	128
7.6	Test setup at the Samford Ecological Research Facility (SERF), Australia.	131
7.7	Drafted flight plan in QGroundControl following a lawnmower pattern using the OAK-D camera properties from Table 3.	131
7.8	Executed flight modes for detection and mapping of desiccation cracks.	131
7.9	Comparison of recorded GNSS coordinates of desiccation cracks for UAV surveys at 10 m above ground level (AGL).	132
7.10	Comparison of recorded GNSS coordinates of desiccation cracks for UAV surveys at 20 m AGL.	132
7.11	Traversed path by the UAV in hybrid mode to inspect an area with a potential desiccation crack.	133
7.12	Traversed path by the UAV in hybrid mode to inspect an area from a false positive detection.	133
7.13	Comparison of elapsed time of the UAV in hybrid mode to complete the survey at 10 m and 20 m.	134
7.14	Level of granularity of projected processed frames into a point cloud for mapping.	134
7.15	Level of granularity in a Quadtree map of a segmented crack texture in hybrid (top) and mission (bottom) flight modes.	135

List of Tables

1.1	Thesis structure and chapter (Ch.) contributions to research questions (RQs) from published papers.	10
2.1	Classification of UAVs based on their maximum take-off weight (MTOW) and ground impact risk T_{GI} , defined as the minimum time between ground impact accidents.	12
2.2	Classification approach of autonomous navigation methods for UAVs.	14
2.3	Literature review summary of autonomous decision-making methods for autonomous navigation under environment and object detection uncertainty onboard small UAVs.	25
3.1	Small UAV set of actions.	32
3.2	System rewards for a small UAV target finding task.	32
3.3	Location uncertainty success rates under SIL.	35
3.4	Success rates for defined levels of target uncertainty under SIL.	36
4.1	Set of chosen actions for the problem formulation.	51
4.2	Hyperparameter values for the POMDP formulated problem.	59
4.3	Performance metrics for Hardware in the Loop (HIL) simulations (S) and real flight tests (FT).	59
5.1	Set of action commands comprised of local position commands referenced to the world coordinate frame.	81
5.2	Applied reward values to the reward function R, defined in Algorithm 1.	83
5.3	Flight plan parameters for mission mode.	91
5.4	Initial conditions set to the POMDP motion planner.	92
5.5	Accuracy and collision metrics of the system to locate a victim at two locations (L1 and L2).	95
5.6	Average duration among flight modes to locate the victim.	97
6.1	Applied reward values to the reward function R, defined in Algorithm 1.	111
6.2	Accuracy metrics of the system to locate a victim at two locations (L1 and L2).	115
6.3	Elapsed time by the UAV to locate a victim at two locations (L1 and L2) per flight mode.	117
6.4	Flight plan parameters based on RGB camera properties.	119
6.5	Set of hyper-parameters used in TAPIR and initial conditions to operate the UAV in offboard and hybrid flight modes.	120
7.1	Applied reward values to the reward function R, defined in Algorithm 1.	129
7.2	Accuracy metrics of the proposed framework in mission and hybrid flight modes.	135
7.3	Flight plan parameters from the OAK-D sensor properties.	136
7.4	Set of hyper-parameters used in the TAPIR toolkit for flights in hybrid mode.	136

List of Publications

Published/Accepted Chapter Papers

- (Chapter 3): J. Sandino, F. Vanegas, F. Gonzalez, and F. Maire, “Autonomous UAV navigation for active perception of targets in uncertain and cluttered environments,” in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2020, pp. 1–12. DOI: [10.1109/AERO47225.2020.9172808](https://doi.org/10.1109/AERO47225.2020.9172808).
- (Chapter 4): J. Sandino, F. Vanegas, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez, “UAV framework for autonomous onboard navigation and people/object detection in cluttered indoor environments,” *Remote Sensing*, vol. 12, no. 20, p. 3386, Oct. 2020. DOI: [10.3390/rs12203386](https://doi.org/10.3390/rs12203386).
- (Chapter 5): J. Sandino, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez, “Drone-based autonomous motion planning system for outdoor environments under object detection uncertainty,” *Remote Sensing*, vol. 13, no. 21, p. 4481, Nov. 2021. DOI: [10.3390/rs13214481](https://doi.org/10.3390/rs13214481).
- (Chapter 6): J. Sandino, P. Caccetta, C. Sanderson, F. Maire, and F. Gonzalez, “Reducing object detection uncertainty from rgb and thermal data for UAV outdoor surveillance,” in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2022, (accepted).
- (Chapter 7): J. Sandino, J. Galvez-Serna, N. Mandel, F. Vanegas, and F. Gonzalez, “Autonomous mapping of desiccation cracks via a probabilistic-based motion planner onboard UAVs,” in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2022, (accepted).

Accepted Non-chapter Papers

The following publications were also generated as part of the work of this PhD, and further validate the robustness and scalability opportunities of the presented UAV framework in other remote sensing fields. However, these papers are not explicitly included in any of the chapters of this thesis.

- N. Mandel, J. Sandino, J. Galvez-Serna, F. Vanegas, and F. Gonzalez, “Resolution-adaptive quad-trees for semantic segmentation mapping in UAV applications,” in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2022, (accepted).
- J. Galvez-Serna, N. Mandel, J. Sandino, F. Vanegas, N. Ly, D. Flannery, and F. Gonzalez, “Real-time segmentation of desiccation cracks onboard UAVs for planetary exploration,” in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2022, (accepted).

List of Acronyms

2D	two-dimensional
3D	three-dimensional
ABT	adaptive belief tree
AMPLE	anytime meta planner
BN	bayesian network
CNN	convolutional neural network
DEM	digital elevation model
DRL	deep reinforcement learning
DSM	digital surface model
FCU	flight controller unit
FL	fuzzy logic
FOV	field of view
GA	genetic algorithm
GNSS	global navigation satellite system
GPU	graphical processor unit
HIL	hardware in the loop
LiDAR	light detection and ranging
MAVLink	micro air vehicle link
MDP	Markov decision process
MOMDP	mixed observability Markov decision process
MTOW	maximum take-off weight
OAK	OpenCV AI kit
POMCP	partially observable Monte Carlo planning
POMDP	partially observable Markov decision process
PSO	particle swarm optimisation
QUT	Queensland University of Technology
RGB	red, green, blue
ROI	region of interest

ROS	robot operating system
RQ	research question
SAR	search and rescue
SDP	sequential decision process
SIL	software in the loop
SIP	system identification process
SLAM	simultaneous localisation and mapping
TAPIR	toolkit for approximating and adapting POMDP solutions in real time
UAV	unmanned aerial vehicle
VPU	vision processing unit

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at the Queensland University of Technology or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signed,

[QUT Verified Signature](#)

Juan David Sandino Mora

Date: 2022-05-20

Acknowledgements

THIS dissertation became a reality only because of the priceless and generous support of my supervisors, research colleagues, the Queensland University of Technology (QUT) technical and administrative staff, relatives, and friends. I am immensely grateful to my principal supervisor Professor Felipe Gonzalez for his continuous dedication, encouragement, and patience in mentoring me to become a better researcher and accomplish this research. I would like to thank my associate and external supervisors (Dr Frederic Maire, Dr Peter Caccetta and Dr Conrad Sanderson) for their valuable insights, critical thinking, and feedback to increase the quality of this thesis and published papers. Special thanks to professional editor, Karyn Gomano (KLG Communications), who provided copy editing and proofreading services, according to the guidelines laid out in the university-endorsed national “Guidelines for editing research theses”.

I am pleased to extend my sincere gratitude to my work colleagues at QUT, who boosted the execution of the experimental phases of this study, namely: Dr Fernando Vanegas for his patience, theoretical discussions, and technical advice and training on the development and execution of frameworks for autonomous decision-making onboard small UAVs; Nicolas Mandel for his generous early-bird support during the experimental phase with UAVs and thermal imagery in outdoor environments (starting from 4:30 am at 7°C); and, Julian Galvez-Serna for his endless and altruistic cooperation conducting flight tests in outdoor environments. Special thanks to Sharlene Lee-Jendili and Levi Hunter for sharing their vision-based people detector for thermal flight tests. My deepest appreciation is for my parents, relatives and friends, who encouraged me to be resilient to undertake and complete this research endeavour.

I acknowledge the ongoing support from QUT through The Centre for Robotics. Special thanks to the Samford Ecological Research Facility (SERF) team (Marcus Yates and Lorrelle Allen) for their continuous assistance and equipment provided during flight tests. I thank the QUT Research Engineering Facility (REF) team (Dr Dmitry Bratanov, Gavin Broadbent, Dean Gilligan) for their technical support, making it possible to conduct test flights. Special thanks to Hexagon through the Hexagon SmartNet RTK corrections service that enabled high accuracy surveying and positioning data using the EMLID Reach RTK receiver during the experimental phase.

I am immensely grateful for the following institutions which sponsored this research through research grants and scholarships: 1) the Commonwealth Scientific and Industrial Research Organisation (CSIRO) through the CSIRO Data61 PhD and Top Up Scholarships (Agreement 50061686); 2) the Australian Research Council (ARC) through the ARC Discovery Project 2018 “Navigating under the forest canopy and in the urban jungle” (grant number ARC DP180102250); and, 3) QUT through the Higher Degree Research (HDR) Tuition Fee Sponsorship.

Chapter 1

Introduction

THIS chapter introduces the concepts behind the research undertaken to enable autonomous decision-making for small UAVs operating under environment and object detection uncertainty. The sections below discuss the background of autonomous navigation capabilities of small UAVs for time-critical applications, the presence of object detection uncertainty in UAV flight operations, and current decision-making limitations of small UAVs to autonomously navigate in uncertain environments. The research problem, which presents the research questions, aims and scope, is then presented, followed by the research significance and the structure of this thesis.

1.1 Background

Unmanned aerial vehicles (UAVs)—also known as drones—are aerial platforms whose use has grown exponentially in remote sensing and time-critical applications [1]. UAVs are currently employed in fields such as humanitarian relief, disaster monitoring, biosecurity, surveillance, and wildlife monitoring [2]–[5]. It is because of their reduced cost and dimensions, extensive adaptability to attach sensors and cameras, and user-friendly navigation capabilities including obstacle avoidance, take-off and landing, and waypoint navigation, that UAVs offer high level of versatility to survey areas of interest [6]–[8].

UAVs have been used particularly in applications that require onboard rapid decision-making and data processing of vision-based payloads (or cameras). These types of survey needs are referred to as *time-critical* applications, and are found in real-world emergency environments such as search and rescue (SAR) and disaster monitoring. In these applications, the identification, localisation, and quantification of objects (*i.e.*, victims and structures) is vital for first responders to coordinate and prioritise response efforts in affected, or concerned zones. Operating *small UAVs*, those with a maximum take-off weight (MTOW) of up to 13.5 kg [9], in time-critical applications such as SAR is a challenging task as these environments are, in most of the cases, dangerous and unexplored to first responders (*i.e.*, rescue squads). The limitations of the sensors equipped in small UAVs cause partial observability in surveyed environments, making it difficult to detect objects of interest (*i.e.*, victims), obstacles, hazardous structures, and to accurately perceive the dynamics and position of the aircraft. Partial observability also limits UAVs to predict unexpected situations in real-world environments such as unexpected wind gusts, weather events, unstable or weak global navigation satellite system (GNSS) signal coverage, and external hazards. Real-world indoor and outdoor environments are highly complex areas for UAVs to autonomously, and in some cases even manually navigate, owing to the high level of uncertainty.

Uncertainty is an issue when planning and monitoring navigation strategies to survey areas with UAVs [10]. Ideally, a fully autonomous small UAV should adjust its motion planning strategy while interacting with the environment, as shown in Figure 1.1. This iterative interaction between the UAV and the environment reduces partial observability and increases the likelihood of obtaining better representations of the objects it aims to detect (*i.e.*, victims). Then, modifications to the motion planning might occur when there are changes, or updates, in either the state of environment or their objects. Those mid-flight trajectory adjustments enable small UAVs to have a higher level of adaptability in uncertain environments, and ultimately, more accurate statistics of any detected victims in unattended survey missions.



Figure 1.1: Autonomous navigation capabilities onboard small unmanned aerial vehicles (UAVs) for obstacle avoidance, exploration, and object finding in partially observable environments. The path planning strategy (Path A) is subject to mid-flight updates (Paths B or C) as the UAV interacts with the environment and detects potential victims.

In time-critical applications, real-time streaming of camera frames is essential for human pilots to manoeuvre small UAVs, and understand the context and complexity of the mission to coordinate successful intervention strategies [11]. However, relying heavily on communication systems to control the UAV could compromise their integrity, and even trigger a safety risk to others if those systems fail [12]. A second issue with overuse of communication systems is when human pilots are reported to experience long-term fatigue after prolonged photo-interpreting of streamed frames to search for victims and manoeuvre small UAVs [13]. Therefore, incorporating increased cognitive power onboard UAVs to autonomously navigate in complex environments under uncertainty could further extend their contribution in civilian and time-critical applications. Small UAVs with onboard object detection capabilities for real-time decision-making dramatically reduce the need of photo-interpreting streamed frames, and reliance on human operators to control single and multiple UAVs.

Recent research on small UAVs for autonomous navigation has shown promising results about object finding under uncertain environments using onboard sequential decision processes (SDPs) in simulated environments only [14]–[16], and are yet to be validated in real small UAV scenarios. A lack of real-world testing in these systems is mostly due to the following three key challenges of:

1) increased difficulties to detect complex objects, such as victims, from streamed camera frames exposed to factors such as poor illumination, strong vibrations, and occlusion; 2) performance issues when running SDPs under resource-constrained hardware onboard small UAVs; and 3) lack of robust navigation systems onboard small UAVs when dealing with object detection uncertainty from partial observability and false positive readings from convolutional neural network (CNN) models. These three key challenges remain because of the way data is acquired in real-world applications. For example, data collected from surveyed environments is partial, non-existent, or prone to inaccuracies owing to the limitations of small UAV sensors. Little research has been conducted in the area of uncertainty in terms of predictions from vision-based object detectors. Conversely, many approaches have focused their efforts to improve confidence metrics from CNN model predictions. No work has yet been undertaken on autonomous small UAV decision-making for object detection to model or estimate object detection uncertainty.

The focus of this research therefore is to increase the autonomy level and cognition of small UAV systems under environment and object detection uncertainty using vision-based sensors and onboard computer vision. This research investigated how the modelling of an SDP and its execution, onboard small UAVs, improves their robustness to deal with high levels of environment and object detection uncertainty.

1.1.1 Time-critical Applications

Employing small UAVs with onboard cognition to interact with surveyed environments for humanitarian relief and SAR has been extensively explored. The key goal for first responders is to enable quick localisation, identification, and quantification of victims, and to identify and prioritise an emergency response in the affected zones. Traditionally, small UAV operators photo-interpret real-time footage from cameras attached to such drones to then plan and decide on the next sequence of navigation commands [17]. However, the prolonged use of these visual systems is claimed to produce fatigue and sensory overload. Despite recent advances being made to small UAVs to more autonomously navigate for SAR purposes [16], [18], [19], there are still three key challenges, namely, the lack of: 1) solid tested platforms in real-world environments; 2) methods to identify, locate, and quantify non-trivial objects (*i.e.*, victims); and 3) autonomous navigation capabilities running onboard small UAVs to interact in environments and increase the level of confidence in the detection of objects from vision-based models.

Disaster monitoring is the rapid assessment of economic, environmental and humanitarian losses in affected urban, peri-urban and rural regions, and is fundamental to attenuate the impacts of natural disasters (*e.g.*, landslides, floods, volcanic eruptions, wildfires, nuclear disasters and earthquakes). Previous research has also identified limitations of small UAVs to deal with unexpected mid-flight events in disaster monitoring tasks because of the lack of autonomy [20], [21]. Most UAV approaches for disaster monitoring require the use of time-consuming data processing techniques post-flight, such as data filtering, ortho-rectification, image mosaicking, and generation of elevation models (*e.g.*, digital surface model (DSM) and digital elevation model (DEM)) [22]. Since many flight operations are limited in terms of providing quick insights about the status of surveyed areas or presence of objects of interest, small UAV technology remains limited for broader scale operations.

The capabilities of small UAVs to make a decision autonomously for most applications in uncertain environments is limited. While these decision-making problems can be solved using SDPs, where the small UAV decides its next finite sequence of actions (*e.g.*, update the heading of the UAV, navigate to a new position waypoint, hover itself or track a static or dynamic object mid-flight) based on data

collected from the environment, limited computing power in resource-constrained hardware adds another layer of difficulty to simultaneously run motion planning algorithms, real-time streaming of camera frames, and onboard inference of CNN models. Therefore, small UAV frameworks that enable autonomous decision-making under uncertainty and partial observability onboard small UAVs need to be developed and validated.

1.1.2 Types of Object Detection Uncertainty

Partial observability in UAV flight operations is the limitation of obtaining complete and accurate information regarding: 1) objects in the environment expected to be identified, located and quantified; 2) the UAV itself (*e.g.*, data about its location, pose and dynamics); and 3) the state of the environment. Partial observability, an ever-present issue in real-world applications causing uncertainty in surveyed environments, occurs mostly because of imperfect sensor readings equipped in small UAVs, poor visual representations of objects from streamed camera frames, suboptimal surveying altitudes, and external disturbances from the environment. Factors that can cause object detection uncertainty during UAV surveys are partial or entire occlusion of the object with taller obstacles, viewpoint and scale variations, poor illumination conditions, and excessive vibrations in the UAV frame, as illustrated in Figure 1.2.

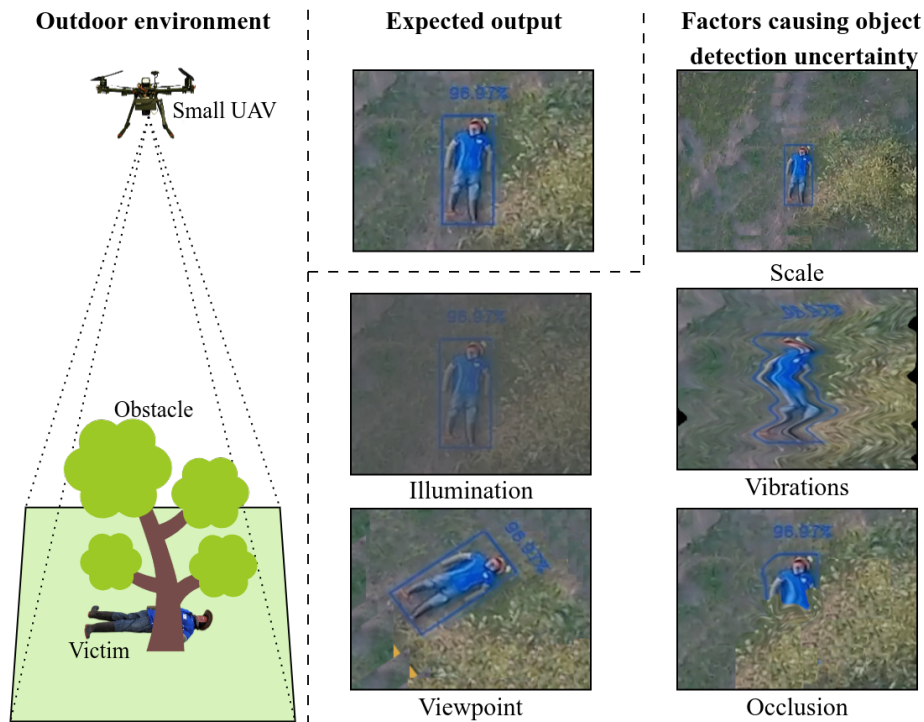


Figure 1.2: Factors that can induce object detection uncertainty during UAV surveys in outdoor environments. These factors are caused by limitations such as imperfect sensors readings, poor visual representations from streamed camera frames, suboptimal surveying altitudes, and external disturbances from the environment.

The absence of absolute or holistic visual representations of objects from vision-based cameras and computer vision algorithms onboard UAVs introduce uncertainty, which can be classified as follows:

- **Identification:** limitations are those of recognising a specific identification property of the object. For instance, identification uncertainty on detected victims include, but are not limited to, age, gender, and health conditions. Identification uncertainty could also cover situations

where various objects are detected, but it becomes challenging discriminating against assigning, or labelling each detected object with the correct class (*e.g.*, mislabelling victims with trees, chairs, bottles, etc.)

- **Localisation:** of objects becomes an uncertainty in many flight operations using small UAVs due to the limited extent of vision-based sensors to capture images that represent the extent of a scene, also known as field of view (FOV). These limitations are common in mapping, surveillance and other object finding applications in cluttered and challenging environments. The location of victims in SAR operations, for example, is challenging when vision-based sensors are unable to cover specific FOV configurations (*e.g.*, viewpoints using non-nadir or side-by-side settings) or when victims are partially occluded by an urban structure or taller obstacles, the effects of the disaster situation (*e.g.*, fire smokes, earthquakes, floods) or natural events (*e.g.*, fog, cloudy and wet conditions). Likewise, object location uncertainty in surveillance operations includes unexpected changes in object pose and behaviour (dynamics), occlusion and even loss of object perception while tracking it.
- **Quantification:** as an uncertainty applies where objects are partially occluded among themselves or between themselves and environmental obstacles, or environmental factors that distort the image quality of vision-based sensors. Quantification is important in humanitarian relief operations, for instance, when rescuers require prioritising emergency response by counting victim numbers. The collection of these statistics determines the most critical zones for immediate intervention.
- **Morphology:** as an uncertainty occurs when the surveying application requires collecting the morphological properties of objects, namely length, volume, or height. Such properties in SAR operations could help understand complementary rescue conditions and define resources and processes needed to assist victims.

Another source of uncertainty arises when computer vision algorithms, for object detection return incorrect data. Developed algorithms use varied strategies to detect objects [23], such as: 1) classical image processing pipelines (*e.g.*, kernel filtering, binary and colour thresholding, edge and contour detection, and morphological operations); 2) feature extraction (*e.g.*, SIFT, SURF); 3) adaptive filtering (*e.g.*, Kalman filters); 4) visual odometry (*e.g.*, simultaneous localisation and mapping (SLAM)); and 5) deep learning (*e.g.*, Region-based and Single Shot Detector CNNs). Regardless of the algorithms and methods used, object detector models are not exempt from uncertainty by their given predictions (or inference). In fact, detection errors may occur when sensor measurements lack accuracy and precision. Formulating and implementing SDPs for autonomous decision-making in small UAVs should consider uncertainty from vision-based object detectors, and adapt their motion strategy when output detections contain substantial imprecisions. With increased cognitive power for autonomous navigation, small UAVs can interact with surveyed environments to confirm or discard positive detections from CNN models, rather than spending significant effort and resources collecting new data and retraining such models.

1.2 Research Problem

First responder resources for SAR and disaster management are critical for emergency services to react effectively and diminish fatalities whenever possible. Natural disasters are events claim human lives globally and annually, and the 2011 and 2018 Queensland floods are examples of events that impacted Australia socially and economically. Emergency events can also be triggered from accidents, with people lost in the bush, river bays and beach shores. An average of 38,000 people are reported missing in Australia annually, and from this number, 720 are never found and 2600 are reported (long-term) missing for more than three months [24]. The potential for intelligent surveying technologies such as small UAVs is predicted to accelerate situational awareness in SAR operations and, therefore, improve decision-making to optimise intervention strategies and increase the likelihood of rescuing victims.

The inclusion of robotic technologies for disaster monitoring and SAR has positively impacted rescue efforts by achieving higher success in their operations [25]–[28]. Such rescue efforts could be even more successful when researchers address the several challenges that still persist [7]. Most small UAVs are limited in terms of collecting absolute data, or observations of surveyed environments and objects within them, due mostly because of imperfect aircraft sensor systems, reduced information bandwidth, high complexity to inspect areas with partly occluded objects, and object detection uncertainty from computer vision algorithms. The design of UAV frameworks for autonomous on-board decision-making in time-critical and emergency environments is a challenging problem in the resource-constrained hardware of small UAVs.

The benefits of formulating problems with SDPs on UAV exploration and object finding for autonomous decision-making under uncertainty have shown the potential of small UAVs to interact in cluttered and challenging environments with unpredictable changing events, and to localise, identify and quantify objects under partial observability. What it is required to make small UAVs work efficiently and effectively is to formulate and implement SDPs under object detection uncertainty running onboard resource-constrained hardware, and validation with real UAV flight tests. That is, the modelling of object detection uncertainty from vision-based sensors. Using an SDP and framework capable of running in sub-2 kg UAVs for time-critical applications will be applicable for small UAVs to autonomously interact and operate effectively, and inspect zones to confirm or discard the identification and location of potential objects under high levels of environment and object detection uncertainty. This research therefore aims to create a fully autonomous UAV system that can operate in real-world environments under uncertainty and partial observability.

1.2.1 Research Questions

The two primary research questions (RQs) and five sub-questions are as follows:

Research Question 1

How can an autonomous navigation problem for small UAVs be formulated as a sequential decision process (SDP) under high levels of environment and object detection uncertainty, so it can account for a solution that works in real time for time-critical applications?

The three sub-questions therefore are:

- 1.1. What level of artificial cognitive learning and uncertainty modelling is required to identify,

localise or quantify objects positioned in cluttered and challenging scenarios using small UAVs and vision-based object detectors?

- 1.2. Which factors define the complexity of an SDP so as to reduce object detection uncertainty from collected environment observations using vision-based sensors in time-critical applications?
- 1.3. What are the modelling considerations in a formulated SDP that enable the scalability of a UAV framework for autonomous navigation for a range of diverse vision-based payloads, and remote sensing application needs beyond object detection?

Research Question 2

How can the decision-making framework best be designed to that it can enable small UAVs to explore and find objects, so as to automate flight surveys and data collection campaigns in environments under uncertainty and partial observability?

The two sub-questions therefore are:

- 2.1. How can computationally intensive tasks such as online SDP algorithms, and onboard inference of CNN models for object detection and segmentation, be integrated to run simultaneously under resource-constrained hardware in sub-2 kg UAVs?
- 2.2. What are the design criteria to scale a UAV framework for autonomous navigation under environment and object detection uncertainty to other remote sensing applications that require autonomous decision-making capabilities onboard small UAVs?

1.2.2 Aims of the Study

The overall aim of this research is to increase the cognitive power of small UAVs by incorporating an SDP into resource-constrained hardware, so the small UAV can navigate autonomously under environment and object detection uncertainty. This aim can be achieved through the following six objectives:

1. Model a system architecture for small UAVs using SDPs for autonomous navigation in environments with high levels of environment and object detection uncertainty.
2. Deploy an SDP onboard a small UAV for autonomous motion planning and interaction with environments under uncertainty and partial observability in real-time.
3. Deploy vision-based sensors, and object detection and image segmentation CNN architectures for onboard inference in embedded systems present in sub-2 kg UAVs.
4. Develop a simulation environment that provides high-fidelity performance metrics to evaluate the potential of developed SDPs for real flight tests using hardware mounted in sub-2 kg UAVs.
5. Design and validate a modular sub-2 kg UAV framework robust against the various types and complexity levels of object detection uncertainty present during flight surveys.
6. Apply the design characteristics of the UAV framework in multiple real-world applications that require real-time object detection and onboard data processing in small UAVs.

1.2.3 Scope

The eight items listed below identify the scope of this research:

- **Number of objects:** multiple instances of objects could visually appear in an image frame. This research limited the design of the SDP to reduce levels of object detection uncertainty assuming a single object appears in processed frames. If a frame depicts multiple objects, the object detector will only provide detection metrics from the object with the highest level of confidence and will discard others.
- **Object dynamics:** this research evaluated levels of object detection uncertainty with static, and not dynamic objects nor those with object tracking capabilities.
- **Situational awareness:** UAV flight operations are rarely conducted under blind conditions. Instead, a minimum set of situational awareness conditions are commonly known to human pilots before operating UAVs in indoor and outdoor environments. During the experimental phase, initial conditions known beforehand included the flight extent (or geo-fence), minimum and maximum surveying altitudes, the takeoff location of the UAV and the three-dimensional (3D) occupancy map of the environment.
- **Environmental setup:** UAV flight operations were conducted mostly on flat terrains in indoor and outdoor environments. For safety and economic reasons, this research validated UAV framework implementations to daytime tests using red, green, blue (RGB) cameras as vision-based payloads. Chapter 6, however, describes preliminary flight tests results using thermal cameras to demonstrate their suitability in poor illumination conditions over high-resolution RGB cameras.
- **Selection of SDP:** this research explored and exploited the performance capabilities of partially observable Markov decision processes (POMDPs) and model-based POMDP solvers, only. Details on the selection criteria of POMDPs against other algorithms is established in Chapter 2.
- **Collection of observations:** observations of the environment came from real-time processing outputs of streaming of vision-based sensors and convolutional neural network (CNN) architectures for object detection and image segmentation. The UAV framework established in this research captured vision-based observations from a singular camera, only. Observations of the local position of the UAV are sourced from internal position estimations of the UAV autopilot.
- **UAV hardware:** only quadrotor UAVs were used in this investigation. The overall weight of the UAV did not exceed 2 kg, and the maximum diameter of the aircraft was equal to or less than 0.5 m.
- **UAV software:** current programming software solutions were constrained to open-source software to ensure ease of access and reproducibility in future research. The selection of software autopilots did not include commercial solutions from manufacturers such as DJI and Parrot.

1.3 Research Significance

The majority of real-world indoor and outdoor environments are highly dynamic, complex, and full of uncertainties. Enabling autonomous decision-making for exploration and object finding in environments under uncertainty will enhance the demand for applications with limited support [29],

including surveillance operations that require rapid response and deployment of resources, such as SAR, disaster monitoring, humanitarian relief, biosecurity, and border protection. Enabling a higher cognitive power will improve decision-making capabilities of small UAVs to collect more accurate statistics of detected objects despite imperfections from vision-based sensors and computer vision algorithms implemented in the application. The contribution of this research will benefit validation processes of UAV systems for real-time object finding when data is complex to interpret for human pilots, and reduce human bias on scouting strategies.

There are remote sensing fields and flight operations where optimisations on flight campaign times for onboard and real-time object detection might not be completely relevant, especially if post-flight data pipelines are needed. Example applications include soil analysis studies using DEMs, multi- and hyper-spectral image classification and mapping through ortho-rectification and geo-referenced mosaics. However, applications that demand real-time analysis of collected airborne data will be highly benefited. Specifically, time-critical tasks that operate under vision-based cameras, where processing outputs are critical for rapid intervention in surveyed environments.

The outcomes of this research are not expected to restrict their use in other fields for autonomous decision-making in small UAVs, such as: *surveillance* (tactical Police surveillance [30]); *biosecurity* (automatic and rapid detection of biosecurity breaches [31], [32] in small UAV operations [4], [5], [33], [34]); *wildlife monitoring* (detection of endangered species [3], [35]); and *aerospace* (autonomous take-off and landing in static [36]–[38] and dynamic environments [39]–[42]).

1.4 Thesis Outline

This thesis presents a collection of scientific papers across five chapters, 3 to 7. These publications are listed as follows:

- (Chapter 3): J. Sandino, F. Vanegas, F. Gonzalez, and F. Maire, “Autonomous UAV navigation for active perception of targets in uncertain and cluttered environments,” in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2020, pp. 1–12. DOI: [10.1109/AERO47225.2020.9172808](https://doi.org/10.1109/AERO47225.2020.9172808).
- (Chapter 4): J. Sandino, F. Vanegas, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez, “UAV framework for autonomous onboard navigation and people/object detection in cluttered indoor environments,” *Remote Sensing*, vol. 12, no. 20, p. 3386, Oct. 2020. DOI: [10.3390/rs12203386](https://doi.org/10.3390/rs12203386).
- (Chapter 5): J. Sandino, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez, “Drone-based autonomous motion planning system for outdoor environments under object detection uncertainty,” *Remote Sensing*, vol. 13, no. 21, p. 4481, Nov. 2021. DOI: [10.3390/rs13214481](https://doi.org/10.3390/rs13214481).
- (Chapter 6): J. Sandino, P. Caccetta, C. Sanderson, F. Maire, and F. Gonzalez, “Reducing object detection uncertainty from rgb and thermal data for UAV outdoor surveillance,” in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2022, (accepted).
- (Chapter 7): J. Sandino, J. Galvez-Serna, N. Mandel, F. Vanegas, and F. Gonzalez, “Autonomous mapping of desiccation cracks via a probabilistic-based motion planner onboard UAVs,” in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2022, (accepted).

The thesis is structured following Table 1.1:

Table 1.1: Thesis structure and chapter (Ch.) contributions to research questions (RQs) from published papers.

Ch.	Topic	Follows	RQs	Contributions
2	Literature review	N/a	N/a	Top-down review of autonomous navigation methods for small UAVs, SDP methods that model uncertainty and partial observability, and current sub 2 kg UAV frameworks for exploration and object finding.
3	Problem formulation indoors (simulation)	Ch. 2	1.1, 2.1	POMDP problem formulation for victim finding in GNSS-denied environments using small UAVs. Modelling of object detection uncertainty at various positions and victim orientations. System tested using software in the loop (SIL). Onboard inference of CNN model achieved via a vision processing unit (VPU).
4	UAV framework outdoors (real tests)	Ch. 3	1.1, 1.2, 2.1	Design of autonomous navigation framework for cluttered indoor environments using a sub-2 kg UAV. Framework validated with hardware in the loop (HIL) and real flight tests. UAV system architecture design and replica of real-world environments from airborne UAV data bridged the gap between simulated and real flight tests.
5	Problem formulation outdoors (simulation)	Ch. 3, 4	1.1, 1.2, 2.2	Extended POMDP problem formulation for exploration and victim finding in outdoor environments using small UAVs. Novel flight mode design (<i>i.e.</i> , hybrid mode) to inspect areas and confirm presence of potential detected victims. System validated in HIL and with preliminary real flight tests.
6	UAV framework outdoors (real tests)	Ch. 4, 5	1.3, 2.1, 2.2	Extended validation of UAV framework for autonomous exploration and object detection in outdoor environments with real tests. Preliminary scalability test of modular framework using RGB and thermal cameras onboard a sub-2 kg UAV.
7	System scalability (real tests)	Ch. 6	1.3, 2.1, 2.2	Scalability study of established UAV framework to autonomously map desiccation cracks from dry lake beds. Modularity of UAV framework validated by integrating a novel CNN model for image segmentation, and an OpenCV AI kit (OAK)-D camera to simultaneously stream RGB frames and perform onboard CNN inference.
8	Conclusions	Ch. 3—7	N/a	Research findings, recommendations, and future work.

Chapter 2

Literature Review

STRATEGIES to mitigate economic, environmental, and human loss, are critical in emergency situations such as natural disasters, terrorist activities and crime, war conflicts, and severe weather events [43], [44]. Of particular concern disaster monitoring and search and rescue (SAR) contexts, is the collection of information to understand the state of affected zones and the location of any victims as fast as possible [45], and it is “the period within 12 to 24 hours of a distress incident is the most critical for the recovery of survivors” [46]. Rapid data collection about the state of affected areas, the *situational awareness*, is therefore critical for SAR squads or first responders to coordinate and prioritise intervention strategies, and increase the likelihood of finding and rescuing victims.

Unmanned aerial vehicles (UAVs) are one key technology to establish situational awareness during emergency situations, with their use including surveying applications related to SAR or disaster monitoring, as well as remote sensing in the fields of precision agriculture, cinematography, environmental monitoring, and biosecurity [2]–[5], [34], [35]. Compared with manned aircraft solutions, UAVs offer a considerably more affordable, flexible and faster scouting capabilities due to advances in microelectronics, sensors and payload portability, computer vision and image processing, and autonomous navigation capabilities. Figure 2.1 for example illustrates UAV capabilities of surveying challenging and dangerous environments for rapid assessment of areas impacted by distressing events.



(a)



(b)

Figure 2.1: Small UAVs establishing situational awareness of challenging and dangerous environments in (a) wild/bush fires and (b) disaster management. Source: [47].

Diverse UAV designs are available to variety of remote sensing applications. Historically manufactured for the military and defence sector, it is the advances in electronics and microelectronics with compacted and affordable sensors, actuators, microcontrollers, that has expanded the use of UAVs for civilian applications. In fact, it is the *small UAVs*, with a maximum take-off weight (MTOW) equal of less than 13.5 kg [9] (Table 2.1), that are being used in distress events (or *time-critical* surveying applications), as well as industry and research communities.

Table 2.1: Classification of UAVs based on their maximum take-off weight (MTOW) and ground impact risk T_{GI} , defined as the minimum time between ground impact accidents.

UAV Name Prefix	MTOW	T_{GI}
Micro	Less than 1 kg	10^2
Mini	Up to 1 kg	10^3
Small	Up to 13.5 kg	10^4
Light/ultraweight	Up to 242 kg	10^5
Normal	Up to 4332 kg	10^6
Large	Over 4332 kg	10^7

The capacity of small UAVs to operate in remote, dangerous and cluttered environments has become the focus of the robotics research community, industry, and defence sectors [23]. However, their use in real-world emergency events remains restricted by six key operational limitations [1], [48], including: (a) onboard computing power; (b) endurance; (c) payload weight; (d) power; (e) sensor resolution and image quality; and, (f) cognition capabilities in stochastic situations. Recent hardware-wise developments including the release of more powerful sensors, embedded computing processors [49] and increased capacity of power banks are likely to improve UAV capacity.

Time optimisation in time-critical surveying applications demand UAVs to be capable of autonomously navigating and detecting structures and victims in real time. Advances in autonomous navigation, as in those available for commercial small UAVs include automated home return at cri-

tical battery levels, reactive collision avoidance, autonomous take-off and landing and active object tracking [6], [7]. Nevertheless, the development of autonomous decision-making onboard small UAVs in environments with high levels of uncertainty, such as SAR and disaster monitoring, is a far more challenging problem [48]. Indeed, decision-making capabilities of small UAVs are still limited when dealing with mid-flight events, path planning and obstacle avoidance under unexplored environments (*i.e.*, SAR applications), or object finding under partial observability [1], [50]. A further challenge for the broader use of small UAVs is defined by computational power restrictions in resource-constrained hardware for onboard execution of algorithms such as object detection, path planning, obstacle avoidance and streaming of high-resolution camera frames [51], [52]. Small UAVs which can make decisions to reduce partial observability and react in real time to changes in the environment are likely to perform effectively in real-world scenarios [53], [54], highlighting the need for further research on decision-making under uncertainty for autonomous navigation onboard small UAVs.

This chapter critically reviews the literature regarding autonomous decision-making onboard small UAVs to navigate under environment and object detection uncertainty. The literature review is organised as follows: Section 2.1 discusses high-level techniques for autonomous small UAV decision-making; Section 2.2 describes small UAV navigation methods under unstructured (or stochastic) environments from which learning-based methods are chosen; Section 2.3, presents the sequential decision processes (SDPs) under the framework of reinforcement learning; Section 2.4 presents the methods that consider and model uncertainty and partial observability on SDPs; A key component of UAV autonomy is the development and validation of a framework to implement SDPs in real-world applications. The focus of this research is on converging in a modular and scalable framework design in resource-constrained hardware contained in small UAVs, specifically, sub-2 kg UAVs; Section 2.5 reviews existing framework approaches to execute formulated SDPs onboard small UAVs in real time; Section 2.6 presents the final analysis, and identifies the research problem and the research questions (Section 1.2.1) and thesis contributions (Chapters 3 to 7).

2.1 Autonomous Navigation Methods in UAVs

Research to develop the autonomous navigation systems for UAVs typically involves the combination of research areas, specifically UAV motion control, path planning and obstacle avoidance. Literature on remote sensing and navigation for UAVs in environments with and without the source of global navigation satellite system (GNSS) signal is extensive [1], [55], with Table 2.2 presenting these navigation methods in distress events using UAVs according to the categories of: (a) graph-based search; (b) sampling-based; (c) potential fields; (d) optimisation-based; (e) swarm-optimisation-based; and, (f) learning-based algorithms.

Graph-based search algorithms represent surveyed environments as grids, where cell values could be declared as *occupied* or *free*. Popular methods include A*, LazyTheta* and D*-lite, algorithms that provide fast searching abilities and can be implemented online. However, recent works have shown that graph-based search methods excel under well-defined, small surveying areas, though the location of the object must be assumed in advance, rendering these methods unsuitable in small UAV applications where the object location is unknown.

Sampling-based algorithms evaluate sets of graph paths as collision-free. Samples of free spaces that connect the start and end locations are acquired. These methods follow a probabilistic approach, so their solutions are approximations of the optimal path, and therefore suitable for complex environments. Popular algorithms used for small UAV applications include probabilistic road map (PRM), and rapidly exploring random trees (RRT). Sampling-based algorithms require the location of an

Table 2.2: Classification approach of autonomous navigation methods for UAVs, adapted [10]. PRM: probabilistic road map, RRT: rapidly exploring random trees, APF: artificial potential fields, MIQP: mixed-integer quadratic programming, BLP: bi-level programming, PSO: particle swarm optimisation, GA: genetic algorithms, DRL: deep reinforcement learning, and MDP: Markov decision processes.

Method	Examples	Advantages	Limitations
Graph-based	A* D*-Lite	Online implementation, fast convergence	Optimal for small areas only
Sampling-based	PRM RRT	Suitable for complex environments, fast convergence	Sub-optimal solution
Potential fields	APF	Simple implementation, fast convergence	Sub-optimal solution
Optimisation-based	MIQP BLP	Optimal solutions under obstacle-dense environments	Complex to implement, high number of parameters
Swarm-optimisation	PSO, Ant Colony	Efficient and fast for multiobjective problems	Sensitive to parameter tuning
Learning-based	GA, DRL, MDP	Suitable for multiobjective problems	Require training phase and full state observability

object to be assumed, making them unsuitable for autonomous navigation in environments with object location uncertainty such as disaster monitoring and SAR applications.

Artificial potential fields (APFs) are used in terms of potential fields, where the small UAV or *agent*, is represented as a moving particle which experiences attraction potential from the object to find (or track) and repulsive potential from obstacles. While APFs are relatively simple to implement and converge fast, they are likely to fall under local minima as potential fields from several obstacle locations could trap the agent.

Optimisation-based algorithms estimate optimal path planning using cost functions and non-linear optimisations to provide optimal trajectories on obstacle-dense environments. However, researchers applying optimisation-based methods require advanced modelling skills to replicate the dynamics of surveyed environments. The number of hyperparameters to tune, and computational costs are directly proportional to the complexity of the formulated problem. Some use of optimisation-based algorithms in UAVs include mixed-integer quadratic programming (MIQP) and mixed-integer semi-definite programming (MISDP).

Swarm-optimisation-based methods have been used extensively for small UAV path planning, namely particle swarm optimisation (PSO), ant and bee colony, and memetic algorithms, as they are efficient and fast to execute in resolving multi-objective problems, though are sensitive to hyperparameter tuning.

Learning-based algorithms are suitable for multi-objective and highly complex problems and include genetic algorithm (GA), artificial neural networks applied to potential field methods, Markov decision process (MDP) solvers, and deep reinforcement learning (DRL). These algorithms normally require a training phase and full state observability before implementation onboard UAVs. However, recent use of MDPs and partially observable Markov decision processes (POMDPs) to solve navigation problems for small UAVs improved the issue of motion planning under partial state observability.

Overall, the methods to plan paths for autonomous small UAVs are diverse, and selecting the best algorithm for exploration and object finding might be constrained to the needs and situational awareness of the surveying application, and available capabilities of UAVs used. It is swarm-optimisation and learning-based methods that are the best choice based on their capacity to solve multi-objective, and high complexity problems [56]. Formulating problems which account for partial observability and mitigation needs of object detection uncertainty in time-critical contexts, are required to develop the autonomous navigation of small UAVs in real-world environments. The following section presents

an analysis of these methods for autonomous UAV navigation in uncertain environments.

2.2 Autonomous Navigation for Small UAVs in Uncertain Environments

The implementation of online navigation methods in uncertain environments is essential for fully-autonomous small UAV operations [55]. Small UAV navigation can be seen as a dynamic multi-objective optimisation problem, so algorithms based on GAs, PSO, bayesian networks (BNs), fuzzy logic (FL), partially observable Markov decision processes (POMDPs) and DRL have been widely used in robotics research. Nevertheless, the type and amount of data required to solve a navigation problem for small UAVs have motivated recent research efforts to use learning-based rather than optimisation-based methods, which tend to require complete data (or full observability) of surveyed environments to avoid converging local optima [19], [56]. However, assuming full observability of the environment is unrealistic in real-world, time-critical, and unexplored environments.

Learning-based methods for autonomous navigation using small UAVs present similar issues. Deep learning models, for example, are excellent where there is a rich amount and variety of data sets and well-defined tasks [10], [48], [56]–[58]. The potential of DRL under these conditions has been demonstrated by case studies such as Alpha-Go [59] and autonomous small UAV navigation in simulation and controlled indoor environments [16], [60]. In contrast, DRL demands resource-rich simulation environments to train DQNs and subsequently output a motion policy that is executed by the UAV mid-flight [61]. This training process occurs *offline*, or prior to any real flight tests, and the high training and execution times are not always well documented. As model tuning using DRL is unaffordable using real hardware, the reliance of simulated environments restrict validation procedures of DRL-based methods for their use in real-world small UAV operations [56], [62], [63].

Model-based MDP solvers and their extensions are another type of learning-based algorithms, where the *learning* strategy is on the search of a motion policy by formulating the navigation problem as a sequential decision process (SDP), the mathematical framework of MDPs, and explicit modelling of the environment and dynamic variables of concern. The advantage of using MDPs and POMDPs over DRL-based methods is that they do not always require feature-rich simulated environments to calculate offline motion policies as these calculations now depend on the explicit model declaration of the environment, and the availability of *online* policy calculations, which could improve a UAVs traversed paths if it collects more observations of the environment mid-flight. An advanced understanding and the capability of mathematically modelling the dynamics of the environment is required in model-based learning methods.

Previous research demonstrated the possibilities of using model-based methods with online POMDP solvers for autonomous path planning, obstacle avoidance, object detection and tracking in GNSS-denied environments [64]–[66]. The unresolved challenges continue to be on designing efficient online SDP solvers for problems that have an elevated number of actions and states [14], [53], supporting algorithm validation in real-world scenarios using small UAVs [67]. Research is still necessary to further address the computational burden of these algorithms in hardware-constrained hardware [68], and develop model-based online SDP solvers running onboard small UAVs in environments with high levels of uncertainty and partial observability [57], [58].

2.3 Sequential Decision Processes (SDPs)

Real-world flight operations using small UAVs are dynamic, complex, partially observable, and therefore, uncertain. The process of increasing cognitive capabilities in autonomous navigation for small UAVs under these environments can be catalogued as a decision-making problem so the UAV can find adequate sequences of actions that maximise the probabilities of reaching the flight goal of exploration, object finding and tracking, or inspection of specific region of interests (ROIs) such as detailed scanning of structures or confirmation of potential detected victims. UAV navigation problems are normally denominated multi-objective, as the selection of sequential actions are not only constrained by achieving the primary surveillance goal, but also from the indirect interaction rules with the environment of obstacle avoidance, collision prevention, maximum flying distance, maximum and minimum survey heights, and maximum flight times.

Based on the premise that environments are dynamic, the set of chosen actions should not only predict possible outcomes, but also evaluate changes in the environment. Therefore, there is a strong interaction between the small UAV and the environment, where each action taken by the aircraft at a given time t should be monitored using fed back data about the state of environment in the shape of *observations*. Observations collected by UAVs are mostly acquired from sensors used by the autopilot, and attached payloads such as visible and near infrared cameras, microphones, spectrometers, light detection and ranging (LiDAR), and range sensors. The aim of collecting observations is to deduce the actual state of the surveyed environment, the UAV itself, and the objects to be identified, located, or quantified. Every action taken results in a change of state. A technique that gives quantitative values to taken actions is called the reward which means reward values will ultimately depend on the current state and the expected resulting state after the UAV executes an action. Here, the small UAV should decide on the next taken action at time $t + 1$ that returns the highest expected reward, and increase the chances of accomplishing the goal. This optimisation problem basically aims to define a sequence of actions that maximise those rewards in the long run.

The research presented here in this thesis used a framework based on MDPs, which follows the problem description and optimisation needs of finding an optimal sequential set of actions for every declared system state that allow the UAV to achieve its navigation mission. The following section presents an overview about the mathematical formulation of MDPs, followed by an introduction to POMDPs (Section 2.4), a generalisation of MDPs that introduces partial observability in its defined states.

2.3.1 Markov Decision Processes (MDPs)

An MDP is a mathematical framework to formulate sequential decision-making problems under uncertainty [53], assuming the state of the environment is *fully* observable. In the context of UAVs, for example, fully observability implies that the small UAV, formally called the *agent* in MDP theory, receives complete and accurate data about the state of the environment through its sensors or vision-based outputs such as object detection, semantic segmentation, or simultaneous localisation and mapping (SLAM). The aim of an MDP is to find a set of sequential actions needed to reach a goal, characterised by the current state of the environment. The state of the environment S is encoded with an array of variables that are relevant to the problem. Variables that can encode S in UAV navigation problems include the position of the UAV, position of the object to be found, and flags that indicate whether the UAV has collided with an obstacle, and if the object has been detected.

Actions A encoded in an MDP are limited to the operational abilities of the agent. Variables en-

coding low-level actions include position or velocity commands to move the UAV forward, backward, left, right, or change its direction clockwise or counterclockwise. Every time the agent takes an action $a \in A$ a transition to a new state $s \in S$ is triggered. After this transition, the agent receives a reward $r = R(s, a)$ from the environment in an interaction depicted in Figure 2.2.

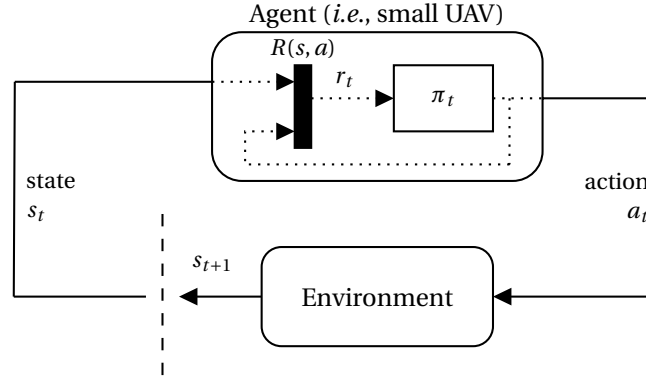


Figure 2.2: Chain diagram of actions, states, and rewards between the agent and the environment in MDPs. An action $a_t \in A$ taken by the agent triggers a transition to a new environment state $s_{t+1} \in S$. The agent receives a reward $r_t = R(s, a)$ from the environment.

An MDP is a stochastic process that follows the *Markov property*, where future state sequences will only depend on current states and are conditional independent of past state sequences, as defined in Equation (2.1):

$$\mathbb{P}(s_{t+1} | s_1, s_2, \dots, s_{t-1}, s_t) = \mathbb{P}(s_{t+1} | s_t). \quad (2.1)$$

MDPs are formally defined by the tuple $\langle S, A, T, R, \gamma \rangle$ where:

- S is a finite set of states.
- A is a finite set of actions.
- T is the state transition function, $T(s, a, s') = \mathbb{P}(s' | s, a)$.
- R is the reward function $R(s, a) = \mathbb{E}(r | s, a)$.
- γ is the discount factor $\gamma \in [0, 1]$.

The *discount factor* γ weights the significance of immediate rewards over long-term rewards. Small values of γ increase the importance of immediate rewards whereas high values increase the importance of long-term rewards. The solution of an MDP is represented with a policy π , which maps states to actions that allow the agent accomplish its goal [54]. Some examples of UAV navigation goals include, but are not limited to, object finding, exploration and mapping, and collision prevention during safe landing.

$$\pi : S \rightarrow A. \quad (2.2)$$

The optimal solution of an MDP is to find the policy that maximises the expected *return* G (or cumulative reward). G is defined using Equation (2.3):

$$G_t = r_{t+1} + \gamma r_{t+2} + \dots = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1}. \quad (2.3)$$

The goal of the discount factor γ is to keep the return bounded, as Equation (2.3) contains a sum with an infinite number of terms. Owing to uncertainty in the environment, the expected return G given a state $s \in S$ is calculated using a value function $v(s)$, as shown in Equation (2.4):

$$v(s) = \mathbb{E} (G_t | s = s_t). \quad (2.4)$$

From Equation (2.3), $v(s)$ can be expanded as follows:

$$\begin{aligned} v(s) &= \mathbb{E} (r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots | s) \\ v(s) &= \mathbb{E} (r_{t+1} + \gamma (r_{t+2} + \gamma r_{t+3} + \dots) | s) \\ v(s) &= \mathbb{E} (r_{t+1} + \gamma G_{t+1} | s) \\ v(s) &= \mathbb{E} (r_{t+1} + \gamma v(s_{t+1}) | s), \end{aligned} \quad (2.5)$$

where Equation (2.5) also known as the Bellman Equation, is a derived recurrence relation of $v(s)$ as the expected immediate reward r_{t+1} , plus the discounted value of the successor state $\gamma v(s_{t+1})$. The optimal value function $v^*(s)$, or the value function of the optimal policy, satisfies another Bellman equation, as shown in Equation (2.6):

$$v^*(s) = \max_a \left(R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s' | s, a) v^*(s') \right). \quad (2.6)$$

The optimal policy is defined in Equation (2.7) provided that $v^*(s)$ converges after calculating Equation (2.6) recursively:

$$\pi^*(s) = \operatorname{argmax}_a \left(R(s, a) + \sum_{s' \in \mathcal{S}} T(s' | s, a) v^*(s') \right). \quad (2.7)$$

As discussed in Section 1.1.2, observations collected by small UAVs through their sensors about the state of the environment, objects of interest, or the UAV itself are imperfect or incomplete. This condition is also known as *partial observability*. An important limitation of MDPs in this context is that they assume full state observability. Dynamic environments add uncertainty that impacts the representation of system states from collected observations. MDP states can not always been inferred with high confidence [50], which makes decision-making problems with partial state observability better formulated mathematically under the framework of POMDPs.

2.4 Decision-making under Uncertainty and Partial Observability

As the randomness of only partially observable data generates challenges for UAV operators to plan optimal paths, uncertainty also comes from unexpected environmental factors, such as sudden wind gusts that impact flight operations in general, and indirectly influences the design of robust motion control systems and navigation algorithms [69]. The deployment of small UAVs in fields such as SAR and disaster monitoring needs to develop the cognitive power for decision-making under unforeseen mid-flight events [50]. POMDPs are one such approach.

2.4.1 Partially Observable Markov Decision Processes (POMDPs)

The information received about the state of the environment from sensors embedded in small UAVs is normally incomplete and noisy. Hidden or partially observable states include: 1) global position

coordinates of the UAV and obstacles in surveyed environments; 2) weather events (*e.g.*, wind gusts, fog, poor illumination); 3) relevant metrics to collect data about objects of interest (*e.g.*, identification, location, quantity, and dimensions); and, 4) UAV status (*e.g.*, integrity of frame, motors and payload, and battery power levels). This partial knowledge has inspired the mathematical formulation of POMDPs [70], which compared with MDPs, agents are provided with a set of observations O , instead of states. Algorithms that solve POMDPs for small UAVs retrieve a motion policy, which outputs action commands derived using environment observations. Figure 2.3 presents the agent-environment interaction formulated in decision-making problems using POMDPs.

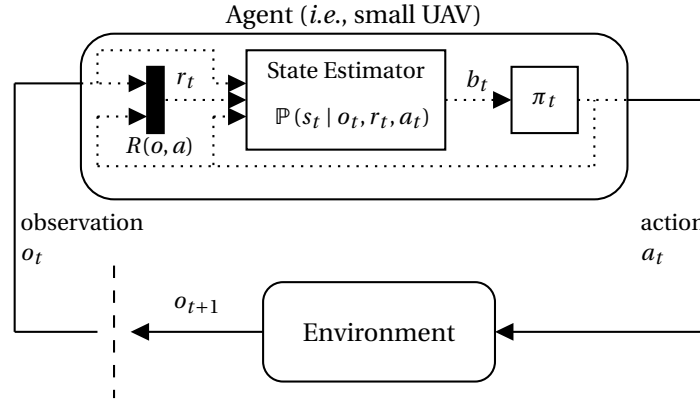


Figure 2.3: Chain diagram of actions, observations, and belief states between the agent and the environment in POMDPs. An action $a_t \in A$ taken by the agent triggers a transition to a new environment observation $o_{t+1} \in O$. As states cannot be directly inferred, a POMDP uses a state estimator to model future belief states b_t from collected observations.

POMDPs are formally defined by a tuple $\langle S, A, T, R, O, Z, b_0, \gamma \rangle$ [53] where:

- S is a finite set of states.
- A is a finite set of actions.
- T is a state transition function, $T(s, a, s') = \mathbb{P}(s' | s, a)$.
- R is a reward function, $R(o, a) = \mathbb{E}(r | o, a)$.
- O is a finite set of observations.
- Z is an observation transition function, $Z(s', a, o) = \mathbb{P}(o | s', a)$.
- b_0 is the initial belief state.
- γ is a discount factor $\gamma \in [0, 1]$.

In POMDPs, uncertainty given by partial observability of observed states is modelled using probability distributions over the system states. This modelling is known as the belief b , and is defined as follows:

$$b(H) = \mathbb{P}(s_1 | H), \dots, \mathbb{P}(s_n | H), \quad (2.8)$$

$$H = a_0, o_1, r_1, \dots, a_{t-1}, o_t, r_t, \quad (2.9)$$

where H is the history of actions, observations, and rewards the UAV has experienced until a time step t . A POMDP solver starts planning from an initial belief b_0 , whose distribution is modelled from known initial conditions (*i.e.*, situational awareness) about the environment and objects of interest (*e.g.*, victims in SAR applications). The belief distribution is then updated following a collected observation by the agent. In POMDPs, the policy π is the mapping from belief states to actions:

$$\pi : b \rightarrow A. \quad (2.10)$$

The value function for POMDPs incorporates the belief concept from Equation (2.6) as follows:

$$v(b) = \max_a \left(R(b, a) + \gamma \sum_{b' \in B} T(b, a, b') v(b') \right). \quad (2.11)$$

A POMDP is solved once the optimal policy π^* is found, defined as follows:

$$\pi^* := \arg \max_{\pi} \left(\mathbb{E} \left(\sum_{t=0}^{\infty} \gamma^t R(S_t, \pi(b_t)) \right) \right). \quad (2.12)$$

2.4.2 Autonomous Decision-making for Small UAVs using POMDPs

Theory on decision-making is extensive and relates to autonomous navigation in small UAVs, as well as fields such as game theory, Bayesian principles, and POMDPs. An advantage of formulating UAV navigation problems with POMDPs is the high level of flexibility to describe the problem and model uncertainty. Equally important is the use of POMDPs enables different distribution configurations of state-belief particles. These configurations meet the needs of real-world object finding applications, given they provide initial information, or situational awareness, about possible locations of objects to the localised, rather than assuming a search strategy from a generic set of initial conditions. In contrast, flexibility provided by POMDPs comes with the cost that finding exact solutions (*i.e.*, finding the optimal motion policy) of a navigation problem is computationally intractable [71], which means virtually every approach to solve problems formulated as a POMDP provides approximate optimal solutions to alleviate computational costs.

Autonomous navigation algorithms modelled under the framework of POMDPs and solved using model-based solvers have proven to be particularly useful for small UAVs [45], [72]–[75], and multi-objective decision-making problems [76]. For instance, POMDP-based frameworks for object detection and tracking under GNSS-denied and cluttered environments have been developed and validated in simulation and real tests in sub 2 kg UAVs [14], [77]. Nevertheless, the motion policies were computed using external workstations rather than using a companion computer onboard the UAV.

Research on POMDP-based solvers for dynamic path-planning applied to multiple object tracking [73], [78] has incorporated tasks such as path planning, collision avoidance, external wind disturbance effects and target tracking. These tests were conducted in simulation environments and provided limited evidence on the use of the framework in a real-world object tracking applications using small UAVs. Similarly, studies on POMDP frameworks have been assessed in humanitarian relief applications through simulation, arguing the need to validate existing methods in emulated disaster events using real flight tests [45].

Studies on small UAVs for autonomous navigation in environments under uncertainty using POMDPs have combined the use of model-based POMDP solvers including anytime meta planner (AMPLE) [79], POMDP-lite [80], decentralised POMDP [67], mixed observability Markov decision process (MOMDP) [81], POMCPOW and PFT-DPW [82]. Despite the development of novel solvers on formulated problems using POMDPs [83], [84], the majority of experimental results are constrained to simulated environments, limiting the understanding and applicability of these algorithms in real-world environments [74], [85]. Overall, a broader validation of POMDP solvers in real-world environments and time-critical applications is required to broaden the scope of small UAVs with enhanced onboard decision-making in complex surveillance scenarios.

POMDP solvers developed over the last decades categorised as *offline* and *online* solvers. Offline solvers compute an approximate solution before the agent interacts with the environment. A drawback of these offline solvers is the restriction of tracking the redistribution of belief particles during policy execution. By comparison, online solvers compute and update motion policies in (near) real-time while the agent chooses actions and collects observations. Provided that real-time motion planning capabilities are a requirement in autonomous navigation under uncertainty for small UAVs, the use of online POMDP solvers are suitable in the design and implementation of frameworks for small UAVs. The model-based online POMDP solver as well as framework implementations that best suit the autonomous decision-making needs in partially observable and uncertain environments for resource-constrained hardware onboard small UAVs is adaptive belief tree (ABT), which is discussed in the following section.

2.4.3 Adaptive Belief Tree (ABT)

Research on model-based online POMDP solvers have significantly progressed over the last years [59], [86]. Nonetheless, most POMDP solvers are unable to calculate approximated solutions (*i.e.*, policy) to formulated POMDP problems in real time, as they recompute the policy at each time step from scratch. This concept of re-planning is inefficient for real-time applications due to waste of computational resources after discarding computed policies from previous time steps. While that loss may not be relevant for autonomous decision-making in static and structured environments, real-world environments often present gradual or partial changes, which aggravate that loss of resources.

ABT [87] is an online POMDP solver that updates its policy at each planning step without incurring computational resource loss. Instead of recomputing the policy from scratch, ABT updates the previous policy when changes to the POMDP model are detected. Similar to partially observable Monte Carlo planning (POMCP), ABT computes optimal policies in continuous state spaces, and approximates the solution by maintaining a set of multiple sample episodes to reconstruct an *augmented belief tree* \mathcal{T} . Subsequently, the transition T and observation functions Z of the POMDP model are not explicitly defined but as a generative model. ABT is an extension of POMCPs as it also uses Monte Carlo simulations to represent belief states with a set of particles and predict future belief states. The root node of the search tree is defined by the initial belief b_0 , followed by tree branches and nodes (*i.e.*, belief nodes) representing the probability of taking an action $a \in A$ and receiving an observation $o \in O$, as presented in Figure 2.4.

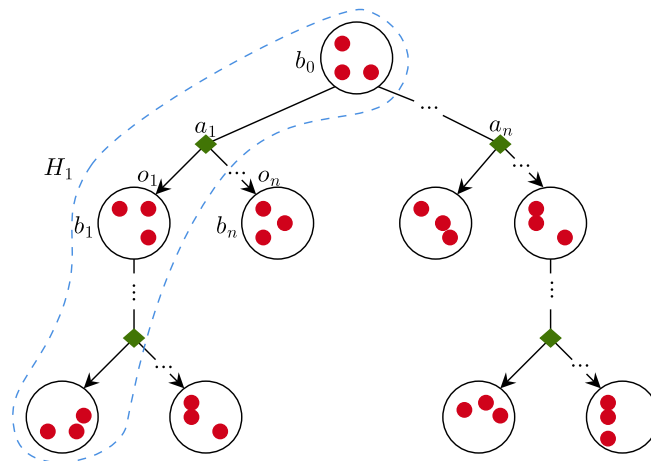


Figure 2.4: Belief tree using POMDPs, adapted from [88]. Belief nodes (illustrated with circles) are probability distributions over system states (red dots). Belief distributions are updated after new observations are collected.

The ABT solver contains two key components during its execution, *preprocessing* and *runtime*, as presented in Algorithm 1:

Algorithm 1 Augmented Belief Tree (ABT) solver, adapted from [89]. Here, \mathcal{T} is the augmented belief tree, H is a set of sampled episodes.

Require: Initial POMDP model m_0 , initial belief b_0 .

Preprocessing Component

- 1: $(\mathcal{T}, H) \leftarrow \text{GeneratePolicy}(m_0, b_0)$
 - 2: Let S' be the set of all sampled states in H , i.e., $S' \leftarrow \{h_i.s \mid i \in [0, |h|], h \in H\}$
 - 3: Let \mathcal{R} be a spatial index (e.g., a range tree) representing S' .
 - 4: $b \leftarrow b_0$.
-

Runtime Component

- 5: **while** running **do**
 - 6: **if** $m_t \neq m_{t-1}$ { m_i is the POMDP model at time i } **then**
 - 7: $H' \leftarrow \text{IdentifyAffectedEpisodes}(P_{t-1}, P_t, H, \mathcal{R}, \mathcal{T})$.
 - 8: $\text{ReviseEpisodes}(P_t, \mathcal{T}, b, H')$.
 - 9: $\text{UpdateValues}(\mathcal{T}, b, H')$.
 - 10: **end if**
 - 11: **while** there is still time **do**
 - 12: $\text{ImprovePolicy}(P_t, H, \mathcal{R}, \mathcal{T}, b)$.
 - 13: $a \leftarrow \text{Get the best action in } \mathcal{T} \text{ from } b$.
 - 14: Perform action a .
 - 15: $o \leftarrow \text{Get observation}$.
 - 16: $b \leftarrow \tau(b, a, o)$.
 - 17: $t \leftarrow t + 1$.
 - 18: **end while**
 - 19: **end while**
-

The preprocessing component enables ABT to search for a policy offline using the initial belief b_0 and POMDP model m_0 . The policy is constructed after sampling a finite set of *episodes*. An episode $h \in H$ is a sampled state trajectory located in a belief node from \mathcal{T} . ABT samples an episode by selecting an initial state $s_0 \in S$ contained in b_0 and choosing an action $a_0 \in A$ using the upper confidence bound algorithm. From a_0 and s_0 , the solver calls its generative model to sample: 1) an observation $o_0 \in O$; 2) the expected immediate reward $r_0 = R(s, a)$; and, 3) a next state $s_1 \in S$. The quadruple (s_0, a_0, o_0, r_0) is then stored as the first element of h , and the process repeats itself to collect the next quadruple starting from $s_1 \in S$ until a terminal state is reached, or after exceeding a number of stored quadruples to sample h . A set of sampled episodes ultimately constitutes H .

The paths in \mathcal{T} are associated with sampled episodes in H . Assuming b as a belief node at level l from \mathcal{T} , the policy π is defined as:

$$\pi(b) = \operatorname{argmax}_{a \in A(E, b)} \hat{Q}(b, a), \quad (2.13)$$

where $\hat{Q}(b, a)$ is the estimated Q-value; and, $A(E, b)$ is the set of actions used to expand b . The solver estimates $\hat{Q}(b, a)$ as:

$$\hat{Q}(b, a) = \frac{1}{|H(b, a)|} \sum_{h \in H(b, a)} V(h, l), \quad (2.14)$$

$$V(h, l) = \sum_{i=l}^{|h|} \gamma^{i-l} R(h_i \cdot s, h_i \cdot a), \quad (2.15)$$

where $H(b, a)$ is the set of sampled episodes containing the sequence (b, a) starting from b_0 ; $V(h, l)$ is the value of an episode h from level l ; γ is the discount factor; and, R is the reward function. As the number of sampled episodes in $H(b, a)$ increases, $V(h, l + 1)$ converges to the true Q-value in probability, allowing ABT for converging to the optimal policy in probability [87].

During execution in runtime, the ABT solver identifies parts to update from the policy by evaluating the episodes in H that contain states affected by changes in the POMDP model m . ABT updates the policy following three key steps by: 1) disconnecting the association between affected episodes and corresponding belief nodes in \mathcal{T} ; 2) revising h based on the new POMDP model and linking it back to \mathcal{T} ; and, 3) updating Q-values of newly associated and disassociated beliefs with h .

2.5 Small UAV Frameworks for Decision-making in Time-critical Applications

Research studies on autonomous decision-making in time-critical applications for small UAVs using POMDP formulations are scarce. That conducted by Chanel *et al.* [90] proved an application of multi-object car recognition using a customised optimisation framework, which executed the POMDP solver onboard the small UAV and was optimised during execution. However, limited experimentation details restrict the reproducibility of this study in terms of the UAV frame, hardware specifications, onboard computer, and vision-based object detector model. Since real tests of this UAV framework were conducted in an open rural field, experiments were simplified as detected cars appeared isolated from nearby obstacles. During an emergency, object (or victim) visuals from camera frames are likely to be challenging because of factors such as partial or full occlusion, viewpoint, and scale.

Other significant research conducted by Vanegas-Alvarez [65] investigated the capabilities of on-line model-based POMDP solvers and UAV frameworks for autonomous object detection and tracking under cluttered and GNSS-denied environments. Evaluation of the two fastest online solvers, partially observable Monte Carlo planning (POMCP) [86] and ABT established the problem formulation and framework with ABT. Experimental validation of their implementation was narrowed for navigation in indoor environments, and object detection used trivial markers. Another limitation is that computing power to run the ABT solver was allocated to an external workstation rather than to the UAV itself, so action commands returned by the solver were transmitted onboard to the autopilot of the aircraft via its communication sub-system. Further research is therefore needed as Valavanis and Vachtsevanos [12] and Carrio *et al.* [48] claim the dependency of communication modules for these tasks is undesirable because the performance of small UAVs might be seriously compromised if those modules fail.

UAV framework implementations on computer vision methods for object detection and decision-making in real time are diverse. These methods can be catalogued into two primary categories: 1) feature extraction; and 2) machine learning [91]. Algorithms based on feature extraction synthesise the contents of input images by extracting key data features using explicit mathematical operations. In the context of small UAVs for real-time object detection, classical image processing algorithms have been established to detect wildlife using thermal imagery [3], augmented-reality markers [34],

[65], clothes from stranded victims [92], weeds [93], platforms for take-off and landing [42], and vehicles [90]. In order to establish an object detector method using feature extraction, a great research effort is normally required to iterate over these operators until extracting relevant features such as edges, corners, specific colours, shape and texture detection, and temporal features.

Machine learning, and specifically, methods based on deep learning are state-of-the-art techniques which require a training set of labelled images and objects in order to tune an artificial neural network able of extracting key features. These networks are known as convolutional neural networks (CNNs) and use convolutional operators to extract such features, and identify and classify objects. An increasing number of UAV applications with onboard inference of CNN models [27], [94] is now possible because of hardware developments such as NVIDIA graphical processor units (GPUs) and Intel vision processing units (VPUs) [51], [95], [96]. The use of the robot operating system (ROS) is extensive in robotic applications including UAVs, and has proven to be a successful middleware for modular software systems [64], [97]. Technologies closely linked with ROS are Gazebo and Airsim as robotic simulators, micro air vehicle link (MAVLink) as the industry standard for UAV communications, Pixhawk flight controller unit (FCU) boards, and PX4 autopilot [62], [98], [99]. Since most framework designs are constrained for single application needs and have not been tested in a diverse range of UAV survey needs [18], [98], researchers or UAV developers may continue to reuse or adopt existing approaches to their own customised UAV solutions, increasing costs of research and development.

Overall, this research aims to expand the use of ABT as it is one of the solvers tested with the most framework implementations, with a higher focus on real flight experimentation onboard small UAVs, and in challenging indoor and outdoor environments. The framework design should run computationally expensive operations onboard a small UAV, such as approximating a POMDP solution via ABT, and onboard inference of cutting-edge object detector methods based on CNNs. The design of the UAV framework should demonstrate robustness and scalability properties against a range of camera payloads, object detectors, and UAV remote sensing fields.

2.6 Summary

This research serves to establish the limits of SDP algorithms such as model-based online POMDP solvers in UAV applications of exploration and object finding, with higher levels of environment and object uncertainty, namely humanitarian relief where the structural conditions of affected zones are unknown, and SAR where the quick identification, localisation, and quantification of victims is critical. Autonomous onboard decision-making that mitigates object detection uncertainty from computer vision algorithms (*i.e.*, identified, located, quantified or measured objects) in complex scenarios will be undertaken in this research using modelling approaches to reduce uncertainty from outputs of modern vision-based detectors such as CNN object detectors. Better data representations that improves data acquisition quality and accuracy in detected objects (or victims) is essential for small UAVs to be fully practical in time-critical applications [22]. A system that offers better situational awareness by evaluating data quality in real-time onboard small UAVs will allow first responders to intervene more effectively. Table 2.3 presents the research analysis undertaken to establish the two key research questions.

Table 2.3: Literature review summary of autonomous decision-making methods for autonomous navigation under environment and object detection uncertainty onboard small UAVs.

Research Topic	Research Gap	Research Question	Contribution
SDPs for autonomous motion planning in complex environments under uncertainty and partial observability using small UAVs.	Need to validate SDPs onboard small UAVs in complex setups and real flight tests.	1.1 What level of artificial cognitive learning and uncertainty modelling is required to identify, localise or quantify objects positioned in cluttered and challenging scenarios using small UAVs and vision-based object detectors?	Problem formulation motion policy computed via a model-based POMDP solver, designed for indoor (Chapter 3) and outdoor environments (Chapter 5).
Addressing identification (<i>i.e.</i> , confidence) uncertainty from vision-based object detectors in surveys for real-time processing onboard small UAVs.	Need to prioritise robust intelligent small UAVs to reduce object detection uncertainty rather than retraining existing CNN models.	1.2 Which factors define the complexity of an SDP so as to reduce object detection uncertainty from collected environment observations using vision-based sensors in time-critical applications?	POMDP-based motion planner onboard small UAVs that interacts in complex environments from off-the-shelf CNN object detectors (Chapter 4).
	Need to examine modelling methods of object detection uncertainty and their application in SDPs for UAV motion planning.		Extended problem formulation with detection confidence uncertainty, reward function, and flight modes tested with several object position distributions (Chapters 4 and 5).
Framework implementations of SDPs and vision-based methods for autonomous navigation in resource-constrained hardware onboard small UAVs.	Need to optimise onboard inference of computationally expensive CNN models for sub-2 kg UAV hardware.	2.1 How can computationally intensive tasks such as online SDP algorithms, and onboard inference of CNN models for object detection and segmentation, be integrated to run simultaneously under resource-constrained hardware in sub-2 kg UAVs?	Framework design for deployment of CNN models through VPUs and OpenVINO (Chapters 4, 6 and 7).
	Need to integrate computationally expensive motion planners such as model-based online POMDP solvers onboard sub-2 kg UAV hardware.		Framework design that computes a motion policy using a model-based POMDP solver onboard a sub-2 kg UAV (Chapters 4, 6 and 7).
Robustness and scalability of frameworks for autonomous decision-making onboard small UAVs in various application domains.	Need to design a modular and scalable UAV framework implementations beyond a single field and application domain.	1.3 What are the modelling considerations in a formulated SDP that enable the scalability of a UAV framework for autonomous navigation for a range of diverse vision-based payloads, and remote sensing application needs beyond object detection?	Modular and scalable UAV framework validated with SAR and planetary exploration case studies (Chapters 6 and 7). Scalability of vision-based payloads achieved via inclusion of camera lens properties in the problem formulation (Chapters 5 and 7). Robustness of framework tested with various CNN models for RGB and thermal object detection and image segmentation (Chapters 6 and 7).
		2.2 What are the design criteria to scale a UAV framework for autonomous navigation under environment and object detection uncertainty to other remote sensing applications that require autonomous decision-making capabilities onboard small UAVs?	

Chapter 3

Autonomous UAV Navigation for Active Perception of Targets in Uncertain and Cluttered Environments

SMALL UAVs are used increasingly in civilian applications that demand rapid human intervention such as surveillance, disaster monitoring and SAR as they become more affordable, offer real-time telemetry in inaccessible or dangerous zones, greater payload flexibility, and an increasing number of autonomous capabilities. This chapter presents the first published paper from this research and establishes the problem formulation for UAV autonomous navigation in cluttered environments under object detection uncertainty and partial observability. The problem is mathematically formulated using a POMDP, and solved online using the ABT solver. The system is designed to run in resource-constrained hardware onboard small UAVs, and tested using the Gazebo robotics simulator, ROS, and PX4 SIL autopilot plugin. This paper presents an HIL implementation for onboard inference of a pre-trained CNN-based model to identify and locate victims from streamed camera frames by a virtual camera, and processed in real-time using an Intel Myriad vision processing unit (VPU). The paper discusses the mathematical formulation of the navigation problem, and the robustness of the POMDP-based motion planner at various levels of victim location and pose uncertainty.

Statement of Contribution of Co-Authors for Thesis by Published Paper

The authors listed below have certified that:

1. they meet the criteria for authorship and that they have participated in the conception, execution, or interpretation, of at least that part of the publication in their field of expertise;
2. they take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;
3. there are no other authors of the publication according to these criteria;
4. potential conflicts of interest have been disclosed to (a) granting bodies, (b) the editor or publisher of journals or other publications, and (c) the head of the responsible academic unit, and
5. they agree to the use of the publication in the student's thesis and its publication on the [QUT's ePrints site](#) consistent with any limitations set by publisher requirements.

In the case of this chapter:

J. Sandino, F. Vanegas, F. Gonzalez, and F. Maire, "Autonomous UAV navigation for active perception of targets in uncertain and cluttered environments," in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2020, pp. 1–12. DOI: [10.1109/AERO47225.2020.9172808](https://doi.org/10.1109/AERO47225.2020.9172808).

Contributor	Statement of Contribution ¹
Juan Sandino QUT Verified Signature Date: <u>2022-05-20</u>	Contributed on: conceptualisation, methodology, software, validation, formal analysis, investigation, data curation, writing-original draft preparation, and visualisation.
Fernando Vanegas	Contributed on: conceptualisation, methodology, software, investigation, and writing-review and editing.
Felipe Gonzalez	Contributed on: conceptualisation, methodology, resources, writing-review and editing, supervision, project administration, and funding acquisition.
Frederic Maire	Contributed on: conceptualisation, writing-review and editing, and supervision.

Principal Supervisor Confirmation		
I have sighted email or other correspondence from all co-authors confirming their certifying authorship.		
<i>Felipe Gonzalez</i>	QUT Verified Signature	<u>2022-05-20</u>
Name	Signature	Date

¹ Contributions follow the CRediT taxonomy. Further details can be found at <https://credit.niso.org/>.

Autonomous UAV Navigation for Active Perception of Targets in Uncertain and Cluttered Environments

Juan Sandino

School of Electrical Engineering and Robotics
Queensland University of Technology
2 George St., Brisbane City, QLD 4000, Australia
j.sandino@qut.edu.au

Felipe Gonzalez

School of Electrical Engineering and Robotics
Queensland University of Technology
2 George St., Brisbane City, QLD 4000, Australia
felipe.gonzalez@qut.edu.au

Fernando Vanegas

School of Electrical Engineering and Robotics
Queensland University of Technology
2 George St., Brisbane City, QLD 4000, Australia
f.vanegasalvarez@qut.edu.au

Frederic Maire

School of Electrical Engineering and Robotics
Queensland University of Technology
2 George St., Brisbane City, QLD 4000, Australia
f.maire@qut.edu.au

Abstract—The use of Small Unmanned Aerial Vehicles (sUAVs) has grown exponentially owing to an increasing number of autonomous capabilities. Automated functions include the return to home at critical energy levels, collision avoidance, take-off and landing, and target tracking. However, sUAVs applications in real-world and time-critical scenarios, such as Search and Rescue (SAR) is still limited. In SAR applications, the overarching aim of autonomous sUAV navigation is the quick localisation, identification and quantification of victims to prioritise emergency response in affected zones. Traditionally, sUAV pilots are exposed to prolonged use of visual systems to interact with the environment, which causes fatigue and sensory overloads. Nevertheless, the search for victims onboard a sUAV is challenging because of noise in the data, low image resolution, illumination conditions, and partial (or full) occlusion between the victims and surrounding structures. This paper presents an autonomous Sequential Decision Process (SDP) for sUAV navigation that incorporates target detection uncertainty from vision-based cameras. The SDP is modelled as a Partially Observable Markov Decision Process (POMDP) and solved online using the Adaptive Belief Tree (ABT) algorithm. In particular, a detailed model of target detection uncertainty from deep learning-based models is shown. The presented formulation is tested under Software in the Loop (SITL) through Gazebo, Robot Operating System (ROS), and PX4 firmware. A Hardware in the Loop (HITL) implementation is also presented using an Intel Myriad Vision Processing Unit (VPU) device and ROS. Tests are conducted in a simulated SAR GPS-denied scenario, aimed to find a person at different levels of location and pose uncertainty.

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. RELATED WORK	2
3. BACKGROUND	3
4. PROBLEM FORMULATION.....	3
5. IMPLEMENTATION.....	6
6. RESULTS	8
7. CONCLUSIONS.....	9
APPENDIX	10
ACKNOWLEDGMENTS	10
REFERENCES	10
BIOGRAPHY	12

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are currently used in significant civilian applications such as remote sensing, disaster monitoring, surveillance, and Search and Rescue (SAR) [1]. Part of the success of this technology is caused by cheaper hardware and advances in sensor systems, computer vision and image processing, and autonomous navigation. Advances in autonomous navigation include automated home return at critical battery levels, reactive collision avoidance, autonomous take-off and landing, and active target tracking. Nonetheless, the deployment of these systems at a broader scale is still restricted by operational limitations in hardware and software [2]. Current limitations on Small UAVs (sUAVs) are drones that weigh less than 13 kg [3] include: (a) on-board computing power, (b) payload weight, (c) energy storage, (d) sensor resolution and image quality and (e) cognition capabilities in unstructured environments [4].

While recent research suggests that some hardware constraints are likely to be resolved soon [5], the development of autonomous decision-making processes on sUAVs is a far more challenging problem [4]. Decision-making capabilities on sUAVs are still limited when dealing with mid-flight events, path planning and obstacle avoidance, or target finding under uncertainty and partial observability [6]. Engineers and roboticists usually reduce uncertainty by adjusting robot working environments to become as structured as possible [7]. However, real-world applications are unstructured, full of uncertainties.

Natural disasters are events that unfortunately still claim human lives around the globe [8, 9]. Emergency situations also occur by unfortunate situations such as people getting into distress in rivers and shores and lost people in rural areas. Disaster management and rescue workers are critical to react in emergency situations and diminish fatalities. In SAR applications, the overarching aim of autonomous sUAV navigation is the quick localisation, identification and quantification of victims to prioritise emergency response in affected zones [10]. Various challenges need to be addressed to enable a sUAV to navigate autonomously in SAR scenarios.

In emergency situations, the available information for the evaluation of access areas, affected structures (if any), and the identification of victims (if any) is usually unknown or limited [11]. Second, the vision tasks performed by a sUAV are challenging because of noise in the data, low image resolution, illumination conditions, and partial (or full) occlusion of the victims. Additionally, the application of sUAVs under

cluttered and GPS-denied environments require the use of Simultaneous Localisation and Mapping (SLAM) algorithms that rely on advanced sensor systems (e.g. LiDAR), which are often expensive, complex to operate and sometimes, computationally intensive. Moreover, optimal control of sUAVs require the use of workstations owing to their resource-constrained onboard hardware. Normally, sUAV operators decide on the next sequence of navigation commands to interact with the environment using visual telemetry [12]. However, the prolonged use of these interactive systems is claimed to produce fatigue and sensory overloads. Last, the sUAV must avoid collisions within the surveyed areas, it should generate a path-planning strategy for exploration based on the environment, natural disturbances (e.g. wind, atmospheric pressure, temperature) and the drone kinematic constraints.

Despite recent advances in autonomous sUAVs for SAR applications [13–15], there are still unresolved issues. Namely, lack of validation of these systems in real-world environments, uncertainty in the identification, location and counting of objects of interest (i.e. victims) under complex image representations, and optimal interaction between the drones and the victims. The above-mentioned limitations motivate the exploration of sUAV systems with increased cognition and autonomy level in stochastic environments under target detection uncertainty from vision-based sensors. This paper presents a framework for autonomous sUAV navigation that models target location and identification uncertainties from vision-based cameras and object detection models. The problem is formulated as a Partially Observable Markov Decision Process (POMDP) and solved online using the Adaptive Belief Tree (ABT) algorithm. In particular, a detailed model of the uncertainty derived from a deep learning object detector algorithm is demonstrated. The presented system handles autonomous sUAV navigation under uncertainty as a multi-objective problem comprising path planning, obstacle avoidance, motion control, and target detection tasks. The proposed approach addresses the following challenges:

1. Autonomous sUAV decision-making under environment uncertainty in SAR applications.
2. Uncertainty on the identification and location of victims in cluttered and GPS-denied environments.
3. Modelling of target detection uncertainty from vision-based sensors.
4. Execution of computationally expensive decision-making and object detection algorithms in resource-constrained hardware.

The proposed system is validated with experiments in simulation. In these experiments, the sUAV is tasked with the mission of finding a lost person in a cluttered environment.

The rest of paper is structured as follows: Section 2 discusses previous works on autonomous sUAV decision-making under uncertainty and target detection uncertainty. Section 3 reviews briefly uncertainty representation, POMDPs and online ABT solvers for autonomous sUAV navigation. Section 4 details the problem formulation and modelling of target detection uncertainty. Section 5 describes the proposed system architecture, software and hardware tools, and experimentation setup. The evaluation of the suggested formulation under a simulated cluttered and GPS-denied scenario is shown in Section 6. Finally, Section 7 discusses future work.

2. RELATED WORK

Theory on decision-making is extensive and relates not only to autonomous sUAV navigation but also to other fields such as multi-objective decision-making, game theory, navigation strategies, Bayesian principles, Markov Decision Processes (MDPs) and Partially Observable Markov Decision Processes (POMDPs) [16]. Since time-critical applications (e.g. SAR) feature uncertainty and partial observability in the system states and targets, MDPs and specially POMDPs have proven to be useful while making navigation decisions under these conditions [10, 17–20]. Literature has likewise shown how the modelling of POMDPs in highly uncertain environments and partial observability is a suitable approach for sUAV navigation problems [16]. Vanegas and Gonzalez [21], for example, implemented an autonomous sUAV navigation algorithm for GPS-denied and cluttered environments. The authors compared two of the fastest POMDP online solvers, namely Partially Observable Monte Carlo Planning (POMCP) [22] and Adaptive Belief Tree (ABT) [23]. The proposed framework detailed the possibility of using the ABT solver for the drone to make decisions in seconds. Nonetheless, the authors narrowed their tests for indoor environments and detected trivial targets, particularly 2D markers. Additionally, the navigation commands were transmitted using the sUAV communication module to a workstation. As discussed by Valavanis and Vachtsevanos [24] and Carrio et al. [4], the dependency of communication modules for these tasks is undesirable because the drone's behaviour under complex environments might become seriously compromised if those modules fail.

Another significant research by Ragi and Chong [18, 25] presents POMDP-based solvers for dynamic path-planning applied to multiple target tracking. The POMDP formulation became more significant towards fully autonomous sUAVs by including path planning, collision avoidance, external wind disturbance effects, and tracking evasive treats. Similarly, Bravo et al. [10] assessed POMDP frameworks in humanitarian relief applications through simulation. The authors concluded a higher need to validate existing methods in real disaster situations. Other related POMDP-based solvers have also shown advances compared to standard models such as Anytime Meta PLannEr (AMPLE) [26], POMDP-lite [27], decentralised POMDP [28] and Mixed Observability Markov Decision Process (MOMDP) [29].

Research studies on onboard autonomous sUAV decision-making for real-time applications using POMDPs are limited. One of the most notable studies is the work conducted by Chanel et al. [30], who could show a multi-target car recognition application using a customised optimisation framework. Their system was able to run the POMDP solver onboard the sUAV and optimised during execution. The authors, nevertheless, did not provide relevant experimentation details such as the sUAV model, hardware specifications and vision-based algorithms. Furthermore, they also demonstrated the framework in an open rural field, where levels of target uncertainty could become considerably low once a car is perceived under the camera's FOV and detected by vision-based detectors. Compared to emergency environments, image representations of victims are more challenging due to partial or full occlusion, pose, and dynamics.

3. BACKGROUND

Types of Uncertainty

sUAV perception is limited by noisy sUAV onboard sensors, poor image representations of the targets caused by partial and entire occlusion from other objects, and the dynamics of the target itself [7]. Partial observability from vision-based cameras introduces target uncertainty, which can be classified as follows:

1. *Location*: target location uncertainty is caused by the limited extent of vision-based sensors to capture images (also known as Field of View (FOV)). These limitations are present in mapping, surveillance, and any other target finding applications under cluttered and challenging environments. The location of victims in SAR operations is, for example, challenging when vision-based sensors cannot cover specific FOV configurations (e.g. sUAV viewpoint in non-nadir or side-by-side settings) or when the victims are partially occluded by an urban structure, the effects of the disaster situation (e.g. fire smokes, earthquakes, floods) or natural events (e.g. fog, cloudy and wet conditions).
2. *Identification*: this refers to limitations on recognising a specific identification property of the object of interest. Identification uncertainty on victims includes age, gender and health conditions.
3. *Quantity*: target counting uncertainty applies where multiple victims are partially occluded among themselves, between themselves and environment objects, or environmental factors that affect the image quality of vision-based sensors. Counting is important in humanitarian relief operations when rescuers are required to prioritise emergency response by estimating the number of victims. The collection of these statistics will determine, therefore, the areas that require immediate intervention.
4. *Dimensions*: this type of uncertainty occurs when the surveying application requires the collection of morphological properties of the victims. Examples include a person's height and volume. In SAR operations, the awareness of a victim's height or volume could infer complementary rescue conditions and define resources and processes needed to assist them.

Another source of uncertainty comes from the outputs of computer vision algorithms for object detection. Developed algorithms use varied strategies for object detection, ranging from classical image processing manipulations (e.g. filtering, thresholding, edge and contour detection, and morphological operations), feature extraction (e.g. SIFT, SURF), adaptive filtering (e.g. Kalman filters), visual odometry (e.g. SLAM) and deep learning (e.g. Region-based and Single Shot Detector CNNs) [31]. SDPs for autonomous sUAV decision-making should consider uncertainty from vision-based target detectors to adjust their path planning when detections lack accuracy.

Uncertainty in sUAV navigation is caused by the kinematics of the drone and disturbances from the environment. Since a sUAV takes off, there is always a present drift in the initial position and heading. Errors in a sUAV local position estimator likewise exist from propagated errors from sensor readings and motion controllers. Last, the dynamics in the environment might affect the behaviour of sUAVs, such as changes in wind direction and currents, temperature and atmospheric pressure.

4. PROBLEM FORMULATION

Partially Observable Markov Decision Process

The process of adding cognition capabilities to a sUAV in uncertain environments can be catalogued as a decision-making problem. An autonomous sUAV system should decide the optimal sequence of actions that maximises the probabilities of accomplishing the flight mission. The flight mission consists in searching for a victim and stops once the first victim is found. This selection of sequence of actions are evaluated not only by achieving the primary search goal, but also by following some interaction rules with the environment, namely collision avoidance, the exploration within a pre-defined Region of Interest (ROI) and short flight times. Considering that real-world environments are dynamic, the sUAV should also evaluate changes in the environment and update its path planning strategy. Therefore, there must be a constant interaction between the sUAV and the environment, where each action taken by the drone at a time t should be monitored using collected data from the environment in the form of observations. This data analysis aims to infer the current conditions (or states) of the system and how close the sUAV is to achieve the primary mission goal. Based on that analysis, the sUAV should decide on the next taken action at a time $t + 1$ that increases the chances of accomplishing the mission. This optimisation problem aims to define a sequence of actions that maximises those rewards in the long run, which depend on the current state of the environment and individual collected rewards.

MDPs are discrete-time mathematical models that allow the description of Sequential Decision Processes (SDPs) on environments under uncertainty [7]. A problem formulation under MDPs assumes that the system states are *fully* observable. Conversely, POMDPs incorporate uncertainty and *partial* observability from the agent (i.e. sUAV) in the system states. POMDPs are defined by the tuple $\langle S, A, T, R, O, \mathcal{Z}, \gamma \rangle$ where S is a finite set of states, A is a finite set of actions, T is a state transition probability matrix $T_{ss'}^a = \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a)$, R is a reward probability matrix $\mathcal{R}_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$, O is a finite set of observations, \mathcal{Z} is an observation probability matrix $\mathcal{Z}_{s'o}^a = \mathbb{P}[O_{t+1} = o | S_{t+1} = s', A_t = a]$, and γ is a discount factor $\gamma \in [0, 1]$.

In POMDPs, uncertainty in the system states are represented by a probability distribution of the system over all possible states in its state space, called belief states b , defined in Equation 1:

$$b(h) = (\mathbb{P}[S_t = s^1 | H_t = h], \dots, \mathbb{P}[S_t = s^n | H_t = h]) \quad (1)$$

where H is the *history* of actions, observations and rewards that the agent has experienced until time t ,

$$H_t = a_0, o_1, r_1, \dots, a_{t-1}, o_t, r_t \quad (2)$$

Given a current belief state b , the goal of any POMDP solver is to find a sequence of actions that maximises the discounted accumulated reward. This sequence of actions is commonly known as the policy π . The behaviour of an agent is represented by mapping a policy $\pi : b \rightarrow A$. The POMDP is solved by finding the optimal policy π^* that maximises the expected accumulated reward.

$$\pi^* := \arg \max_{\pi} \left(\mathbb{E} \left[\sum_{\tau=0}^{\infty} \gamma^{t_{\tau}} R(S^{t_{\tau}}, \pi(b^{t_{\tau}})) | b^{t_0}, \pi \right] \right) \quad (3)$$

Adaptive Belief Tree

Many online POMDP solvers recompute the optimal policy at each time step from scratch. This concept or re-planning is inefficient for real-time applications because of the loss of computational and time resources by discarding computed policies at previous time steps. While that loss might not be as important on static, structured environments, real-world environments often present gradual or partial changes, which aggravate that loss of resources. The ABT solver, developed by Kurniawati and Yadav [23], proposes the reuse of the previous computed policy and generates policy updates when changes in the POMDP model are detected. Similar to Partially Observable Monte Carlo Planning (POMCP), ABT can approximate optimal policies in continuous state spaces. The ABT solver approximates the solution by maintaining a set of multiple sample episodes. Consequently, the probability distributions of the POMDP are not explicitly defined but as a generative model.

The ABT solver uses an approach of planning and execution in real time. First, an offline optimal policy is calculated based on the POMDP model. Then, the agent executes an action from the offline obtained policy. The agent collects an observation following the chosen action. Afterwards, the belief states are updated based on the collected observation. Subsequently, the ABT solver will update the policy from the updated belief states. Finally, the agent is ready to execute the next action from the updated policy.

sUAV Navigation Task

The proposed approach aims for a sUAV to find a lost person in a cluttered indoor environment. The environment, illustrated in Figure 1, contains a restricted flying area, several obstacles and the person to be found at different position configurations (further setup details can be found in Section 5 and Section 6). The sUAV is assumed to incorporate a visual odometry system for pose and motion estimation. Observations from vision-based sensors comprise a downward-looking camera. Functions such as take-off, landing, and return home are delegated to the sUAV autopilot. Therefore, the autonomous sUAV navigation task will begin after the drone reaches an initial waypoint location and will finish after the victim is found. The optimal policy to be learnt is the one that allows the sUAV to accomplish path planning, obstacle avoidance, and finding the victim successfully at different levels of uncertainty. Further details on the system architecture can be found in Section 5.

State Space

The state space S is the Cartesian product of S_r , the state space of the sUAV S_v , the state space of the victim.

$$S = (S_r, S_t) \quad (4)$$

The sUAV states are defined in Equation 5,

$$S_r = (p_r, o_r, f_c, f_b) \quad (5)$$

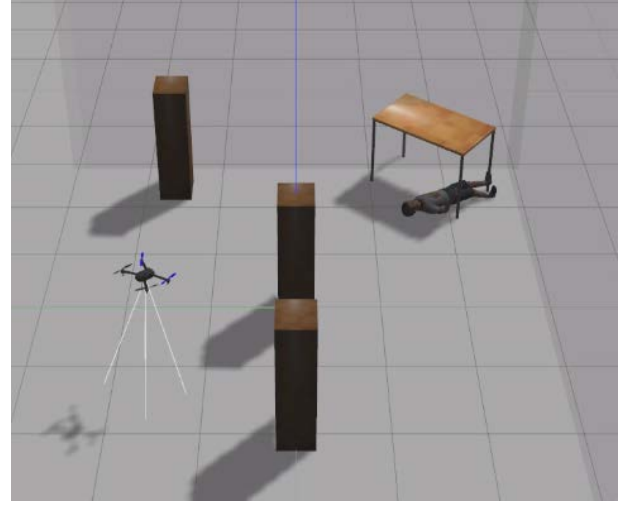


Figure 1: Illustration of the sUAV navigation task to find people in a cluttered environment.

$$p_r = (x_r, y_r, z_r) \quad (6)$$

$$o_r = \psi_r \quad (7)$$

$$f_c = \begin{cases} \text{true} & \text{if sUAV crashes,} \\ \text{false} & \text{otherwise.} \end{cases} \quad (8)$$

$$f_b = \begin{cases} \text{true} & \text{if } p_r > f_{ROI}, \\ \text{false} & \text{otherwise.} \end{cases} \quad (9)$$

where p_r is the position of the robot and o_r is the orientation of the robot in the world Cartesian frame; o_r is simplified to ψ_r , because multi-rotors primarily control their orientation based on their yaw angle only; f_c is a discrete state that defines whether the sUAV has crashed with an obstacle and f_b determines whether the sUAV is flying beyond the limits of the flying area.

The victim states are defined by Equation 10,

$$S_v = (p_v, o_v, f_v) \quad (10)$$

$$p_v = (x_v, y_v, z_v) \quad (11)$$

$$o_v = \psi_v \quad (12)$$

$$f_v = \begin{cases} \text{true} & \text{if target is found,} \\ \text{false} & \text{otherwise.} \end{cases} \quad (13)$$

where p_v is the position of the victim and o_v is the orientation of the victim in the world Cartesian frame; The orientation of a person can likewise be simplified using ψ_v only, the victim's yaw angle; f_v is the discrete state of whether the target has been found by the sUAV.

Table 1: sUAV set of actions

Action	\mathbf{p}_r (m)	\mathbf{o}_r (rad)
Forward	(0.3, 0, 0)	0
Right	(0, 0, 0)	$\pi/4$
Left	(0, 0, 0)	$-\pi/4$
Up	(0, 0, 0.3)	0
Down	(0, 0, -0.3)	0
Hover	(0, 0, 0)	0

Actions

The multi-rotor interacts with the environment using a set of six actions, as shown in Table 1.

For any flight mission, the POMDP assumes that the sUAV is already flying when it starts its autonomous interaction with the environment. Therefore, other common sUAV actions such as autonomous takeoff, landing and return home are addressed by the sUAV autopilot instead.

Transition Function

The motion model of the sUAV is based on the set of actions described above. Changes in rotation can be modelled using the rotation matrix \mathbf{R} of a quad-rotor [32]. Owing to the kinematics of a multi-rotor sUAV, the evaluation of its θ and ϕ angles are discarded in the model. Additionally, an angle deviation φ is added to ψ to incorporate uncertainty caused by pose estimation errors. This uncertainty is modelled as a normal distribution with mean $\mu = \psi_r$ and standard deviation $\sigma = 3.0^\circ$. Thus, \mathbf{R} is simplified as shown in Equation 14.

$$\mathbf{R}_r = \begin{bmatrix} \cos(\psi_r + \varphi_r) & -\sin(\psi_r + \varphi_r) & 0 \\ \sin(\psi_r + \varphi_r) & \cos(\psi_r + \varphi_r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (14)$$

The transformation matrix to model changes in position per time step is defined through Equation 15,

$$\mathbf{p}_{r_{t+1}} = \mathbf{p}_{r_t} + \mathbf{R}_{r_t} \Delta \mathbf{p}_{r_t} \quad (15)$$

which can be expanded as:

$$\begin{bmatrix} x_{r_{t+1}} \\ y_{r_{t+1}} \\ z_{r_{t+1}} \end{bmatrix} = \begin{bmatrix} x_{r_t} \\ y_{r_t} \\ z_{r_t} \end{bmatrix} + \begin{bmatrix} \cos(\psi_{r_t} + \varphi_{r_t}) & -\sin(\psi_{r_t} + \varphi_{r_t}) & 0 \\ \sin(\psi_{r_t} + \varphi_{r_t}) & \cos(\psi_{r_t} + \varphi_{r_t}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_{r_t} \\ \Delta y_{r_t} \\ \Delta z_{r_t} \end{bmatrix} \quad (16)$$

where $\Delta \mathbf{p}_{r_t} = (\Delta x_{r_t}, \Delta y_{r_t}, \Delta z_{r_t})$ is the change in the robot's location from time step t to time step $t + 1$.

The dynamics of the sUAV through changes in position ($\Delta \mathbf{p}_{r_t}$) are modelled using a system identification process. Further details on the calculation of $\Delta \mathbf{p}_{r_t}$ can be found in the Appendix.

Rewards

The system rewards R is defined by Equation 17,

$$R = r_{\text{move}} + r_{\text{crash}} + r_{\text{out}} + r_f + r_d \quad (17)$$

where r_{move} is the cost (negative reward) per move, which encourages the sUAV to find the victim in a minimum number of steps. r_{crash} is the cost for the drone by crashing itself with an obstacle; r_{out} is the cost of flying beyond the explicitly defined Region of Interest (ROI) limits; r_f is the reward if the victim is found; and r_d is the cost given by the Euclidean distance between the sUAV and victim locations, as defined in Equation 18:

$$r_d = -\sqrt{(p_{r_x} - p_{v_x})^2 + (p_{r_y} - p_{v_y})^2 + (p_{r_z} - p_{v_z})^2} \quad (18)$$

The cost values of r_d are directly proportional to the distance between the robot and the victim in the x, y, z axes. Including r_d encourages the sUAV to get closer to the victim and collect better image representations from vision-based sensors. The values for the rest of the rewards were acquired empirically, as shown in Table 2.

Table 2: System rewards for a sUAV target finding task

Reward	Value
r_{move}	-10
r_{crash}	-150
r_{out}	-300
r_f	500

Observation Space

The set of observations O for this problem are defined as:

$$O = (o_{p_r}, o_{p_v}, o_{v_f}) \quad (19)$$

where o_{p_r} is the local pose estimation of the robot; o_{p_v} is the local pose estimation of the victim; and o_{v_f} is a discrete observation defining whether the victim has been detected by the sUAV computer vision object detector. The victim pose observation is received only when the person is found. The object detector executes a pre-trained off-the-shelf deep learning object detector model. The datasets used to fit these models commonly contain thousands of images of classes collected from a frontward-looking camera configuration.

Observation Model

The observation model \mathcal{Z} comprises the estimated sUAV position in the world coordinate frame and the location of the victim if it is detected by the downward-looking camera. The detection of a victim relies on the camera's Field of View (FOV). The modelling of the FOV depends on the sensor properties, the robot's pose and heading observations. First, the horizontal and vertical FOV angles are calculated as defined in Equation 20 and Equation 21.

$$\text{FOV}_V = 2 \tan^{-1} \left(\frac{w}{2f} \right) \quad (20)$$

$$\text{FOV}_H = 2 \tan^{-1} \left(\frac{h}{2f} \right) \quad (21)$$

where w is the sensor width, h is the sensor height, and f is the focal length of the downward-looking camera. The extent of the observed FOV area (or footprint) is calculated as:

$$l_{\text{top}} = p_{r(z)} \cdot \tan(\alpha + 0.5 \cdot \text{FOV}_H) \quad (22)$$

$$l_{\text{bottom}} = p_{r(z)} \cdot \tan(\alpha - 0.5 \cdot \text{FOV}_H) \quad (23)$$

$$l_{\text{left}} = p_{r(z)} \cdot \tan(\alpha + 0.5 \cdot \text{FOV}_V) \quad (24)$$

$$l_{\text{right}} = p_{r(z)} \cdot \tan(\alpha - 0.5 \cdot \text{FOV}_V) \quad (25)$$

where l_* is the footprint extent of any collected image and α is the gimbal angle of the camera from the vertical (i.e. 0 degrees), as depicted in Figure 2.

The calculation of the footprint point coordinates with its center in the origin is defined as:

$$c_1 = (l_{\text{top}}, l_{\text{left}}, 0) \quad (26)$$

$$c_2 = (l_{\text{top}}, l_{\text{right}}, 0) \quad (27)$$

$$c_3 = (l_{\text{bottom}}, l_{\text{right}}, 0) \quad (28)$$

$$c_4 = (l_{\text{bottom}}, l_{\text{left}}, 0) \quad (29)$$

A transformation matrix is then calculated to locate the point coordinates within the sUAV reference frame:

$$\begin{bmatrix} c'_x \\ c'_y \\ c'_z \end{bmatrix} = \begin{bmatrix} p_{r(x)} \\ p_{r(y)} \\ p_{r(z)} \end{bmatrix} + \begin{bmatrix} \cos(\psi_r) & -\sin(\psi_r) & 0 \\ \sin(\psi_r) & \cos(\psi_r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \quad (30)$$

A victim is predicted to be within the camera's FOV if a belief location point of the person is positioned inside the rectangular polygon from the group of c points. This calculation is performed as the sum of the angles between the victim belief position point and each pair of points that comprise the rectangle [33], as defined in Equation 31.

$$\theta = \sum_{i=1}^4 \left\{ \tan^{-1} \left[\frac{c_{i+1}(y) - p_t(y)}{c_{i+1}(x) - p_t(x)} \right] - \tan^{-1} \left[\frac{c_i(y) - p_t(y)}{c_i(x) - p_t(x)} \right] \right\} \quad (31)$$

From this formulation, the victim location point is predicted to be inside of the camera's FOV if $\theta = 2\pi$. Perfect accuracy is, however, assumed here from any vision-based model implemented in the detection subsystem. Target detection uncertainty from computer vision, and specifically, pre-trained deep learning detectors, occurs from different factors, including image noise, illumination conditions, image resolution, image representations of people from the dataset and camera configuration. Even though these object detection models can be improved using different techniques, some of these factors (that cause uncertainty) can be simulated by extending the target finding modelling. Taking into account

that off-the-shelf object detection models give their best results when achieving close image representations from their trained datasets (e.g. ImageNet, COCO), a positive person detection is simulated if the sUAV and victim heading angles are similar (i.e. aligned) each other, as defined in Equation 32.

$$\text{target} = \begin{cases} \text{found} & \text{if } \theta = 2\pi \text{ and } |\psi_r - \psi_t| < 30^\circ \\ \text{not found} & \text{otherwise} \end{cases} \quad (32)$$

From the observation model described above and the reward function defined in Equation 17 and Equation 18, target detection uncertainty from deep learning object detectors is expected to be reduced by encouraging the robot to fly at a close distance between the victim and the drone itself. Furthermore, the sUAV will also adjust its orientation to match potentially a scene representation that increases the likelihood of the detector to detect a person.

5. IMPLEMENTATION

System Architecture

The proposed framework comprises four modules that interact with a multi-rotor sUAV under simulation, as shown in Figure 3.

Software in the Loop module—The sUAV is controlled using PX4, an open source software developed by the Dronecode Project [34]. The PX4 architecture is comprised of two layers: the flight stack layer and the middleware layer. The flight stack layer contains a pipeline of flight controllers for a rich set of UAVs (multi-rotors, fixed-wing and VTOL) and altitude and position estimators. These estimators usually make predictions from one or multiple sensor inputs such as IMU and GPS. The PX4 flight controller follows a feedback control loop process for position and velocity set-point values, a PID controller, and feedback signals from the estimators. The middleware layer contains the device drivers for the sUAV sensors, communication interfaces, and a simulation layer.

The simulation layer (PX4 Software in the Loop (SITL)) ports the PX4 architecture in a simulated sUAV platform and environment to a local machine. The sUAV (a 3DR Iris) and the downward-looking camera (an ov7251) are simulated under the Gazebo simulator.

Hardware in the Loop module—Given hardware limitations on sUAVs, an Intel® Neural Compute Stick is integrated into the system. The stick is a plug and play Vision Processing Unit (VPU) device, an optimised microprocessor that boosts inference from deep learning models. Intel provides a specific software toolkit to obtain performance gains through the OpenVINO library. OpenVINO supports a range of deep learning frameworks such as TensorFlow, Caffe and PyTorch, and optimised versions of OpenCV and OpenVX for standard image processing operations. The source code for target detection with OpenVINO contains ROS bindings in Python, allowing the use of this hardware for both real tests and simulation environments via PX4 SITL and Gazebo. The latter provides, therefore, support under a Hardware in the Loop (HITL) interface.

Detection module—The detection module comprises a deep learning object detector. The selected detector is an open-

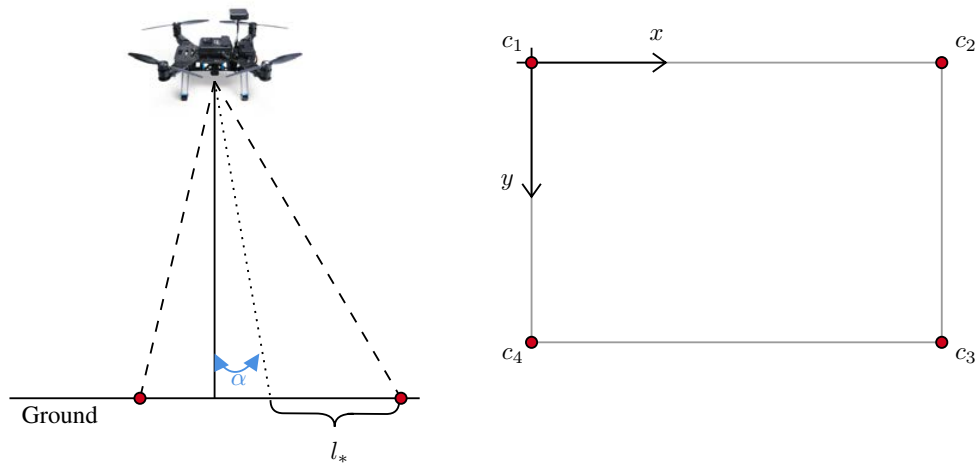


Figure 2: FOV and 2D image representation from a vision-based camera pointing to the ground with an angle α from the vertical.

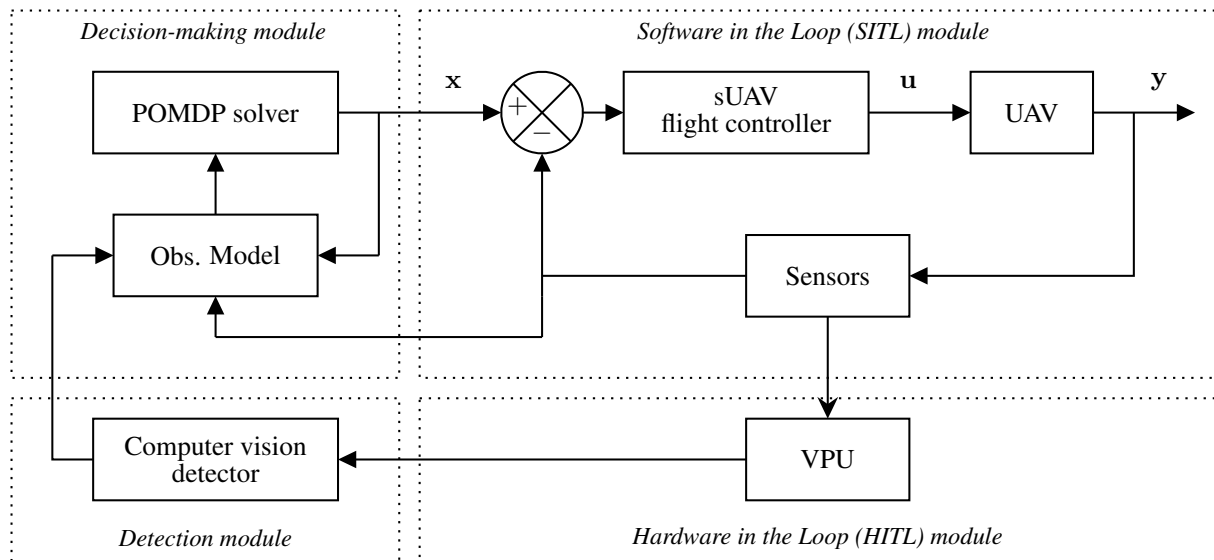


Figure 3: Proposed system architecture under a simulation environment.

source instance of the Google MobileNet Single Shot Multi-box Detector (SSD) architecture [35]. The model is deployed in caffe and fit using pre-trained weights from the PASCAL VOC0712 dataset, achieving a mean average precision of 72.7%. For every read frame, the detection module subscribes to a downward-looking camera ROS topic included in the Iris model. Input frames are resized into dimensions of 300×300 . Any object detections with a confidence value greater than 30% from the output layer are displayed in the processed frame. If the chosen class (*i.e.* the object of interest) is displayed, the position of the object is estimated following the formulation described in Section 4. An illustration of a person detection from the deep learning model is shown in Figure 4.

Decision-making module—The POMDP formulation, which was described in Section 4, is computed using the TAPIR toolkit [36]. The toolkit incorporates the online ABT solver

which handles continuous states. Additionally, TAPIR includes a ROS interface that eases communication between the online solver and the sUAV flight controller (PX4 SITL).

Experimentation Setup

The system was tested in a cluttered and GPS-denied simulated environment using the Gazebo simulator. The goal of the sUAV is to find a child who needs assistance to contextualise the experiments under SAR situations. The flying area, with dimensions 6×6 m in length and width and 3 m in height, contains several column obstacles using cardboard boxes and a table, as shown in Figure 1.

The test scenario contains three cardboard obstacles in form of columns placed throughout the scene, a table that partially occludes the victim, a safety net which delimits the flying area and a child dummy to be found. For all the experiments, the child is always located under the table at world Cartesian

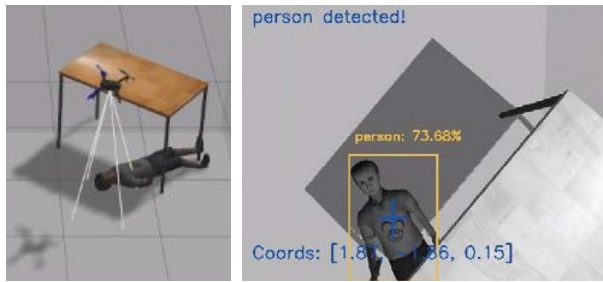


Figure 4: Detection example of a child dummy from a downward-looking camera under Gazebo SITL. the sUAV takes actions to align itself with the child and minimise uncertainty.

coordinates $(2.0, -2.0, 0.2)$. Visuals of the child come from a downward-looking camera attached to the Iris UAV frame. The sUAV uses the take-off and landing modes from the PX4 autopilot, and start the navigation task at world Cartesian coordinates $(-2.0, -2.0, 2.0)$.

6. RESULTS

The autonomous decision-making system and target detection uncertainty is evaluated with two types of setups. The first setup declares different levels of child location uncertainty within the flying area. The second setup evaluates the system under different child orientation configurations.

Location Uncertainty

Location uncertainty for the child is defined through three case studies, illustrate in Figure 5 and described as follows:

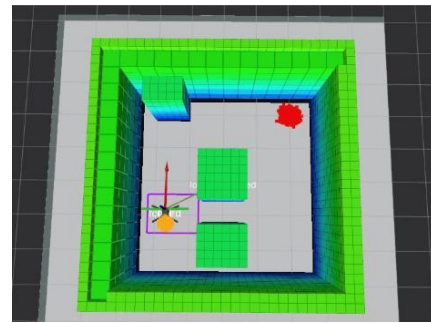
- A *single location estimation*: the belief states of the victim location are represented as a point cloud, sorted as a normal distribution ($\mu = 0m, \sigma = 0.5m$) at one specific region (top right corner from Figure 5a).
- B *multiple location estimation*: the belief states of victim location now include two possible locations, which are sorted as a normal distribution ($\mu = 0m, \sigma = 0.5m$) and represented as two point clouds (Figure 5b).
- C *uniform location estimation*: the belief states of the victim location are uniformly distributed into the flying area, assuming thus, that the location of the victim is unknown (Figure 5c).

Each case study was run 40 times, with a mean duration per time step of 1.049 seconds. The success rate per case study is illustrated in Table 3.

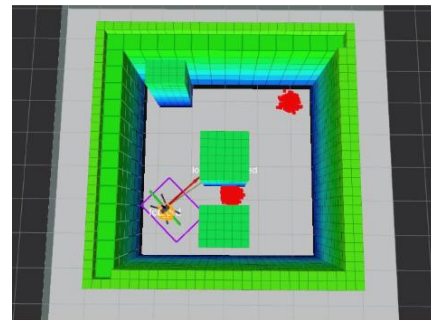
Table 3: Location uncertainty success rates under SITL.

Case Study	Success Rate (Target found)	Failure Rate (Crash)	Failure Rate (Timeout)
A	100%	0%	0%
B	87.5%	12.5%	0%
C	52.5%	2.5%	45%

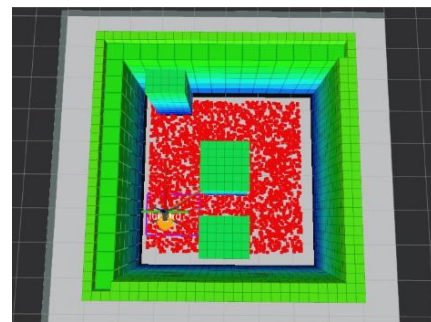
Overall, the success rate for case study A is remarkable, achieving 100% of successful runs. The sUAV was able to follow the reward structure with the absence of crash reports or flying beyond the ROI limits. The drone planned a trajectory that ensured matching its location above the target



(a)



(b)



(c)

Figure 5: SITL environment illustration that overlays an occupancy map (in green) and the belief map (in red).

point cloud first, followed by adjusting its heading angle ψ to filter target location particles until the detection subsystem found the target. Similarly, case study B, which encouraged the sUAV to fly nearby two of the cardboard boxes, achieved a success rate of 87.5%. As shown in Figure 5b, a second point cloud of child location estimations was added in the middle of two cardboard columns. Even though the drone was able to adjust its trajectory after filtering all the particles in zones without the presence of the child, drifts in the x and y axes were clearly visible while the sUAV spun next to the columns. For case study C, the number of failures became evident by exceeding the maximum flight time of the sUAV. The evaluation of the sUAV and target heading angles in the observation model provoked an increase in the number of time steps to discard particles, as demonstrated in Figure 6.

The number of time steps was consistent for case study A, with a median of 35 steps and an inter-quartile range of four steps. When the number of target particles covers a bigger

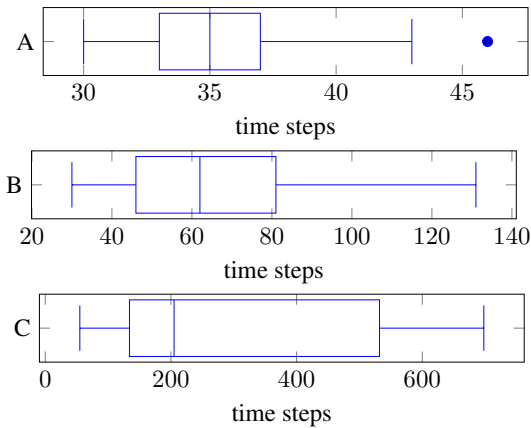


Figure 6: Time steps of successful runs under location uncertainty.

region in the flying area, so is the number of required actions (spins) to evaluate different heading angles. In fact, the median number of time steps almost doubled for case study *B* and a evident greater variability on time steps for case study *C*.

Orientation Uncertainty

The sUAV navigation task and the POMDP observation model were also evaluated by varying the heading (or orientation) of the child. The child was placed using four orientation configurations, as shown in Figure 7. For each configuration, the distribution of possible locations (location uncertainty) for the dummy followed the settings of case study *A*.

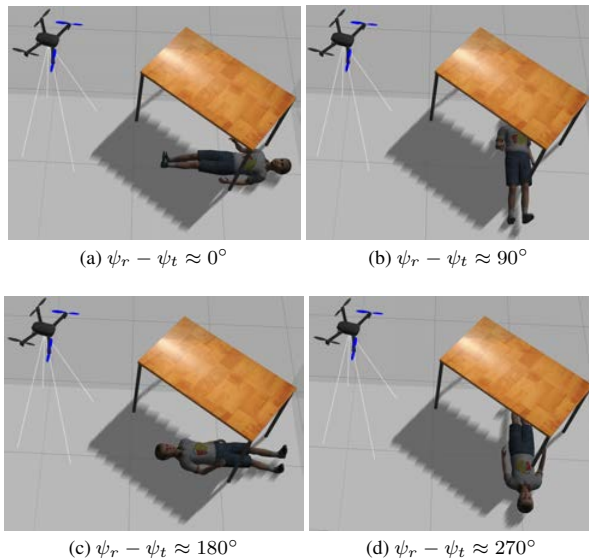


Figure 7: Dummy child at different orientation configurations.

The success rate per case study is illustrated in Table 4.

The sUAV was able to navigate and detect the child with a mean success rate of 98%. Indeed, the lost child was

Table 4: Success rates for defined levels of target uncertainty under SITL.

Case Study	Success Rate (Target found)	Failure Rate (Crash)	Failure Rate (Timeout)
A	100%	0%	0%
B	97.5%	2.5%	0%
C	100%	0%	0%
D	95%	5.0%	0%

always found when the particle distribution surrounded the child location at different orientations. The distribution of time steps for all the episodes is illustrated in Figure 8.

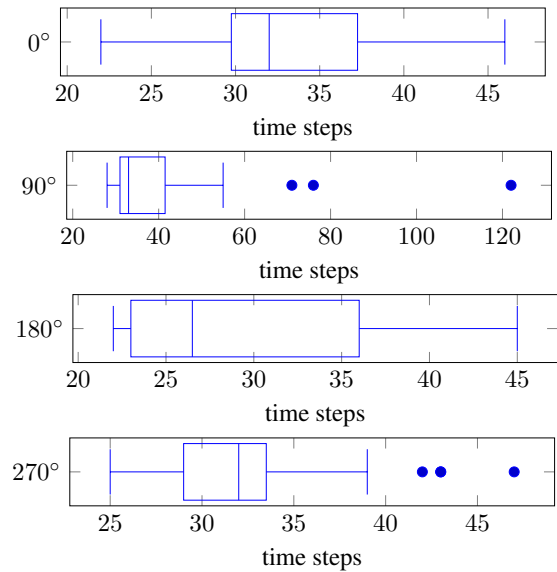


Figure 8: Time steps of successful runs under orientation uncertainty.

The median number of time steps for the sUAV while collecting child frames was the lowest when its image representations were aligned with the sUAV ($\psi_r - \psi_t \approx 0^\circ$), as depicted in Figure 4. Conversely, an alignment difference of 180° resulted in a bigger number of time steps to detect the child. The big variance and outliers in time steps for alignment differences of 90° and 270° happened because of the partial occlusion from the table, requiring a higher number of actions for the sUAV to get a clear visual of the child. This variance also illustrates the adaptability of the sUAV in a environment under uncertainty, given the way PX4 Gazebo SITL emulates stochastic odometry errors in the drone and external disturbances in the environment.

7. CONCLUSIONS

This paper introduced a solution to the problem of searching victims with a prior probability with respect to the likely location of the victims. The problem was formulated as a Partially Observable Markov Decision Process to solve the task as a multi-objective sUAV navigation problem that incorporates uncertainty and partial observability represented in state belief. The proposed framework was validated on SAR mission simulated in an indoor cluttered environment under different levels of location and orientation uncertainties. The

experiments were tested using PX4, Gazebo SITL, ROS, an Intel VPU for deep learning inference (HITL) and TAPIR.

The problem formulation and system architecture constitute a substantial extension on previous contributions on autonomous sUAV decision-making for target finding under uncertainty. Specifically, the proposed system presents an extension on the work of Vanegas et al. [37] by incorporating:

- An observation model that incorporates target detection uncertainty for vision-based object detectors.
- Experimentation close to real-world conditions under SITL and HITL capabilities.
- Onboard target detection using deep learning and VPUs for real-time inference under resource-constrained hardware.

The results demonstrate the capability of the proposed system to deal with high levels of environment and target detection uncertainty as well as progress towards automating surveillance operations for applications that require rapid intervention such as SAR.

APPENDIX

System Identification

Motion response of the plant (*i.e.* sUAV) is collected by measuring the robot's position values $y(t)$ under a step response $r(t)$ in x , z and ψ from the world coordinate frame. As an illustration, the process to identify the system under a step position response in x is illustrated herewith. First, the transfer function of the plant was calculated using the System Identification Toolbox™ from MATLAB®, as shown in Equation 33.

$$F(s) = \frac{0.204s + 1.136}{s^2 + 1.253s + 1.134} \quad (33)$$

The calculation of the plant in time discrete is done using the Tustin approximation method, which is defined in Equation 34,

$$s \approx \frac{2(z-1)}{T_s(z+1)} \quad (34)$$

where T_s is the sampling period. Assuming $T_s = 0.1s$, the discretised plant $F(z)$ equals:

$$F(z) = \frac{0.01224 + 0.005333z^{-1} - 0.006905z^{-2}}{1 - 1.872z^{-1} + 0.8824z^{-2}} \quad (35)$$

The difference equation from $F(z)$ is calculated by applying the inverse Z transform:

$$F(z) = \frac{Y(z)}{R(z)}$$

$$Y(z) = \frac{(A_0 + A_1z^{-1} + A_2z^{-2})R(z)}{1 + B_1z^{-1} + B_2z^{-2}}$$

$$y(k) = A_0r(k) + A_1r(k-1) + A_2r(k-2) - B_1y(k-1) - B_2y(k-2) \quad (36)$$

where

$$A_0 = 0.012237830217107$$

$$A_1 = 0.005333276901521$$

$$A_2 = -0.006904553315587$$

$$B_1 = -1.871779712793530$$

$$B_2 = 0.882425299507294$$

The value of Δx_{r_t} is ultimately calculated by iterating Equation 36 every T_s seconds until reaching the total duration per time step.

ACKNOWLEDGMENTS

The authors would like to thank The Commonwealth Scientific and Industrial Research Organisation (CSIRO) through the CSIRO Data61 PhD and Top Up Scholarships (Agreement 50061686), and the Australian Research Council (ARC) through the ARC Discovery Project 2018 'Navigating under the forest canopy and in the urban jungle' for funding this research work.

REFERENCES

- [1] G. Pajares, "Overview and current status of remote sensing applications based on unmanned aerial vehicles (uavs)," *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 4, pp. 281–330, Apr. 2015.
- [2] B. J. Stark and Y. Q. Chen, "Remote sensing methodology for unmanned aerial systems," in *Unmanned Aircraft Systems*, E. M. Atkins, A. Ollero, and A. Tsourdos, Eds. New York, USA: Wiley, 2016, ch. 2, pp. 17–27.
- [3] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, *On integrating unmanned aircraft systems into the national airspace system: Issues, challenges, operational restrictions, certification, and recommendations*. Dordrecht: Springer Netherlands, 2012.
- [4] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, "A review of deep learning methods and applications for unmanned aerial vehicles," *Journal of Sensors*, vol. 2017, pp. 1–13, 2017.
- [5] M. Bouhali, F. Shamani, Z. E. Dahmane, A. Belaidi, and J. Nurmi, "Fpga applications in unmanned aerial vehicles - a review," in *Lecture Notes in Computer Science*, 2017, vol. 10216 LNCS, pp. 217–228.
- [6] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based uav path planning," *Knowledge-Based Systems*, vol. 158, pp. 54–64, Oct. 2018.
- [7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.
- [8] Australian Bureau of Statistics, "Understanding natural hazard impacts on australia," Canberra, 2008. [Online]. Available: <https://www.abs.gov.au/ausstats/abs@.nsf/Previousproducts/1301.0FeatureArticle42008?>

- opendocument{&}tabname=Summary{&}prodno=1301.0{&}issue=2008{&}num={&}view=
- [9] ———, “Natural disasters in australia,” Canberra, 2008. [Online]. Available: <https://www.abs.gov.au/ausstats/abs@.nsf/7d12b0f6763c78caca257061001cc588/feb2ab6de16171eca2570de0005871b!OpenDocument>
 - [10] R. Z. B. Bravo, A. Leiras, and F. L. Cyrino Oliveira, “The use of uavs in humanitarian relief: An application of pomdp-based methodology for finding victims,” *Production and Operations Management*, vol. 28, no. 2, pp. 421–440, Feb. 2019.
 - [11] M. Erdelj and E. Natalizio, “Uav-assisted disaster management: Applications and open issues,” in *International Conference on Computing, Networking and Communications*. IEEE, Feb. 2016, pp. 1–5.
 - [12] R. Murphy, “Human–robot interaction in rescue robotics,” *Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 2, pp. 138–153, May 2004.
 - [13] C. Sampedro, A. Rodriguez-Ramos, H. Bavle, A. Carrio, P. de la Puente, and P. Campoy, “A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques,” *Journal of Intelligent & Robotic Systems*, pp. 1–27, Jul. 2018.
 - [14] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, “Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 46–56, Sep. 2012.
 - [15] V. San Juan, M. Santos, and J. M. Andújar, “Intelligent uav map generation and discrete path planning for search and rescue operations,” *Complexity*, vol. 2018, pp. 1–17, 2018.
 - [16] M. J. Kochenderfer, *Decision making under uncertainty: theory and application*. Cambridge, MA: MIT Press, 2015.
 - [17] S. A. Miller, Z. A. Harris, and E. K. Chong, “A pomdp framework for coordinated guidance of autonomous uavs for multitarget tracking,” *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 1, p. 724597, Dec. 2009.
 - [18] S. Ragi and E. K. P. Chong, “Uav path planning in a dynamic environment via partially observable markov decision process,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2397–2412, Oct. 2013.
 - [19] Y. Zhao, X. Wang, W. Kong, L. Shen, and S. Jia, “Decision-making of uav for tracking moving target via information geometry,” in *Chinese Control Conference*. Chengdu, China: IEEE, Jul. 2016, pp. 5611–5617.
 - [20] C. M. Eaton, E. K. Chong, and A. A. Maciejewski, “Robust uav path planning using pomdp with limited fov sensor,” in *Conference on Control Technology and Applications (CCTA)*. Hawaii, USA: IEEE, Aug. 2017, pp. 1530–1535.
 - [21] F. Vanegas and F. Gonzalez, “Enabling uav navigation with sensor and environmental uncertainty in cluttered and gps-denied environments,” *Sensors*, vol. 16, no. 5, p. 666, May 2016.
 - [22] D. Silver and J. Veness, “Monte-carlo planning in large pomdps,” *Neural Information Processing Systems*, pp. 1–9, 2010.
 - [23] H. Kurniawati and V. Yadav, “An online pomdp solver for uncertainty planning in dynamic environment,” in *Springer Tracts in Advanced Robotics*, 2016, vol. 114, pp. 611–629.
 - [24] K. P. Valavanis and G. J. Vachtsevanos, “Future of unmanned aviation,” in *Handbook of Unmanned Aerial Vehicles*, K. P. Valavanis and G. J. Vachtsevanos, Eds. Dordrecht: Springer Netherlands, 2015, ch. 126, pp. 2993–3009.
 - [25] S. Ragi and E. K. P. Chong, “Uav guidance algorithms via partially observable markov decision processes,” in *Handbook of Unmanned Aerial Vehicles*, K. Valavanis and G. Vachtsevanos, Eds. Dordrecht: Springer Netherlands, 2015, ch. 73, pp. 1775–1810.
 - [26] C. Ponzoni Carvalho Chanel, A. Albore, J. T’Hooft, C. Lesire, and F. Teichteil-Königsbuch, “Ample: an anytime planning and execution framework for dynamic and uncertain problems in robotics,” *Autonomous Robots*, vol. 43, no. 1, pp. 37–62, Jan. 2019.
 - [27] M. Chen, E. Frazzoli, D. Hsu, and W. S. Lee, “Pomdp-lite for robust robot planning under uncertainty,” *International Conference on Robotics and Automation*, pp. 5427–5433, Feb. 2016.
 - [28] U. Ilhan, L. Gardashova, and K. Kilic, “Uav using dec-pomdp model for increasing the level of security in the company,” *Procedia Computer Science*, vol. 102, pp. 458–464, 2016.
 - [29] S. C. W. Ong, S. W. Png, D. Hsu, and W. S. Lee, “Pomdps for robotic tasks with mixed observability,” in *Robotics: Science and Systems V*. Seattle, US: Robotics: Science and Systems Foundation, Jun. 2009.
 - [30] C. Chanel, F. Teichteil-Königsbuch, and C. Lesire, “Multi-target detection and recognition by uavs using online pomdps,” in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*. Bellevue, Washington: AAAI Press, 2013, pp. 1381–1387.
 - [31] A. Al-Kaff, D. Martín, F. García, A. de la Escalera, and J. María Armingol, “Survey of computer vision algorithms and applications for unmanned aerial vehicles,” *Expert Systems with Applications*, vol. 92, pp. 447–463, Feb. 2018.
 - [32] A. Chovancová, T. Fico, L. Chovanec, and P. Hubinsk, “Mathematical modelling and parameter identification of quadrotor (a survey),” *Procedia Engineering*, vol. 96, pp. 172–181, 2014.
 - [33] P. Bourke, “Polygons and meshes,” 1997. [Online]. Available: <http://paulbourke.net/geometry/polygonmesh/>
 - [34] L. Meier, D. Honegger, and M. Pollefeys, “Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms,” in *International Conference on Robotics and Automation*, no. June. IEEE, May 2015, pp. 6235–6240.
 - [35] Chuanqi305, “Caffe implementation of google mobilenet ssd detection network, with pretrained weights on voc0712 and map=0.727.” 2019. [Online]. Available: <https://github.com/chuanqi305/MobileNet-SSD>
 - [36] D. Klimenko, J. Song, and H. Kurniawati, “Tapir: A software toolkit for approximating and adapting pomdp solutions online,” in *Australasian Conference on*

Robotics and Automation, Melbourne, Australia, 2014, pp. 1–9.

- [37] F. Vanegas, D. Campbell, M. Eich, and F. Gonzalez, “Uav based target finding and tracking in gps-denied and cluttered environments,” in *International Conference on Intelligent Robots and Systems*. Daejeon, South Korea: IEEE/RSJ, Oct. 2016, pp. 2307–2313.

BIOGRAPHY



Juan Sandino holds a BEng (Mechatronics) and is currently undertaking a PhD in robotics and autonomous systems at QUT, Australia. His primary interests comprise autonomous UAV decision-making, machine learning and computer vision for UAV remote sensing, with a focus on hyperspectral and high-resolution image processing. Juan has worked for research projects in

biosecurity, environment monitoring and search and rescue.



Fernando Vanegas received his B.S. in Mechatronics Engineering from UMNG in 2004 and M.Sc. in Electrical Engineering from Halmstad University in 2008. He is currently a Ph.D. candidate in Robotics and Autonomous Systems at The Australian Research Centre for Aerospace Automation and Queensland University of Technology. His current research activities include motion plan-

ning for UAV in cluttered and uncertain environments modelled as POMDPs.



Felipe Gonzalez holds a BEng (Mech) and a PhD from the University of Sydney. Gonzalez is an Associate Professor at the School of Electrical Engineering and Robotics (ECCS), Science and Engineering Faculty with a passion for innovation in the fields of aerial robotics and automation. Gonzalez interest is in creating aerial robots, drones or UAVs that possess a high level of cognition

using efficient on-board computer algorithms using advanced optimisation and game theory approaches that assist us to understand and improve our physical and natural world. Dr Gonzalez lead the Airborne Sensing Lab at QUT.



Frederic Maire received the M.Sc. degree in pure mathematics and computer science engineering in 1989 and the Ph.D. degree in discrete mathematics from the Universite Pierre et Marie Curie, Paris 6, France, in 1993. He is currently a Senior Lecturer with the School of Electrical Engineering and Robotics, Queensland University of Technology, Brisbane, Australia. His

research interests include computer vision and robotics.

Chapter 4

UAV Framework for Autonomous Onboard Navigation and People/Object Detection in Cluttered Indoor Environments

WITH UAVs providing a flexibly deployed eye in the sky to inaccessible or dangerous areas, first responder efforts are optimised in emergency scenarios. However, it is the suboptimal reliance on communication systems from the UAV, and the need for outstanding photo-interpretation skills from the UAV operator about the environment, location of obstacles, and objects to detect (*i.e.*, victims) that ultimately compromises aircraft behaviour in indoor flight operations, particularly where GNSS signals are weak or absent.

This chapter presents the second published paper from this research and establishes a UAV framework for autonomous navigation in cluttered indoor scenarios under uncertainty and partial observability. This framework design allocates required computational resources to run the POMDP-based motion planner and inference of CNN object detectors onboard a companion computer attached to the UAV frame. This paper extends the problem formulation, software, and hardware tools presented in paper 1, Chapter 3. The framework was tested in a SAR case study where a sub 2 kg UAV detected and located victims inside a simulated office building. The framework was successfully evaluated using HIL simulations and real flight tests, ensuring personal safety of human operators by enabling the UAV to remotely survey dangerous environments.

Statement of Contribution of Co-Authors for Thesis by Published Paper

The authors listed below have certified that:

1. they meet the criteria for authorship and that they have participated in the conception, execution, or interpretation, of at least that part of the publication in their field of expertise;
2. they take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;
3. there are no other authors of the publication according to these criteria;
4. potential conflicts of interest have been disclosed to (a) granting bodies, (b) the editor or publisher of journals or other publications, and (c) the head of the responsible academic unit, and
5. they agree to the use of the publication in the student's thesis and its publication on the [QUT's ePrints site](#) consistent with any limitations set by publisher requirements.

In the case of this chapter:

J. Sandino, F. Vanegas, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez, "UAV framework for autonomous onboard navigation and people/object detection in cluttered indoor environments," *Remote Sensing*, vol. 12, no. 20, p. 3386, Oct. 2020. DOI: [10.3390/rs12203386](https://doi.org/10.3390/rs12203386).

Contributor	Statement of Contribution ¹
Juan Sandino QUT Verified Signature Date: <u>2022-05-20</u>	Contributed on: conceptualisation, methodology, software, validation, formal analysis, investigation, data curation, writing-original draft preparation and visualisation.
Fernando Vanegas	Contributed on: conceptualisation, methodology, software, hardware conceptualisation and integration, investigation, writing-review, and editing.
Frederic Maire	Contributed on: conceptualisation, writing-review and editing, and supervision.
Peter Caccetta	Contributed on: writing-review and editing, supervision, and funding acquisition.
Conrad Sanderson	Contributed on: writing-review and editing, and supervision.
Felipe Gonzalez	Contributed on: conceptualisation, methodology, resources, writing-review and editing, supervision, project administration, and funding acquisition.







Principal Supervisor Confirmation		
I have sighted email or other correspondence from all co-authors confirming their certifying authorship.		
<i>Felipe Gonzalez</i>	QUT Verified Signature	2022-05-20
_____ Name	_____ Signature	_____ Date

¹ Contributions follow the CRediT taxonomy. Further details can be found at <https://credit.niso.org/>.



Article

UAV Framework for Autonomous Onboard Navigation and People/Object Detection in Cluttered Indoor Environments

Juan Sandino ^{1,2,3,*} , Fernando Vanegas ^{1,3} , Frederic Maire ^{1,3} , Peter Caccetta ² ,
Conrad Sanderson ²  and Felipe Gonzalez ^{1,3} 

- ¹ School of Electrical Engineering and Robotics, Queensland University of Technology (QUT), 2 George Street, Brisbane City, QLD 4000, Australia; f.vanegasalvarez@qut.edu.au (F.V.); f.maire@qut.edu.au (F.M.); felipe.gonzalez@qut.edu.au (F.G.)
 - ² Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Building 101, Clunies Ross Street, Black Mountain, ACT 2601, Australia; peter.caccetta@data61.csiro.au (P.C.); conrad.sanderson@data61.csiro.au (C.S.)
 - ³ QUT Centre for Robotics (QCR), Queensland University of Technology (QUT), Level 11, S Block, 2 George Street, Brisbane City, QLD 4000, Australia
- * Correspondence: j.sandino@qut.edu.au

Received: 7 September 2020; Accepted: 12 October 2020; Published: 16 October 2020



Abstract: Response efforts in emergency applications such as border protection, humanitarian relief and disaster monitoring have improved with the use of Unmanned Aerial Vehicles (UAVs), which provide a flexibly deployed eye in the sky. These efforts have been further improved with advances in autonomous behaviours such as obstacle avoidance, take-off, landing, hovering and waypoint flight modes. However, most UAVs lack autonomous decision making for navigating in complex environments. This limitation creates a reliance on ground control stations to UAVs and, therefore, on their communication systems. The challenge is even more complex in indoor flight operations, where the strength of the Global Navigation Satellite System (GNSS) signals is absent or weak and compromises aircraft behaviour. This paper proposes a UAV framework for autonomous navigation to address uncertainty and partial observability from imperfect sensor readings in cluttered indoor scenarios. The framework design allocates the computing processes onboard the flight controller and companion computer of the UAV, allowing it to explore dangerous indoor areas without the supervision and physical presence of the human operator. The system is illustrated under a Search and Rescue (SAR) scenario to detect and locate victims inside a simulated office building. The navigation problem is modelled as a Partially Observable Markov Decision Process (POMDP) and solved in real time through the Augmented Belief Trees (ABT) algorithm. Data is collected using Hardware in the Loop (HIL) simulations and real flight tests. Experimental results show the robustness of the proposed framework to detect victims at various levels of location uncertainty. The proposed system ensures personal safety by letting the UAV to explore dangerous environments without the intervention of the human operator.

Keywords: partially observable Markov decision process (POMDP); machine learning; search and rescue (SAR); probabilistic decision-making; embedded systems; computer vision; autonomous system; unmanned aerial system (UAS); path planning; artificial intelligence

1. Introduction

High resolution satellite and aircraft imagery has and can assist in relief efforts after natural disasters such as earthquakes, floods, landslides and bush/forest fires. Earthquakes alone are

estimated to have claimed the lives of almost 1.87 million people in the last century [1]. Research has demonstrated how urbanisation can increase risks to population from natural disasters in vulnerable areas [2]. Recent studies indicate that more than 40% of human fatalities caused by earthquakes occur by weak and collapsed building structures [3]. Therefore, studies on improving disaster management efforts in urban and peri-urban indoor areas are key to decrease the number of fatalities.

Intelligent aerial platforms such as Unmanned Aerial Vehicles (UAVs)—commonly referred as drones—have improved response efforts in time-critical applications such as border protection, humanitarian relief and disaster monitoring [4]. Small UAVs—UAVs whose Maximum Take-off Weight (MTOW) is lower or equal to 13.5 kg [5]—have offered portability and versatility to their users thanks to advances in autonomous behaviours such as obstacle avoidance, highly stable take-off, landing, hovering and waypoint flight modes, as well as extensive payload adaptability [6,7].

The contribution of UAVs in time-critical applications such as Search and Rescue (SAR) has become significant in recent years. Reported key areas on the use of UAVs post-disasters include aerial monitoring of damage evaluation, localisation of victims, SAR logistics and cargo delivery [8,9]. UAVs have also assisted through the rapid post-disaster assessment of damaged buildings after an earthquake [10,11], the custom design of defibrillator payloads [12,13] and the deployment of first aid kits in remote areas [14]. Recent research has also showed how UAVs can provide fast assessments on the identification of victims and their conditions. A remote sensing life signs detector for multiple victims, for instance, has been developed using a UAV and a vision-based algorithm [15]. Similarly, automated detection of victims using computer vision is now possible by manually flying small UAVs above them [16]. Despite these advances, operational software limitations of UAVs to navigate autonomously in unknown environments have impeded their use in more real-world scenarios [17,18]. Developing autonomous decision-making processes in UAVs is a challenging issue that has attracted the attention of the research community [19].

Whenever an emergency situation occurs, it is of utmost importance to evaluate the environment conditions to identify critical zones that require immediate intervention and to coordinate adequate response [20]. Real-world emergency environments are dynamic, complex, unknown or partially known. Adding cognition capabilities in UAVs for environments under uncertainty is a problem that can be evaluated using decision-making theory. Applied theory on decision making addresses not only autonomous UAV navigation problems but it is also used in fields such as game theory, navigation strategies, Bayesian principles, multi-objective decision-making, Markov Decision Processes (MDP) and Partially Observable MDPs (POMDP) [21–23]. Research has shown how modelling UAV navigation problems with POMDPs in environments with high levels of uncertainty is a suitable approach. For instance, Vanegas and Gonzalez [24] developed an autonomous navigation framework for a GNSS-denied cluttered environment using small UAVs. The framework was evaluated using Partially Observable Monte Carlo Planning (POMCP) [25] and Augmented Belief Trees (ABT) [26], two of the fastest POMDP online solvers known up to date. Despite the potential shown in the proposed framework by giving the UAV the capability of making decisions in seconds with ABT, the authors narrowed their tests using black and white rectangular augmented reality markers [27]. The POMDP solvers were also run using an external workstation and their action commands sent to the UAV. As sustained by Carrio et al. [19] and Valavanis and Vachtsevanos [28], it is undesirable to depend on communication modules for autonomous UAV navigation because if such modules fail, the UAV performance might become seriously compromised.

Research by Ragi and Chong [29,30] also presented significant progress, where dynamic path-planning in multiple target tracking was accomplished using POMDPs. Reported progress towards fully autonomous UAVs by including path planning, collision avoidance, external wind disturbance effects and tracking evasive threats in their problem formulation, showed the prospects of modelling multi-objective problems using POMDPs. The tests conducted by the authors were carried out in simulation environments only and did not provide evidence on the use of the framework in a real-world UAV target tracking application. Similarly, Bravo et al. [20] and Waharte

and Trigoni [31] tested humanitarian relief operations with POMDP frameworks in simulation, suggesting the demand to validate existing approaches with real flight tests and more realistic disaster situations. Similar advances on autonomous UAV navigation using POMDP-based theory include POMDP-lite [32], Anytime Meta PLannEr (AMPLE) [33], Mixed Observability Markov Decision Process (MOMDP) [34] and decentralised POMDP [35]. Nevertheless, most of the proposed solvers have only been tested in simulation environments. Validation of these approaches with real UAV flight tests in complex environments is still an unresolved gap [36,37].

Literature on onboard autonomous UAV decision-making in GNSS-denied environments and time-critical applications using POMDPs is scarce. The study from Chanel et al. [38] shows one of the most significant approaches through the development of a multi-car detection application using an optimised UAV framework. The designed framework allows running a POMDP onboard the UAV and optimised during execution. However, missing experimentation details such as the UAV frame, drivers, companion computer and algorithms for computer vision have impeded reproducing their research work.

This paper describes a UAV framework for autonomous navigation under victim detection and location uncertainty in complex GNSS-denied scenarios. The framework details a system architecture for onboard execution of computer vision and decision-making methods in resource-constrained hardware, removing the dependency of the UAV on external ground control stations and communication systems, so it can interact with the environment by itself and accomplish the flight mission. The problem is mathematically formulated as a POMDP, which allows modelling uncertainty using probabilistic distributions. The POMDP model is implemented in software through the Toolkit for approximating and Adapting POMDP solutions In Real time (TAPIR) [39], which encapsulates the ABT algorithm for real-time decision making.

The framework is illustrated with an indoor SAR scenario to detect victims in office buildings. The UAV system was tested by defining three (3) case studies of situational awareness on the victim hypothetical location: (i) a single survey patch from the surveyed environment; (ii) two survey patches covering two areas of interest; (iii) a survey patch covering the entire flying area. The evaluations are separated into two groups: experiments designed to incorporate Flight Controller Units (FCU) and companion computers in Hardware in the Loop (HIL) simulations, and experiments with real flight tests. Experimental results show how the formulation of the problem as a POMDP optimises UAV behaviour by calculating robust path planning under unstable UAV motion response. More importantly, the results indicate the potential of the system to ensure rapid monitoring (for the identification and location of possible victims in office buildings) and personal safety by letting the UAV to explore dangerous environments without the intervention of the human operator.

This paper extends the published work by Sandino et al. [40] through the following primary contributions:

- A more detailed description of the entire UAV framework and system architecture rather than the POMDP problem formulation for autonomous UAV navigation in GNSS-denied environments.
- An improved observation model of target detection uncertainty, which introduces a summary statistic that measures detection frequency (to account for false positive detections).
- An improved cost function which contains more reward variables for better UAV behaviour (i.e., distance calculation between UAV and victim, and added memory capability for analysis of traversed path).
- Better onboard object detector performance by applying rotation transformations on input camera frames to detect victims at various visual perspectives.
- Validation of the proposed framework using real flight tests.

2. Background

This section describes the fundamentals of POMDP planning and Augmented Belief Trees (ABT), the online solver used in this work. A comprehensive review of POMDP and ABT can be found in the research works by Dutech and Scherrer [41] and Kurniawati and Yadav [26], respectively.

2.1. Partially Observable Markov Decision Processes

The main focus of autonomous UAV decision-making systems is to generate sequences of actions to avoid obstacles, explore unknown areas and detect objects of interest (i.e., victims). The information acquired about the surveyed environment and their targets is in most cases, however, inaccurate due to imperfections in the UAV sensor readings, occlusion from obstacles and challenging surveying conditions. These imperfections restrict the inference of the actual conditions of the environment (e.g., search extent, obstacles, wind disturbances) and victims (e.g., location, classification, quantity). A possible approach to model sequential decision-making processes when dealing with high levels of uncertainty is based on POMDPs [41].

A POMDP is defined by the tuple $\langle A, S, O, T, \mathcal{Z}, R, b_0, \gamma \rangle$ [42], where A is a finite set of UAV actions, S is a finite set of states, and O is a finite set of collected observations from the environment. Whenever the UAV takes an action $a \in A$ from a state $s \in S$, the UAV moves to a new state $s' \in S$ with probability $T(s, a, s') = \mathbb{P}(s' | s, a)$ and receives an observation $o \in O$ with probability $\mathcal{Z}(s', a, o) = \mathbb{P}[o | s', a]$. Each taken action is also valued with a reward or cost function R , defined as the expected reward after taking an action $a \in A$ from state $s \in S$.

Considering that the UAV is limited to obtain partial information of real system states through its collected observations, it generates a belief b , defined as a probability distribution over the system states S . A belief b can be defined as follows:

$$b(H) = \mathbb{P}[s^1 | H], \dots, \mathbb{P}[s^n | H], \quad (1)$$

$$H = a_0, o_1, R_1, \dots, a_{t-1}, o_t, R_t, \quad (2)$$

where H is the history of actions, observations and rewards that the UAV has experienced until time step t . The UAV always starts the planning with an initial belief b_0 , which is generated based on the initial conditions (and assumptions) of the problem (i.e., situational awareness). Given a belief b , a POMDP is solved once it finds a sequence of actions that maximises the discounted cumulative reward. The motion policy π of the UAV is represented by mapping belief states to actions $\pi : b \rightarrow A$. The optimal policy π^* is calculated as follows:

$$\pi^* := \arg \max_{\pi} \left(\mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R(S_t, \pi(b_t)) \mid b_0, \pi \right] \right), \quad (3)$$

where $\gamma \in [0, 1]$ is the discount factor, which determines how much immediate rewards are preferred over more distant rewards.

2.2. Augmented Belief Trees

Finding the exact solution of a POMDP is deemed to be a computationally intractable problem [43]. However, recent approaches have made substantial progress on creating algorithms that approximate the solution such as Partially Observable Monte Carlo Planning (POMCP) [25]. Nevertheless, most of the available online POMDP solvers recompute policies at every time step from scratch, wasting computational resources which can impact the performance of resource-constrained hardware devices, such as onboard computers in small UAVs. Therefore, this work uses the ABT solver [26], which contains methods to reuse previous computed policies and update the policy after detecting changes in the POMDP model. Compared to other online POMDP solvers, ABT allows declaring

continuous variables for states and actions (rather than discrete values) to calculate the approximated optimal policy.

ABT contains a method for planning and execution in real time with augmented belief trees. The process is divided into two parts: preprocessing (or offline policy estimation) and runtime (or online policy update). Once ABT is run to generate an offline policy from the POMDP model, the UAV executes a first action. Afterwards, the UAV collects an observation and the ABT updates the belief states based on the collected observation. Subsequently, ABT updates the policy online and executes the next action. The solver approximates the optimal motion policy by maintaining a set of multiple sampled episodes. Instead of providing explicit probability distributions for T and Z , ABT uses a generative model. A generative model is a black box simulator that outputs observations, rewards and next states once the UAV performs an action from a current state.

3. System Architecture

The proposed system architecture allows fully autonomous decision-making onboard small UAVs in unknown GNSS-denied environments. Following the POMDP terminology introduced in Section 2.1, Figure 1 illustrates the system modules of the UAV (also known as the agent in sequential decision-making theory), and the interaction between the UAV, the surveyed environment and the operator.

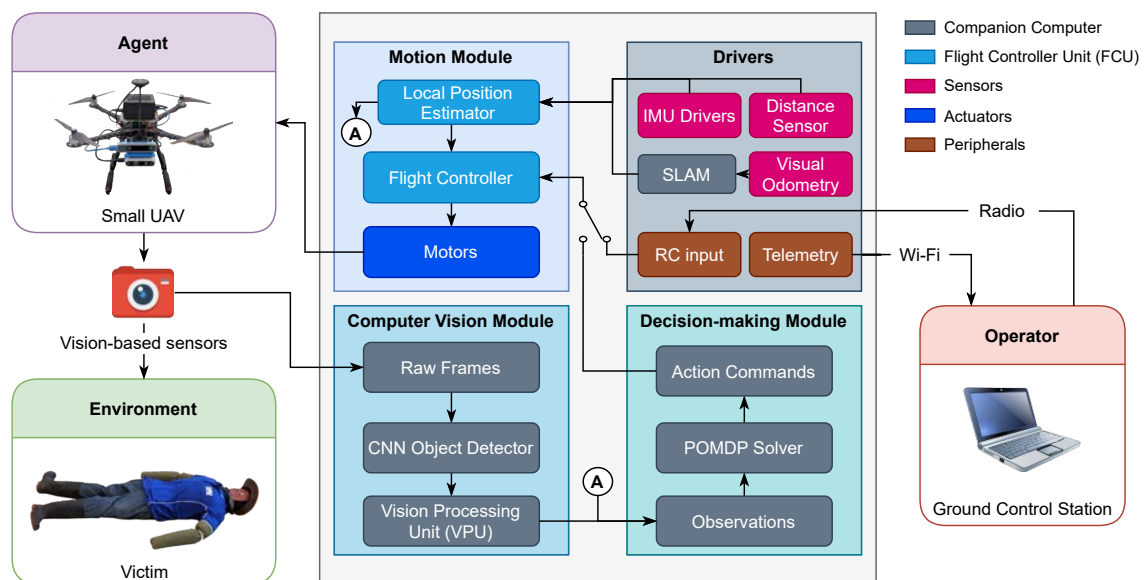


Figure 1. Proposed modular system architecture for an autonomous onboard navigation in Global Navigation Satellite System (GNSS)-denied environments in small Unmanned Aerial Vehicles (UAVs). It is composed of a computer vision module, which processes raw data from vision-based sensors, a decision-making module which sends low-level action commands to the flight controller, and a motion module that controls the dynamics of the UAV via a set of drivers mounted in the UAV frame.

The UAV contains a set of modules to distribute operations such as collecting and processing data from the environment (computer vision), evaluating the optimal sequence of actions to accomplish the flight mission (decision-making) and managing the speed of the actuators to control the dynamics (or motion) of the UAV. The computer vision and decision-making modules are run on a companion computer attached to the UAV frame, whereas the motion module is managed by the onboard FCU. The UAV also includes a set of drivers to assist the local position estimation in GNSS-denied environments and peripherals to establish communication to the operator. The operator receives real-time telemetry and decides whether to let the UAV interact with the environment or regain manual control using the remote control.

For every interaction cycle while operating the UAV in autonomous mode, the UAV starts capturing data from vision-based sensors in the form of image frames. Those frames are read by the computer vision module which processes these observations into percepts for the decision-making module. The current implementation of this module uses a Deep Convolutional Neural Network (D-CNN) to detect victims. A D-CNN is an artificial neural network designed for processing structured data arrays such as images [44]. The module contains dedicated hardware to meet the computational demand of running D-CNNs on resource-constrained hardware via Vision Processing Units (VPUs). Information produced by the computer vision module details whether a victim is detected in the frame, and if detected, the estimated location of the victim and a summary statistic about the confidence of the detection. The decision-making module reads this data as well as the estimated local position of the UAV (from the motion module) as observations. Then, the POMDP solver determines the corresponding motion command for the next iteration. Finally, these actions are passed to the flight controller which ultimately sends appropriate control signals to the actuators and manoeuvre the aircraft to the desired position.

The UAV requires a set of external sensors from the FCU to successfully operate in GNSS-denied environments. In this implementation, estimations of local UAV positioning are achieved using Light Detection and Ranging (LiDAR)-based distance and visual odometry sensors. It is common when working with visual odometry sensors to externally run (using a companion computer, for example) visual Simultaneous Localisation and Mapping (SLAM)-based algorithms to provide the FCU with relevant pose and twist data. Modern FCUs contain dedicated algorithms to read localisation data from multiple sensory systems (i.e., IMU, distance sensor and SLAM output) and estimate the local UAV position in real time.

4. Framework Implementation

The implemented hardware and software (i.e., framework) is designed to be as modular as possible based on the system architecture presented in Section 3. First, the paper presents UAV frame, drivers and payload, followed the software communication interface of the system, and software solutions to implement the computer vision and decision-making modules onboard the UAV companion computer.

4.1. UAV Frame and Drivers

The current implementation consists of but it is not restricted to the UAVs and sensors mentioned below. The UAV frame used is a Holybro S500 multi-rotor kit (Holybro, China), which offers a right balance between payload adaptability and size to navigate in cluttered indoor environments. The aircraft features a Pixhawk 4[®] autopilot (i.e., onboard FCU), 22.86 cm plastic propellers, 2212 KV920 brushless motors, and 433 MHz Telemetry Radio. The aircraft length \times width \times height dimensions are of 38.3 cm \times 38.5 cm \times 24.0 cm, with a total load payload capacity of 0.4 kg. The UAV uses a four (4) cell 5000 mAh LiPo battery which provides an approximate flight autonomy of 10 min payload-free and eight minutes with the sensor payload and companion computer mounted in the frame. An illustration of the UAV frame with its payload, companion computer and drivers is shown in Figure 2.

The companion computer which runs the computer vision and decision-making modules is an UP² (AAEON Technology Inc., New Taipei City, Taiwan). The computer features a 64-bit quad-core Intel[®] Pentium[®] N4200 processor at 1.1 GHz, 8 GB DDR3 RAM, 64 GB eMMC SSD, four FL110 USB 3.0 connectors, two Ethernet controllers, two High-Speed UART controllers, an Intel[®] Dual Band Wireless-AC 3165 and one mPCIe connector. The UP² is selected here against similar computer boards owing to its competitive price tag for its provided features, peripherals and the familiar 64-bit CPU architecture.

The UAV requires several sensor drivers to estimate its local position in the absence of GNSS. For these experiments, the list of sensors is composed by the embedded Pixhawk 4[®] IMU, a TFMMini Plus range sensor (Benewake, Beijing, China) pointing downwards which provides the UAV altitude,

and an Intel[®] Realsense[™] T265 tracking camera (Intel Corp., CA, US) pointing to the front from the UAV frame. The T265 sensor uses a closed source SLAM software implementation for local position and motion estimation. Configuring the camera to the front improves the reliability of the sensor readings by capturing and detecting more objects (e.g., obstacles, walls, floor and victims) than by pointing the camera to the ground. Collected observations from the environment are performed with a HBV-1615 Red Green Blue (RGB) camera, mounted in a downward-looking configuration from the UAV frame (Figure 2b). The camera features a resolution of 640×480 pixels, focal length of 2.484 mm, sensor width of 1.968 mm and sensor height of 1.488 mm. It is worth mentioning that other multi-rotor UAVs, companion computers and drivers with similar characteristics can also be utilised for the proposed system architecture depicted in Figure 1.



Figure 2. Proposed UAV frame with mounted drivers, companion computer and payload. (a) Front view of the UAV displaying: (1) Holybro S500 frame; (2) Pixhawk 4[®] flight controller; (3) UP² companion computer; and (4) Intel[®] Realsense[™] T265 tracking camera. (b) Lateral view of the UAV displaying: (5) 433 MHz telemetry radio; (6) HBV-1615 RGB camera; and (7) TFMini Plus range sensor.

4.2. Operating Systems and Middleware

The system modules implemented in the companion computer are developed for 64-bit Linux Operating Systems (OS) and run in Ubuntu Server 18.04. Communication between the vision-based sensors, and computer vision and decision-making modules is achieved using the Robot Operating System (ROS) Melodic [45] middleware. The flight controller runs under NuttX (a real-time OS) and the PX4 flight control software [46]. The PX4 architecture consists of: 1) a flight stack layer, which details a pipeline of flight controllers for multi-rotors, fixed-wing and vertical take-off and landing (VTOL) UAVs, altitude and position estimators, and; 2) A middleware layer, which contains the device drivers for multiple UAV sensors, communication interfaces, and a simulation layer to enable Hardware in the Loop (HIL) capabilities of the FCU.

Communication between the decision-making module (from the companion computer) and the motion module (from the FCU) is done using MAVROS via a High-Speed UART interface. MAVROS is a ROS wrapper of the Micro Air Vehicle Link (MAVLink) protocol, an industry standard for UAV communication [47]. Telemetry to the ground control station was performed using QGroundControl via Wi-Fi and the Holybro 433 MHz Telemetry Radio (Holybro, China).

4.3. Computer Vision Module

This module consists of a deep learning object detector processing raw frames from the HVB-1615 RGB camera. Taking into account the performance limitations of running deep learning models in resource-constrained hardware, a Vision Processing Unit (VPU) is installed to the companion computer. The use of a VPU boosts the computations that allow inference in deep learning models by optimising convolutional operations in its microprocessor. In this implementation, the selected VPU is

an Intel® Movidius™ Myriad™ X, which is connected to the companion computer via the mPCIe slot. The detection module is programmed in Python and uses the OpenVINO library that allows a direct interface between various deep learning frameworks and the VPU. OpenVINO supports TensorFlow, Caffe, PyTorch, among other deep learning frameworks. Standard image processing methods are also covered by OpenVINO through optimised versions of the OpenCV and OpenVX libraries.

The used deep learning model architecture to detect persons is an off-the-shelf Google MobileNet Single-Shot Detector (SSD) [48]. This model is deployed in Caffe [49] and tuned with pre-trained weights from the PASCAL VOC2012 dataset [50], scoring a mean average precision of 72.7%. The dataset covers up to 21 class objects (including persons). However, only positive detections for the class person are evaluated. Acquired camera frames are fitted into the input layer of the neural network (i.e., MobileNet SSD model) by shirking the frames into dimensions of 300×300 pixels. Once inference is performed onto the fit model, positive detections of persons with a score confidence (from the output layer of the neural network) greater than 60% are depicted in the processed frame and shown to the human operator for telemetry purposes, as displayed in Figure 3.



Figure 3. Victim detection trial using the proposed Google MobileNet Single-Shot Detector (SSD) architecture from Chuanqi [48]. Detections with an output score confidence greater than 60% are displayed in the processed frame. (a) Top view of the UAV flying above an adult mannequin. (b) Victim detected with a confidence score of 78.42% after processing raw frames from the HBV-1615 RGB camera.

Taking into account the nature of PASCAL VOC dataset the object detector was trained for the class person, the frequency and detection scores are significantly higher when the UAV is aligned with the mannequin (as shown in Figure 3b) than with other visual representations. Consequently, the object detector is unable to detect the mannequin if spatial transformations in the x or y axes are applied. Similarly, there are slight chances to detect the mannequin if its lower body is occluded by other objects. The optimal distance between the UAV and the mannequin to maximise the detection scores ranges between 1 m and 10 m. In order to address these limitations on the detections, image rotation transformations are applied on software for each input frame. A total of six transformations (i.e., image rotations every 60°) are processed for every read camera frame, achieving, thus, an approximate processing speed of 2.9 Frames per Second (FPS).

4.4. Decision-Making Module

The decision-making module contains algorithms that translate information from the environment (i.e., observations) into action commands. In this implementation, the decision-making module contains the POMDP, in which the navigation problem is required to be formulated. The decision-making module uses ABT—an online POMDP solver [26]—implemented on software using TAPIR [39]. TAPIR is developed in C++ and requires the tuning of several hyper-parameters to obtain the best possible approximation of the POMDP solution (i.e., optimal motion policy). Details on assigned hyper-parameter values for TAPIR can be found in Section 5.2.3.

This paper proposes an approach for a UAV to find a victim in cluttered indoor environments. An example scenario, which is depicted in Figure 4, consists of a limited flying area, several obstacles randomly placed, with weak or absent GNSS signal, and a static victim located inside the area to be surveyed.



Figure 4. Example of a cluttered indoor environment for UAV Search and Rescue mission. The extent of the free volumetric space area is restricted by randomly placed obstacles. The victim is always lying on the ground and its location is static.

The estimation of the optimal motion policy allows the UAV to perform efficient obstacle avoidance, victim detection and path planning for various uncertainty levels in the location of the victim. The problem formulation is partially defined based on the following assumptions:

- The UAV pose and motion are estimated by a SLAM-based sensor (i.e., visual odometry) embedded on the UAV frame.
- Observations come from real-time streaming of processed camera frames and the estimated local UAV position from the FCU.
- Flying modes such as take-off and landing are also delegated to the FCU and automatically triggered by the UAV, or the operator if they want to regain control on the UAV motion.
- The task starts after the UAV gets close enough to a chosen starting point.
- The task finishes once the victim is detected with high detection frequency (e.g., exceeding a minimum threshold detection value), or if the UAV runs out of power resources (or timeout) to keep flying before a detection is made.

The text below describes the problem formulation for the elements of the POMDP tuple, which consists of the set of possible taken actions A by the UAV; the system states S ; the motion model of the system after an action $a \in A$ is executed by the UAV; the system rewards and cost function R ; the collected observations O from the environment; the observation model; and the initial belief b_0 . The problem formulation is presented as generic as possible in this section. Technical details on the assigned values in the experiments are described in Section 5.

4.4.1. Actions (A)

In the current implementation, the UAV interacts with the environment by applying an action $a \in A$. As shown in Table 1, seven actions have been selected in this paper. However, more actions can be added as per problem requirements such as setting UAV yaw orientation and camera gimbal angle commands.

Table 1. Set of chosen actions for the problem formulation. Each action is a position command where δ is the magnitude of change of position coordinates x_u , y_u and z_u from time step k to time step $k + 1$. Position values are referenced to the world coordinate frame.

$\mathbf{a}(\mathbf{k}) \in \mathbf{A}$	$\mathbf{x}_u(\mathbf{k} + 1)$	$\mathbf{y}_u(\mathbf{k} + 1)$	$\mathbf{z}_u(\mathbf{k} + 1)$
Forward	$x_u(k) + \delta_x$	$y_u(k)$	$z_u(k)$
Backward	$x_u(k) - \delta_x$	$y_u(k)$	$z_u(k)$
Left	$x_u(k)$	$y_u(k) + \delta_y$	$z_u(k)$
Right	$x_u(k)$	$y_u(k) - \delta_y$	$z_u(k)$
Up	$x_u(k)$	$y_u(k)$	$z_u(k) + \delta_z$
Down	$x_u(k)$	$y_u(k)$	$z_u(k) - \delta_z$
Hover	$x_u(k)$	$y_u(k)$	$z_u(k)$

The above-mentioned actions are position commands defined in the world coordinate frame, where δ is the magnitude of change of position coordinates x_u , y_u and z_u from time step k to time step $k + 1$. Assigning various values for δ allows flexibility in the speed of the UAV for big and small flying areas. Other standard UAV actions such as take-off and landing are off-the-shelf commands managed by the onboard FCU and triggered by the system before and after executing the POMDP solver, respectively. A description of the initial conditions of the UAV is covered in detail in Section 5.

4.4.2. States (S)

For this implementation the system states consist of the UAV and victim states. The state of the UAV is defined by the position of the UAV $p_u(x_u, y_u, z_u)$ in the world coordinate frame; the UAV instantaneous velocity $v_u = \dot{p}_u$; the flag f_{crash} that defines whether the UAV has collided with an obstacle; and the flag f_{roi} which specifies whether the UAV is navigating out of the limits of the region to be surveyed. The victim state is defined by the position of the victim $p_v(x_v, y_v, z_v)$ in the world coordinate frame; the flag f_{dct} that determines whether a victim has been detected by the UAV; and the flag f_{conf} that confirms the detection state of the victim, which is defined as:

$$f_{\text{conf}} = \begin{cases} \text{true} & \text{if } \zeta \geq \text{threshold} \\ \text{false} & \text{otherwise} \end{cases}, \quad (4)$$

where $\zeta \in [0, 1]$ is the victim's detection confidence between time steps. An expanded explanation of ζ and its usage is covered in Section 4.4.5. In the current formulation, f_{crash} , f_{roi} and f_{conf} are considered terminal states (or stopping conditions). Other states such as states for a second or more victims can also be added to the framework.

4.4.3. Motion Model

The motion model for a multi-rotor UAV is defined as:

$$p_u(k + 1) = p_u(k) + \mathbf{X}_u(\mathbf{k})\Delta p_u(k) \quad (5)$$

which can be expanded as:

$$\begin{bmatrix} x_u(k + 1) \\ y_u(k + 1) \\ z_u(k + 1) \end{bmatrix} = \begin{bmatrix} x_u(k) \\ y_u(k) \\ z_u(k) \end{bmatrix} + \begin{bmatrix} \cos(\varphi_u(k)) & -\sin(\varphi_u(k)) & 0 \\ \sin(\varphi_u(k)) & \cos(\varphi_u(k)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_u(k) \\ \Delta y_u(k) \\ \Delta z_u(k) \end{bmatrix}, \quad (6)$$

where $p_u(k)$ is the UAV's position at time step k ; $\mathbf{X}_u(\mathbf{k})$ is a simplified multi-rotor rotation matrix after assuming changes in the Euler angle values $\Delta\psi = 0^\circ$, $\Delta\theta = 0^\circ$ and $\Delta\phi = 0^\circ$ [51]; $\Delta p_u(k)$ is the change in the UAV's position from time step k to time step $k + 1$; and $\varphi_u(k)$ represents pose estimation errors in the Euler yaw angle of the UAV. This variable is modelled as a normal distribution with

a mean of 0° and standard deviation of 3.0° . An approximation of the dynamic model of the UAV through changes in position ($\Delta p_u(k)$) was conducted empirically using the UAV frame and a system identification process. An expanded description for calculation of $\Delta p_u(k)$ can be found in Section 5.

4.4.4. Rewards and Cost Function (R)

The cost function R is defined as follows:

$$R = r_{\text{action}} + r_{\text{crash}} + r_{\text{out}} + r_{\text{dtc}} + r_{\zeta} + r_{\text{fov}}, \quad (7)$$

where r_{action} is the negative reward (or penalty) per action taken to encourage the UAV to find the victim in the less number of possible action sequences; r_{crash} is the cost if the UAV crashes itself with an obstacle; r_{out} is the cost if the UAV flies beyond the survey area limits; r_{dtc} is the reward if a victim is detected (regardless of the confidence level), defined as follows:

$$r_{\text{dtc}} = \begin{cases} \rho & \text{if victim is detected} \\ -(\rho + r_{\zeta}) \frac{d_u}{2 \cdot l_a} & \text{otherwise} \end{cases}, \quad (8)$$

where ρ is the constant reward value assigned to r_{dtc} ; d_u is the Manhattan distance between the UAV and victim; r_{ζ} is the reward if the victim is detected with a high confidence level, defined by the threshold ζ (as mentioned above in Equation (4)); and l_a represents the longest length of the search area. Manhattan distance was chosen over Euclidean distance for d_u based on better UAV traversed paths using the former in preliminary simulations. Adding d_u to the cost function aims to get the UAV closer to the believed location of the victim and acquire camera frames with clearer visual representations of them. Nevertheless, d_u might generate ambiguity and sub-optimal behaviour in the UAV if it is surrounded by equidistant victim belief particles. In order to add memory about a path previously traversed by the UAV, R includes r_{fov} , which is the cost for any taken action that places the UAV in a region that was previously explored. An illustration of the concept is shown in Figure 5.

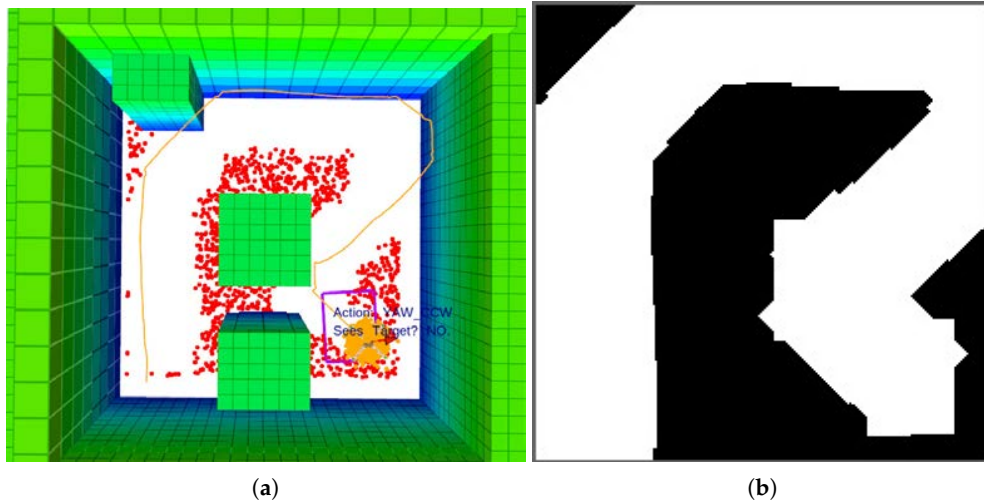


Figure 5. Example of a recorded trajectory. (a) Traversed path by the UAV. The figure is composed by: The path of the UAV (orange splines); the environment obstacles (green blocks); the UAV position belief (orange point-cloud); the possible victim location coordinates (red point-cloud); and the camera's Field of View (FOV) (purple rectangle). (b) Traced footprint using the camera's Field of View (FOV). Future actions that place the UAV inside the white areas trigger r_{fov} .

4.4.5. Observations (O)

The system observations consist of the available information about the state of the environment and the UAV from its sensors. As previously illustrated in Figure 1, certain observations require a pre-processing stage such as the detection and localisation of the victim from vision-based camera frames. The current observations O are defined as:

$$O = (o_{p_u}, o_{\text{obs}}, o_{\text{dte}}, o_{p_v}, o_{\zeta}), \quad (9)$$

where o_{p_u} is the position of the UAV in the world coordinate frame, which is obtained from the local position estimator of the motion module; o_{obs} is the flag that defines whether there is an obstacle in front of the UAV, which is obtained by reading the location of the UAV inside an occupancy map object (described in Section 4.4.8); o_{dte} is the flag that determines whether a victim has been detected by the computer vision module; and o_{p_v} provides the location estimation of the victim, defined as:

$$o_{p_v} = \begin{cases} (x_v, y_v, z_v) & \text{if } o_{\text{dte}} = \text{true} \\ \text{null} & \text{otherwise} \end{cases}; \quad (10)$$

observation o_{ζ} is a summary statistic that measures the frequency of victim detections between the last and current observation calls (and referred as the detection confidence), defined as:

$$o_{\zeta} = \frac{\sum \text{victim detections}}{\sum \text{processed frames}}. \quad (11)$$

Other observations such as the orientation of the victim or similar object detection outputs from two or more cameras are not implemented but can also be considered in the formulation.

4.4.6. Observation Model

Considering that ABT uses a generative model that outputs an observation $o \in O$, a reward R and a next state $s' \in S$ based on a taken action $a \in A$ from a current state $s \in S$, probabilistic transition functions T and \mathcal{Z} are not required to be explicitly declared. Therefore, the generative model requires modelling a potential observation o given s' and a . The observation model is composed by the local position estimation of the UAV o_{p_u} in the world coordinate frame, the local position of the victim o_{p_v} if it is detected by the vision-based sensors and the detection confidence o_{ζ} . Victim detection depends on the footprint extent of the camera's Field of View (FOV), which is defined by the sensor properties of the camera and o_{p_u} . The vertical and horizontal FOV angles are defined as follows:

$$\text{FOV}_V = 2 \tan^{-1} \left(\frac{w}{2f} \right), \quad (12)$$

$$\text{FOV}_H = 2 \tan^{-1} \left(\frac{h}{2f} \right), \quad (13)$$

where w is the sensor width; h is the sensor height; and f is the focal length of the camera. The extent of the observed FOV area (or footprint) is calculated as:

$$l_{\text{top, bottom}} = p_u(z) \cdot \tan(\alpha \pm 0.5 \cdot \text{FOV}_H), \quad (14)$$

$$l_{\text{left, right}} = p_u(z) \cdot \tan(\alpha \pm 0.5 \cdot \text{FOV}_V), \quad (15)$$

where l is the footprint extent of the camera frame at its top, right, bottom and left limits, and α is the pointing angle by the camera gimbal from the vertical z axis, as shown in Figure 6.

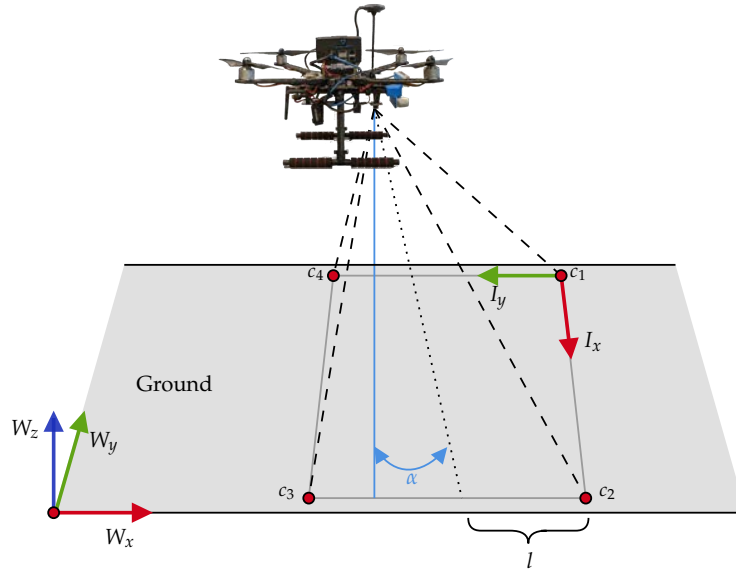


Figure 6. Two-dimensional (2D) projection of vision-based sensors pointing to the ground, where W is the world coordinate frame, I is the image coordinate frame, α is the camera's gimbal angle from the vertical, c_* are the rectangular corners from the camera's Field of View (FOV), and l is the footprint extent of the FOV.

The two-dimensional (2D) projection corner coordinates of the camera's FOV are defined as:

$$c_1 = (l_{\text{top}}, l_{\text{left}}), \quad (16)$$

$$c_2 = (l_{\text{top}}, l_{\text{right}}), \quad (17)$$

$$c_3 = (l_{\text{bottom}}, l_{\text{right}}), \quad (18)$$

$$c_4 = (l_{\text{bottom}}, l_{\text{left}}). \quad (19)$$

Following Equation (5), a transformation matrix is also applied to link the FOV corners to the UAV reference frame:

$$\begin{bmatrix} c'(x) \\ c'(y) \end{bmatrix} = \begin{bmatrix} p_u(x) \\ p_u(y) \end{bmatrix} + \begin{bmatrix} \cos(\varphi_u) & -\sin(\varphi_u) \\ \sin(\varphi_u) & \cos(\varphi_u) \end{bmatrix} \begin{bmatrix} c(x) \\ c(y) \end{bmatrix} \quad (20)$$

where φ_u is the pose estimation error in the Euler yaw angle of the UAV (mentioned in Equation (5)). If the projected 2D point coordinate of the victim is located within the corner c points from the formed rectangular footprint, the victim is assumed to be visualised in the camera's FOV. As defined in Equation (21), an angle θ is calculated as the sum of angles between the victim's position and each pair of points that comprise the FOV corners [52],

$$\theta = \sum_{i=1}^4 \left\{ \tan^{-1} \left[\frac{c'_{i+1}(y) - p_v(y)}{c'_{i+1}(x) - p_v(x)} \right] - \tan^{-1} \left[\frac{c'_i(y) - p_v(y)}{c'_i(x) - p_v(x)} \right] \right\}. \quad (21)$$

If $\theta = 2\pi$, the coordinate point of the victim is inside the camera's FOV. Nevertheless, this calculation assumes perfect detection outputs from any vision-based model implemented in the computer vision module. Uncertainty from computer vision models (including deep learning detectors) come from factors such as noise from the camera frames, poor illumination conditions, low image resolution, occlusion from obstacles and sub-optimal camera settings. Even though it is possible to allocate extra resources to improve the performance of these object detection models, this paper

presents an approach that covers detection uncertainty. Some factors that cause uncertainty such as the display of false positives are simulated here by incorporating the object detection confidence ζ and evaluating such confidence with thresholds ζ_{thres} in the problem formulation. In this implementation ζ is modelled using a linear regressor, defined in Equation (22):

$$\zeta = \frac{1 - \zeta_{\min}}{\max_z - \min_z - g} (d_u - \min_z - g) + \zeta_{\min}, \quad (22)$$

where ζ_{\min} is the minimum detection confidence threshold; \min_z and \max_z are the minimum and maximum allowed flying altitudes respectively; g is the distance gap applied to \min_z ; and d_u is the Manhattan distance between the UAV and the victim.

4.4.7. Initial Belief (b_0)

Formulating the problem as a POMDP allows modelling uncertainty and partial observability with probabilistic data distributions. The proposed system contains two sets of belief states: The position of the UAV, and the position of the victim. The position of the UAV is defined as a normal probability distribution with mean μ_{p_u} and standard deviation σ_{p_u} . As previously shown in Figure 5a, the position of the victim can be defined as a uniform probability distribution or as a set of one or more normal distributions placed across the flying area. This capability allows the rescue personnel and UAV operator to freely define possible areas where a victim could be located, following the concept of situational awareness in SAR operations. Details on the specific configurations used in the experiments can be found in Section 5.2.2.

4.4.8. Obstacle Avoidance

The decision-making module relies on the concept of occupancy maps for obstacle avoidance. Occupancy maps are represented in three-dimensional (3D) occupancy grids whose cells contain binary values for the specific volumetric representation of space such as free, occupied or unknown. The occupancy map for this task was generated using the Octomap library [53], which allow the creation of occupancy maps using 3D point clouds, data that is commonly acquired from depth cameras and LiDAR sensors. In this work, the octomap of the environment was generated manually and read by the decision-making module prior to any flight campaigns (in simulation and hardware).

5. Experiments

The presented system is tested for a victim finding mission in a cluttered indoor environment using HIL simulations and real flight tests. The text below describes the environment setup and assigned values to the formulated POMDP problem presented in Section 4.4.

5.1. Environment Setup

The search area has length \times width \times height dimensions of 6 m \times 6 m \times 3 m. As shown in Figure 7, the surveyed area contains several obstacles in the shape of columns, the victim is lying on the ground and located at the opposite end from the take-off position of the UAV. The victim and column obstacles are static and no disturbances such as wind or variable light conditions are applied to the setup. Due to the limited flying space available in the testing facility for hardware tests, it was not possible to evaluate the system in an environment with bigger dimensions that suit more realistic SAR operations. However, the focus and contribution of this paper relies on: introducing a framework for autonomous UAV operations in GNSS-denied environments under partial observability; illustrating a system architecture that incorporates and executes computer intensive deep learning models (for realistic object detection) and online POMDP solvers onboard resource-constrained hardware; and presenting a proof of concept of the system robustness in SAR scenarios, with the potential of operating the system in more challenging conditions.

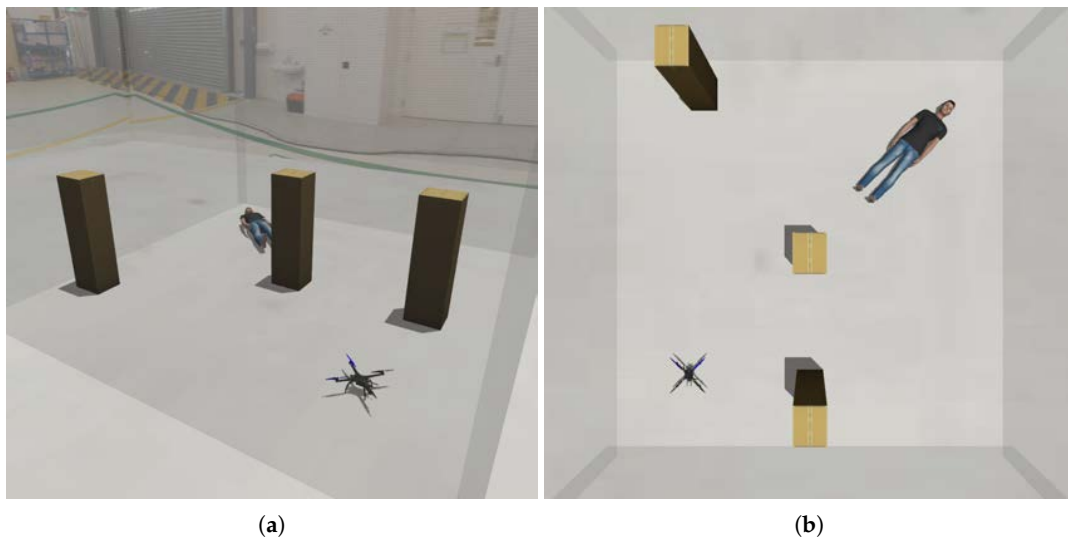


Figure 7. Environment setup for autonomous UAV victim finding in HIL simulations and real-flight tests. The victim is static and is always located at the opposite end from the take-off position of the UAV. (a) Isometric view of the environment simulated in Gazebo. (b) Top view of the environment simulated in Gazebo.

The example environment is validated with HIL simulations and hardware experiments. Simulation experiments are executed using a desktop workstation, replicating the environment setup using the Gazebo ROS Simulator. The desktop workstation features a 64-bit 12-core Intel® Core® i7-8700 CPU at 3.2 GHz, a 32 GB DDR4 RAM, a 512 GB eMMC Solid State Drive, a 6 GB NVIDIA GeForce GTX 1060, six (6) USB 3.0 ports and an Ethernet controller.

Hardware experiments are conducted at the Queensland University of Technology (QUT) Da Vinci Precinct (DVP) hangar area, 24/22 Boronia Rd, Brisbane Airport, QLD 4008, Australia. Data collection campaign occurs in four (4) opportunities from the 25 June 2020 to the 2 July 2020 between 11:00 a.m. and 3:00 p.m. For safety reasons, an adult mannequin is used as the victim to be detected, as shown in Figure 8. Illumination conditions are controlled by exposing the setup with a constant light intensity from fluorescent light bulbs. No external wind disturbances are applied during the data collection process.



Figure 8. Environment setup for hardware experiments. For safety reasons, an adult mannequin is used as the victim to be detected and column obstacles are replaced with carpet tiles. (a) UAV navigating inside a netted area at the Queensland University of Technology (QUT) Da Vinci Precinct (DVP) hangar. (b) Top view of the setup with the mannequin displayed at the bottom.

In the setup neither a depth camera nor LiDAR sensor is included in this framework implementation. The system is therefore limited to operate using a fixed occupancy map of the environment. Despite this limitation, the proposed framework allows extending the system capabilities by incorporating any of the above-mentioned sensors to the UAV frame and updating the map mid-flight using the Octomap library flawlessly. Taking into account that local UAV position estimations from the visual odometry sensor (i.e., T265 tracking camera) are not corrected from any ground truth data source, the environment column-shaped obstacles (Figures 7 and 8a) are replaced by carpet tiles to ensure the integrity of the aircraft in real flight tests (Figure 8b).

5.2. POMDP Problem Formulation

The text below describes the assigned values for variables presented in the POMDP formulation (Section 4.4). They consist of the approximated response of the controlled UAV dynamic system, the case studies of situational awareness (or initial belief) on the victim's position, and TAPIR hyper-parameter values.

5.2.1. Controlled UAV System Response

In this research, the position changes of the UAV ($\Delta p_u(k)$) are modelled by identifying the transfer function of the entire controlled UAV system, composed by the controller, the UAV motors, sensors and feedback loop. Consider y as the independent position response of the UAV for the x , y and z Cartesian frame. A step response of the aircraft $y(t)$ is measured after triggering a constant position setpoint $r(t) = 0.5$ m. For each coordinate axis, incremental and decremental step responses are recorded for five seconds between each change using a VICON motion capture system (Vicon, Oxford, UK) as ground truth. The recorded data is processed using the MATLAB[®] System Identification Toolbox[™], which estimates the transfer function of the UAV in the frequency domain (s). The transfer functions of the UAV for x , y and z in the frequency domain are defined as follows:

$$F_x(s) = \frac{0.204s + 1.136}{s^2 + 1.253s + 1.134} \quad (23)$$

$$F_y(s) = \frac{0.2875s + 0.9085}{s^2 + 0.9825s + 0.9227} \quad (24)$$

$$F_z(s) = \frac{0.8068s^3 + 0.7306s^2 + 1.041s + 0.1368}{s^4 + 1.561s^3 + 1.653s^2 + 1.175s + 0.1367} \quad (25)$$

Afterwards, the transfer functions were discretized using the Tustin approximation method, defined in Equation (26):

$$s \approx \frac{2(z-1)}{T_s(z+1)} \quad (26)$$

where T_s (i.e., $T_s = 0.1$ s) is the sampling period. The UAV motion $\Delta p_u(k)$ is calculated by obtaining the difference equation of the discretized system $F(z)$ after applying the inverse Z-transform, as defined from Equation (27) to Equation (30):

$$F(z) = \frac{Y(z)}{R(z)} \quad (27)$$

$$y(k) = Z^{-1}F(z)Z^{-1}R(z), \quad (28)$$

$$y(k) = \sum_{i=0}^N a_i r(k-i) - \sum_{i=1}^N b_i y(k-i), \quad (29)$$

$$\Delta p_u(k) = y(T_s^{-1}) - y(0), \quad (30)$$

where N is the order of the transfer function $F(s)$; $r(k-i)$ and $y(k-i)$ represent previous setpoint and response values respectively; and a_i and b_i are the numerical constants for each $r(k-i)$ and $y(k-i)$ function variables.

5.2.2. Uncertainty and Initial Belief (b_0)

As shown in Figure 9, the possible location of the victim was evaluated following three case studies, which are inspired on available information (situational awareness) from a SAR perspective: (1) single cluster of position points following a normal probability distribution with mean $\mu_{1_{p_v}}$ and standard deviation σ_{p_v} (Figure 9a); (2) two position clusters defined as normal probability distributions with the second cluster declared with mean $\mu_{2_{p_v}}$ and standard deviation σ_{p_v} (Figure 9b); (3) a uniform probability distribution assuming that there is no knowledge of where the victim might be located in the surveyed area (Figure 9c). It is worth mentioning that the physical mannequin was always located at position coordinates $p_v = (1.5, -1.4, 0.0)$ and orientation $\psi_v = -45^\circ$ from the world coordinate frame.

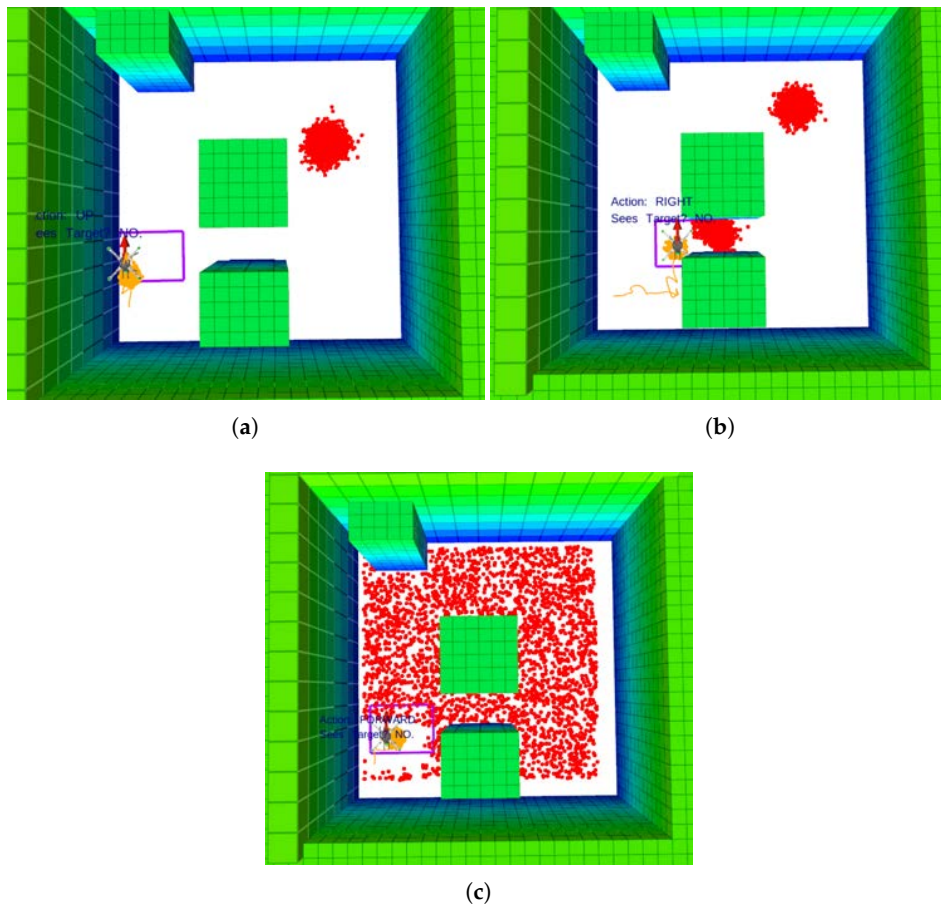


Figure 9. Tested case studies of search and rescue situational awareness regarding the victim location. Orange splines represent the path of the UAV; green blocks are the environment obstacles; the orange point-cloud is the UAV position belief; the red point-cloud is the victim position belief; and the purple rectangle the camera's Field of View (FOV). (a) Victim position points distributed in a single cluster. (b) Victim position points distributed in two clusters. (c) Cluster of victim position points uniformly distributed along the flying area.

5.2.3. TAPIR Hyper-Parameters

TAPIR requires tuning several hyper-parameters to obtain the best possible approximation of the POMDP solution (i.e., optimal motion policy). For these experiments, the offline policy timeout is set

to five (5) seconds, the maximum belief tree depth is of 100 nodes, time steps are of one (1) second, and the discount factor $\gamma = 0.99$. The UAV is conditioned to find the victim for a maximum of 480 iterations (equivalent to approx. eight minutes of flight time). The minimum and maximum flying altitudes of the UAV are of 1.0 m and 1.8 m respectively. Specific values for the POMDP variables defined in Section 2.1 are shown in Table 2,

Table 2. Hyperparameter values for the Partially Observable Markov Decision Process (POMDP) formulated problem. The initial belief position and orientation values were defined in reference to the world coordinate frame.

Category	Variable	Value	Category	Variable	Value
Actions	δ	0.25	States	ζ_{thres}	0.7
	r_{action}	-2.5		μ_{p_u}	(-1.8, 1.8, 1.5) m
	r_{crash}	-25		σ_{p_u}	1.0 m
Rewards	r_{out}	-10	Initial belief (b_0)	$\mu 1_{p_v}$	(1.5, -1.4, 0.0) m
	ρ	25		$\mu 2_{p_v}$	(-0.9, 0.2, 0.0) m
	r_{ζ}	75		σ_{p_v}	1.5 m
	r_{fov}	-5		ψ_v	-45°
				Observations	g

where δ is the magnitude of change of UAV position coordinates between time steps (i.e., $\Delta t(k) = 1$ s, $\therefore p_u \approx 0.25$ m/s) for the x , y and z Cartesian axes; the set of r_* variables constitute the system rewards defined in Section 4.4.4; ζ_{thres} is the minimum victim detection confidence that should be achieved by the UAV; the set of variables defined in the initial belief defined in Section 5.2.2; and g is the distance gap applied to the minimum UAV altitude defined in Section 4.4.5; The assigned values for the system rewards are found after performing grid-search into the ABT solver.

6. Results

A set of success metrics were evaluated for each one of the three situational awareness case studies regarding the covered area(s) where the victim was believed to be located: victim position points distributed in single, dual and uniform clusters (as discussed in Section 5.2.2). The metrics consisted on the victim confirmation rate (i.e., $f_{\text{conf}} = \text{true}$), the victim miss rate (i.e., $f_{\text{dte}} = \text{false}$), the UAV collision rate (i.e., $f_{\text{crash}} = \text{true}$), the UAV navigation rate flying beyond the area limits (i.e., $f_{\text{roi}} = \text{true}$), the occurrences where the aircraft followed a sub-optimal path, and the timeout rate (i.e., $k > 480$ steps, or $t \geq 480$ s). A summary of the collected metrics is shown in Table 3:

Table 3. Performance metrics for Hardware in the Loop (HIL) simulations (S) and real flight tests (FT), where \bar{x}_w is the weighted average of the measured variables.

Belief Cluster	Iterations	Detections (%)	Misses (%)	Sub-Optimal Path (%)	Timeout (%)
Single (S)	50	100.0	0.0	0.0	0.0
Single (FT)	7	85.7	14.3	0.0	0.0
Dual (S)	44	100.0	0.0	0.0	0.0
Dual (FT)	7	71.4	14.3	14.3	0.0
Uniform (S)	50	92.0	0.0	6.0	2.0
Uniform (FT)	9	88.9	0.0	0.0	11.1
$\bar{x}_w(\text{S})$	144	97.2	0.0	2.1	0.7
$\bar{x}_w(\text{FT})$	23	82.6	8.7	4.4	4.3

Where “Detections” measures the instances where the victim was detected with $\zeta \geq 0.7$. Otherwise, failed instances were either classified as “Misses” if $\zeta < 0.7$, “Sub-optimal Path” if the UAV got stuck in a patch that does not cover the victim’s location, or “Timeout” if the UAV consumed all the flying time ($k > 480$) without detecting the victim.

Overall, the presented system achieved a victim confirmation rate for all the cluster belief configurations of 97.2% in simulation and 82.6% in flight tests. For experiments with single clusters, a slight difference in the detection confidence was found between simulation and flight tests. An increase on the setup complexity by the victim silhouette, scene and camera conditions was attributed to lower victim detection confidence values during real flight tests. Namely, the imperfect anthropomorphic properties of the mannequin compared to its simulation counterpart and lower image quality from the RGB camera, represented a cost in the performance of the computer vision module to detect a person. A similar finding was discovered in tests with dual but not with uniform declared clusters. This behaviour could be attributed to the bigger search space required to find the victim using uniform clusters than with the former. In addition, the UAV was unable to detect a victim in 14.3% of the flight tests with dual cluster declarations because of a sub-optimal UAV trajectory. This behaviour was also present but less frequent in 6.0% of HIL simulation experiments under uniform cluster declaration. A few timeout stopping conditions were also triggered in missions with uniform clusters because the UAV kept navigating in unexplored areas after flying above and not detecting the mannequin. Lastly, none of the experiments triggered terminal states caused by collisions or UAV trajectories violating the flying area limits.

The most frequent trajectories generated by the UAV in experiments with single clusters are displayed in Figure 10. A set of arrows drawn on top of the UAV path represent the actions taken at every time step to clarify the influence of the decision-making module over the behaviour and stability of the aircraft during the missions.

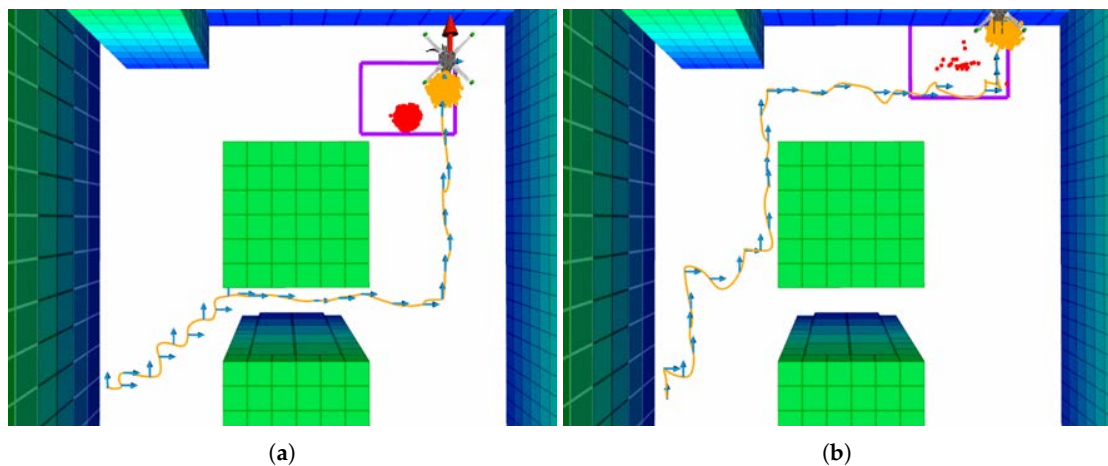


Figure 10. Flight mission examples of most frequent types of UAV trajectories under a single belief cluster. Orange splines represent the path of the UAV; the blue arrows are action commands per time step; green blocks are the environment obstacles; the orange point-cloud is the UAV position belief; the red point-cloud is the victim position belief; and the purple rectangle the camera's Field of View (FOV). (a) UAV motion policy which crosses two of the column-shaped obstacles. (b) UAV motion policy which avoids crossing two of the column-shaped obstacles.

Simulation and flight tests with dual clusters evidenced just one type of generated trajectory as opposed to the first case study, as shown in Figure 11. Positioning one of the clusters between two of the environment obstacles caused the UAV to explore such area patch first (owing to be at a closer distance than the cluster at the top). As expected, the UAV followed the same navigation route once it cleared the first cluster as this strategy requires less time steps than alternative routes.

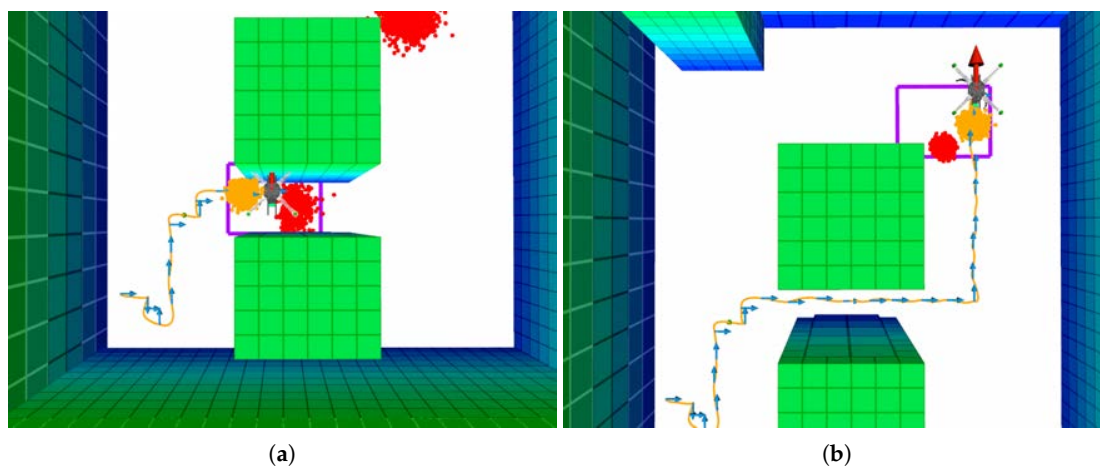


Figure 11. Flight mission example trajectory under two defined belief clusters. (a) UAV motion policy status when it reached the first cluster. (b) UAV motion policy after clearing the first cluster and reaching the second cluster.

The behaviour of the UAV with uniform clusters slightly differed compared to other case studies. Even though the UAV executed motion policies following similar global trajectories as with the first discussed case study (crossing and avoiding obstacles), a zigzag pattern was visualised as illustrated in Figure 12.

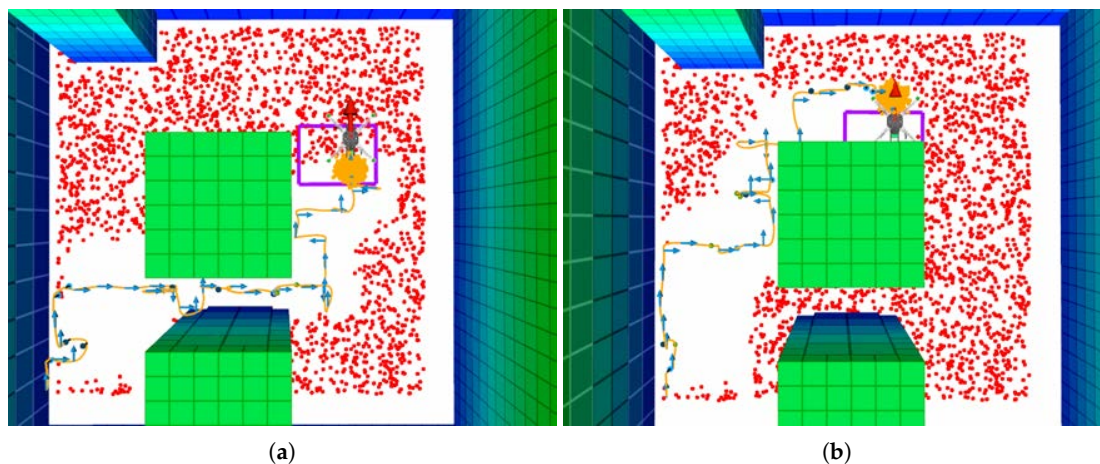


Figure 12. Example global trajectories followed by the UAV under a uniform belief cluster in flight tests. (a) Motion policy that encourages the UAV to cross over the column obstacles. (b) Motion policy that encourages the UAV that avoids crossing over the column obstacles.

For real flight tests where the victim was not detected, the traversed path by the UAV followed a pattern to cover remaining unexplored areas until it reached its maximum endurance, as shown in Figure 13.

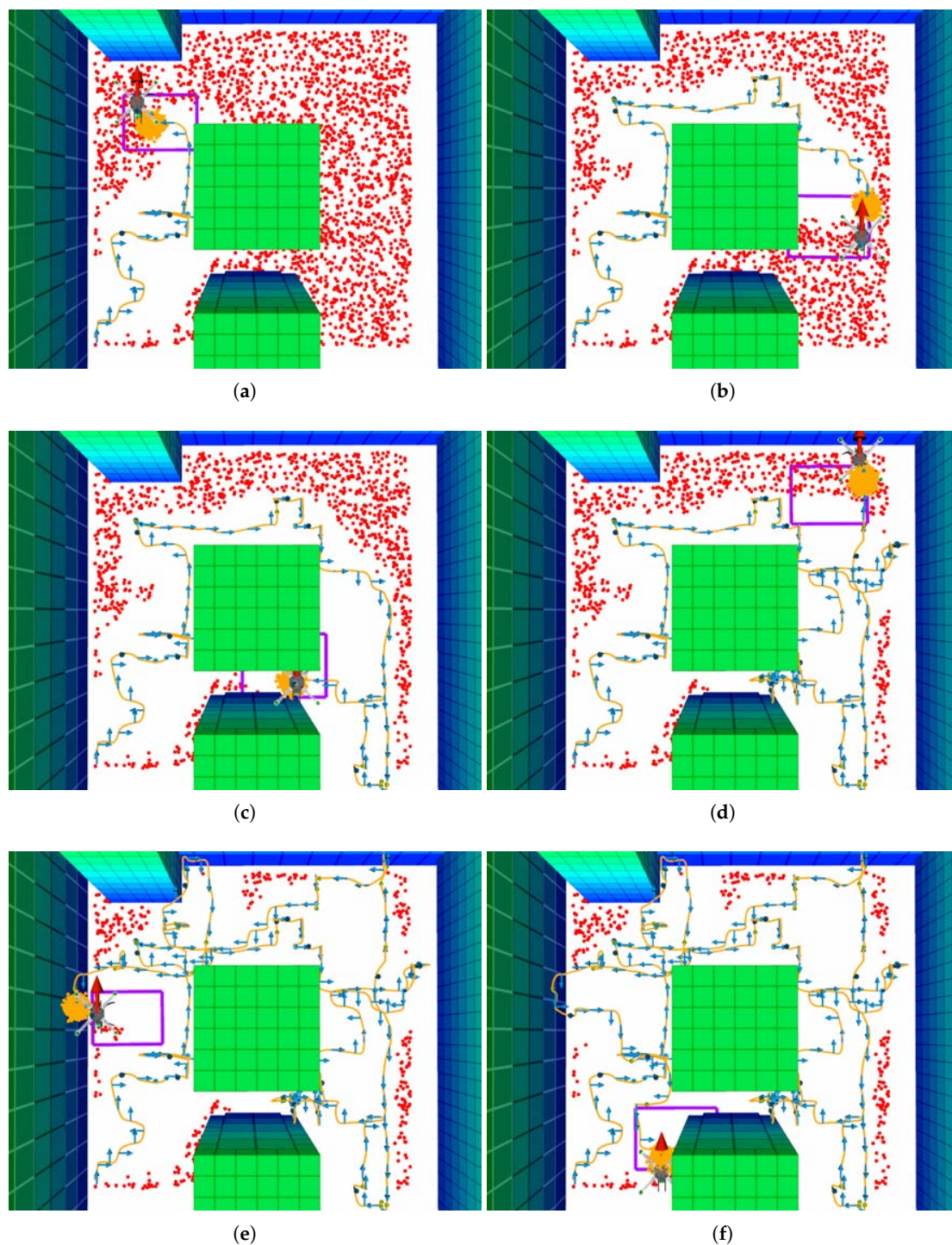


Figure 13. Example traversed path in a real flight test if a victim is undetected under a uniform belief cluster. (a) Traversed path at time step $k = 30$; (b) $k = 55$; (c) $k = 75$; (d) $k = 105$; (e) $k = 150$; (f) $k = 165$.

During the data collection process through real flight tests, the UAV experienced small stability problems in some runs, as seen in Figure 14. The issues altered the smoothness of the UAV trajectory but did not cause any consequence for the mission goal of finding the victim.

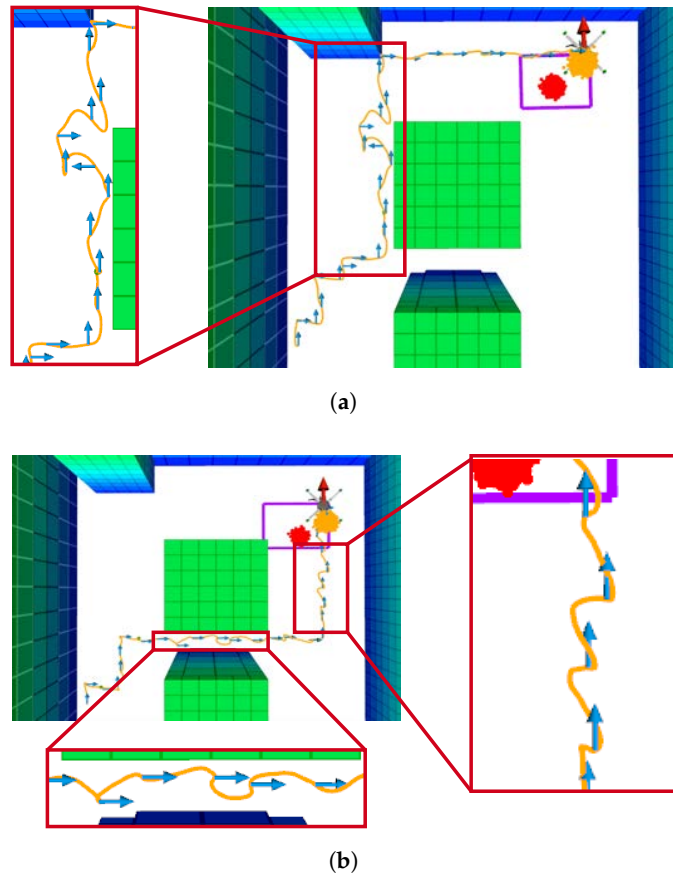


Figure 14. Stability anomalies on the traced UAV trajectory during real flight tests. (a) Noisy UAV trajectory in a mission with single belief cluster. (b) Noisy UAV trajectory in a mission with dual belief cluster.

Each one of the displayed arrows illustrate the action commands from the decision-making module at every time step. The arrows indicate how the UAV was still capable of following a consistent motion policy despite the added uncertainty by the aggressive motion response of the UAV. An analysis of recorded flight logs suggests that these perturbations in the UAV motion were caused by a sub-optimal performance of the FCU position controller. Specifically, the constant mounting and dismounting of the UAV LiPo battery during real flight tests provoked unintentional balancing issues, which may have altered the transition function of the UAV from which the flight controller was tuned by default. A graphical comparison of the UAV position response between two sets of flights is shown in Figures A1 and A2. The results suggest, however, that the system is robust enough to account for uncertainties caused by position estimation errors from the UAV motion module and still accomplish the flight mission.

An analysis of executed time steps to find the victim with a detection confidence $\zeta \geq 0.7$ was also conducted to assess the performance of the system by analysing repeatability in the measurements and gaps between HIL simulations and real flight tests. Box plots summarising the nature of collected data are shown in Figure 15.

Experiments defined with a single belief cluster (Figure 15a) presented a median of 32 and 31 time steps in simulation and flight tests respectively. Tests with dual (Figure 15b) and uniform belief clusters (Figure 15c) reported a higher variance with median values of 46.5 and 29 time steps, and 99.5 and 59.5 time steps respectively. The variance in the length of the whiskers between simulation and flight tests in all the case studies was caused by the equations that define the UAV motion in Gazebo.

Those functions approximate by default the motion response of a generic multi-rotor UAV rather than the Holybro S500 frame utilised for this research.

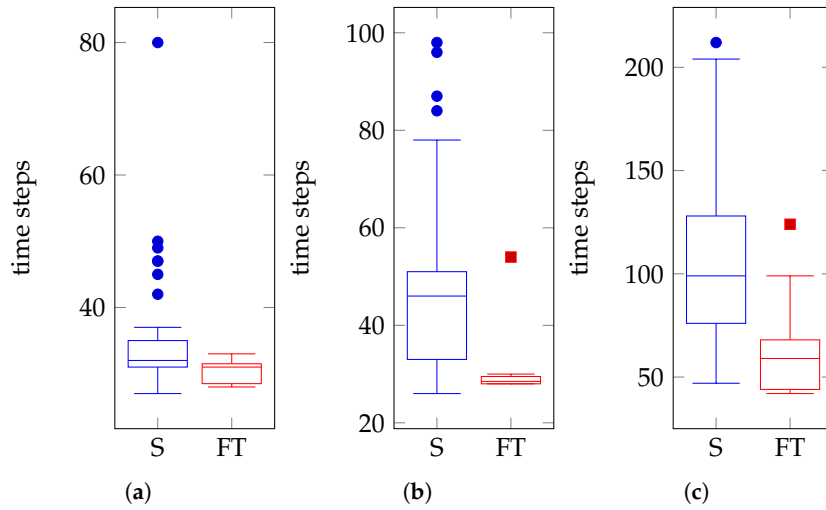


Figure 15. Data distribution of executed time steps by the POMDP solver until the victim was detected with a confidence $\zeta \geq 0.7$ and position coordinate $p_v = (1.5, -1.4, 0.0)$. HIL simulations (S) are indicated in blue and real flight tests (FT) in red. (a) Results under a single victim belief cluster. (b) Results under two victim belief clusters. (c) Results under a uniform belief cluster.

Bigger top whiskers and several outliers were also present in the distribution of collected data. These abnormal time step values and data asymmetry occurred as part of the proposed problem formulation, where victim belief particles are repopulated in the flying area only if the UAV finishes exploring the area of interest and is unable to find any victims. As a result, the UAV was prompted to further explore again and iterate around the cluster area. Additional time steps were also registered in situations where the UAV was able to detect a victim but with a confidence rate below the defined threshold value. In these situations, the UAV was encouraged to take additional actions in order to increase the confidence rate and confirm a detection. Some examples of recorded simulation test outliers are shown in Figure 16.

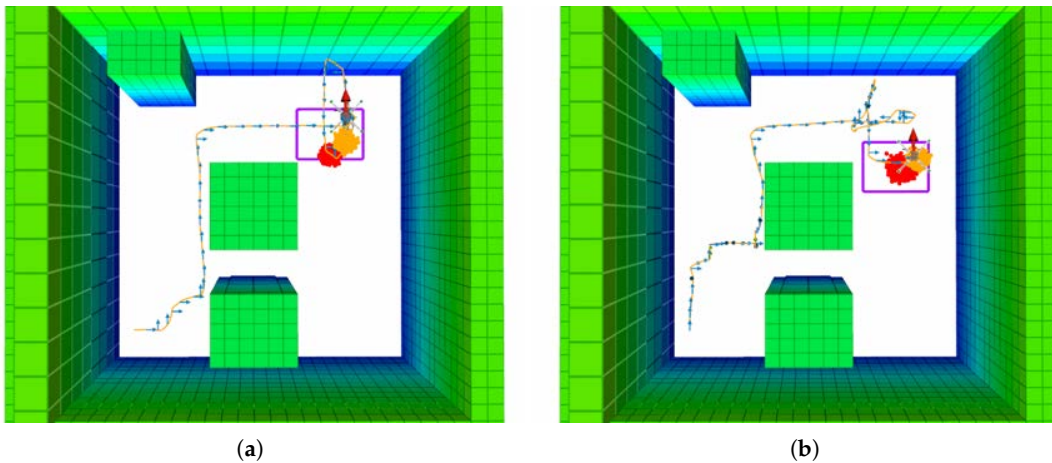
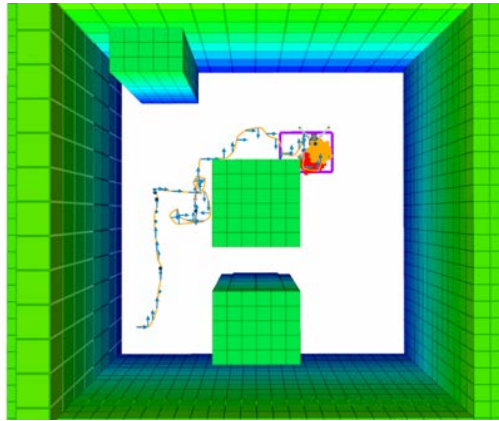


Figure 16. *Cont.*



(c)

Figure 16. Anomalies on the amount of executed time steps in simulation. (a) Example trajectory with repopulated single cluster. (b) Example trajectory for an initially defined dual cluster, repopulated after low detection confidence values. (c) Example abnormal trajectory under a uniform cluster.

7. Discussion

The proposed system architecture represents a competitive approach in the domain of onboard UAV decision-making under environment uncertainty and partial observability in GNSS-denied environments. This research extends the contributions of Vanegas and Gonzalez [24] by running the computer vision and decision-making modules onboard the UAV companion computer instead of using an external workstation and having a strong dependency from communication modules to transfer and process the data. Furthermore, the complexity for the target detection task was increased by detecting an adult mannequin instead of predefined augmented vision markers. Similarly, results obtained by Sandino et al. [40] are further enhanced by: (1) offering a more robust simulation environment (PX4 flight controller and onboard computer in HIL) and a more comprehensive system evaluation illustrating results in both simulation and hardware and; (2) extending the problem formulation by incorporating detection errors from the computer vision module (through the modelling of the detection confidence o_z in the problem formulation, covered in Section 4.4.5), rather than assuming detections with null instances of false positives. Obtained results also suggest that the traced trajectories by the UAV became smoother after including a concept of memory by recording previously explored areas from the flying area and adding the reward r_{fov} (defined in Section 4.4.4) in the reward function.

Overall, this study presents a flexible framework that provides scalability through portability depending on the flight mission goals, provided sensors and run algorithms for vision and decision-making. The mathematical formulation of the problem as a model-based POMDP brings enough flexibility to expand the functionality of the system with other multi-rotor UAVs of variable size as long as the dynamic model of the aerial platform and the environment are available to the researchers. The problem formulation covered in Section 4.4 is not specific to the scope of the experimental design of this paper and can be expanded to bigger surveying areas with more complex occupancy map representations. In fact, the UAV motion and flying boundaries of the UAV and victim can be increased without impacting the performance of the ABT solver, Octomap and TAPIR toolkits.

Despite the progress discussed in this paper, there are still several challenges which need to be addressed in the future. First, the occupancy map was provided before flying the aircraft and it was not updated mid-flight. Even though it is possible to reconstruct occupancy maps by using, for example, existing building floor plans of the surveyed area, it might not be suitable to fly in more complex environments with dynamic obstacles. Additionally, not updating the occupancy map online constitutes a strong dependency on the local position estimation system (i.e., visual odometry sensor). Indeed, the risk of the UAV colliding with other obstacles increases if position estimation errors are high. Augmenting the proportion of the obstacles was an applied workaround during

the data collection process. Under the assumption of operating with preloaded occupancy maps, the system complexity was simplified by not including UAV actions such as changes in its heading direction. However, it is possible to either incorporate heading actions in the problem formulation and disregard the backward, left and right action commands, or include additional sensors which can provide enough sensing coverage around the UAV in future experiments. The use of one or many depth cameras or a LiDAR sensor to update the occupancy map mid-flight might diminish the likelihood of collisions, regardless of the error magnitude of the local position estimation algorithm. Second, the off-the-shelf model to detect persons might not produce the expected performance at more complex experiment setups. For instance, detections are likely to be poor when the person is visualised in conditions from which the detector was not trained for such as debris occluding the person and if the person is observed from other camera perspectives as the one shown in Figure 3b.

Previous research also indicates the convenience of using sensors to detect bio-signals from humans, such as microphones for audio signals, thermal cameras, gas sensors and Doppler radars for breathing and heart-beating signals respectively [54–56]. Bio-signals have also been proven to be detected through the use of computer vision and RGB cameras as long as UAVs are positioned close enough to the victims [15]. Even though the employed MobileNet SSD detector is efficient enough to distinguish the presence of persons regardless of their health conditions, the modularity design presented in the system architecture allows adding other vision-based detectors without substantial modifications, so that they could provide further valuable data to the decision-making module (in the form of observations), UAV operators and SAR squads. Additional sensors such as thermal and depth cameras can also be added to the UAV frame without altering the workflow of the system architecture. Moreover, other decision-making algorithms could also be ported to the framework with ease, such as model-free reinforcement learning, Observe–Orient–Decide–Act (OODA) loops, Bayesian networks, etc. Nevertheless, a comparison study between the trade-offs of other decision-making algorithms for UAV navigation under environment uncertainty should be reviewed in future research before adapting them to the framework.

A successful implementation of the proposed system in real disaster events requires the examination of various practical challenges, which include but are not limited to:

- The size of UAV, which may restrict the survey in very confined places and compromise the integrity of the UAV and nearby victims, if any.
- Low lighting conditions, which might decrease the performance of the visual odometry system and people/object detector.
- The UAV endurance, which could constrain SAR operations if the remote assessment of a hazardous structure exceeds 20 min of flying time.
- Collisions, which may occur owing to the absence of propeller protectors or if victims make accidental contact with the UAV.
- Chemical and electrical hazards present in the surveyed area, which may compromise electronic circuits and sensors.
- Mishandling of LiPo batteries, which might provoke fire hazards if not isolated from impacts and ignition sources.

An additional extension to the UAV frame could be the incorporation of a front-view camera, which will aid the assessment and SAR logistics if physical intervention to the surveyed structure is necessary.

8. Conclusions and Future Work

This paper presented a framework for automated UAV motion planning under target location uncertainty in cluttered, GNSS-denied environments. The system architecture details the functionality of system modules for unsupervised decision making onboard resource-constrained hardware platforms such as small UAVs (MTOW \leq 13.5 kg). The proposed approach is illustrated in the SAR context by locating a victim in a simulated office building. The system is validated using HIL

simulations to ensure a high-fidelity setup against real-world conditions; and real flight tests using a multi-rotor UAV frame and vision-based sensors for SLAM and collection of system observations.

The problem is mathematically formulated as a POMDP, whose probabilistic model allows representing uncertainty with probability distributions. This approach allows defining potential locations of the victim with normal and uniform probabilistic distributions (and, thus, model victim location uncertainty). The performance of the UAV was evaluated under three case studies of situational awareness; a single cluster of victim coordinates covering a small patch from the surveyed environment; two clusters of victim coordinates covering two areas of interest; and a cluster of victim coordinates uniformly distributed across the flying area. Incorporating the ABT algorithm as the POMDP solver does not only provide the system with a UAV motion policy in seconds prior to any flight mission, but it also improves previously computed policies mid-flight by modelling potential changes in the environment and levels of uncertainty based on possible future actions in its internal search tree. This feature allows the UAV to optimise its behaviour in various scenarios such as preserving a constant path under unstable UAV motion response or holding its position while more episodes and internal simulations are generated to execute a better policy. Ultimately, the system ensures rapid monitoring and personal safety by letting the UAV to explore the area without UAV operator intervention.

The primary contributions of this paper are:

1. A UAV framework for autonomous navigation under target detection uncertainty. The computer vision and decision-making modules run onboard resource-constrained hardware (i.e., a companion computer mounted to the UAV), discarding the dependency of the UAV from third-party systems to perform its motion policy calculations. The framework offers enough flexibility to expand or adapt the functionality of the system by using other vision or light-based sensors, UAV frames and onboard computing hardware.
2. An approach to handle target detection uncertainty from false positive detection instances through the concept of detection confidence and the definition of a confirmed detection in the POMDP problem formulation.
3. A detailed case study of the implementation of the system modules for a simulated land SAR mission in GNSS-denied environments, which allows integrating the flight controller and companion computer under HIL simulations to bridge the gap between simulations and real flight tests.

A recorded system demonstration can be observed using the following link: <https://youtu.be/fEWVd-GC7Fs>.

Future avenues for research include evaluation of the performance of the system with an unknown environment map, dynamic obstacles and the robustness of the POMDP solver under environment changes by updating the occupancy map mid-flight. An additional system performance analysis by locating the victim at various locations and finding multiple victims might help to understand the limits of the proposed framework. A study comparing the performance between the ABT solver and other model-based POMDP solvers, and other decision-making algorithms should also be conducted. Evaluating the performance of the UAV using an object detector tuned with a domain-specific dataset (i.e., footage of people in distress and at various occlusion levels) will aid the understanding of the UAV capacity to detect victims under more challenging scenarios. Incorporating widely used sensors for land SAR such as thermal cameras as well as processing camera frames at variable gimbal angle configurations is expected to be conducted in future system implementations.

Author Contributions: Conceptualisation, J.S., F.V., F.M. and F.G.; methodology, J.S., F.V. and F.G.; hardware conceptualisation and integration, F.V.; software, J.S. and F.V.; validation, J.S.; formal analysis, J.S.; investigation, J.S. and F.V.; resources, F.G.; data curation, J.S.; writing—original draft preparation, J.S.; writing—review and editing, F.V., F.M., P.C., C.S. and F.G.; visualisation, J.S.; supervision, F.M., P.C., C.S. and F.G.; project administration, F.G.; funding acquisition, P.C. and F.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The Commonwealth Scientific and Industrial Research Organisation (CSIRO) through the CSIRO Data61 PhD and Top Up Scholarships (Agreement 50061686), The Australian Research Council (ARC) through the ARC Discovery Project 2018 “Navigating under the forest canopy and in the urban jungle” (grant number ARC DP180102250) and The Queensland University of Technology (QUT) through the Higher Degree Research (HDR) Tuition Fee Sponsorship. The APC was funded by the QUT Office for Scholarly Communication (OSC).

Acknowledgments: The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. The authors would also like to gratefully thank the QUT Research Engineering Facility (REF) team for their technical support that made possible conducting this research work.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

2D	Two-dimensional
3D	Three-dimensional
ABT	Augmented Belief Trees
D-CNN	Deep Convolutional Neural Network
FCU	Flight Controller Unit
FOV	Field of View
FPS	Frames per Second
GNSS	Global Navigation Satellite System
HIL	Hardware in the Loop
LiDAR	Light Detection and Ranging
MDP	Markov Decision Process
MTOW	Maximum Take-off Weight
OS	Operating System
POMCP	Partially Observable Monte Carlo Planning
POMDP	Partially Observable Markov Decision Process
RGB	Red, Green, Blue
ROI	Region of Interest
ROS	Robot Operating System
SAR	Search and Rescue
SIL	Software in the Loop
SLAM	Simultaneous Localisation and Mapping
SSD	Single Shot Detector
TAPIR	Toolkit for approximating and Adapting POMDP solutions In Real time
UAV	Unmanned Aerial Vehicle
VPU	Vision Processing Unit

Appendix A

Figures [A1](#) and [A2](#) show an example UAV position response comparison between a smooth and a noisy traversed path.

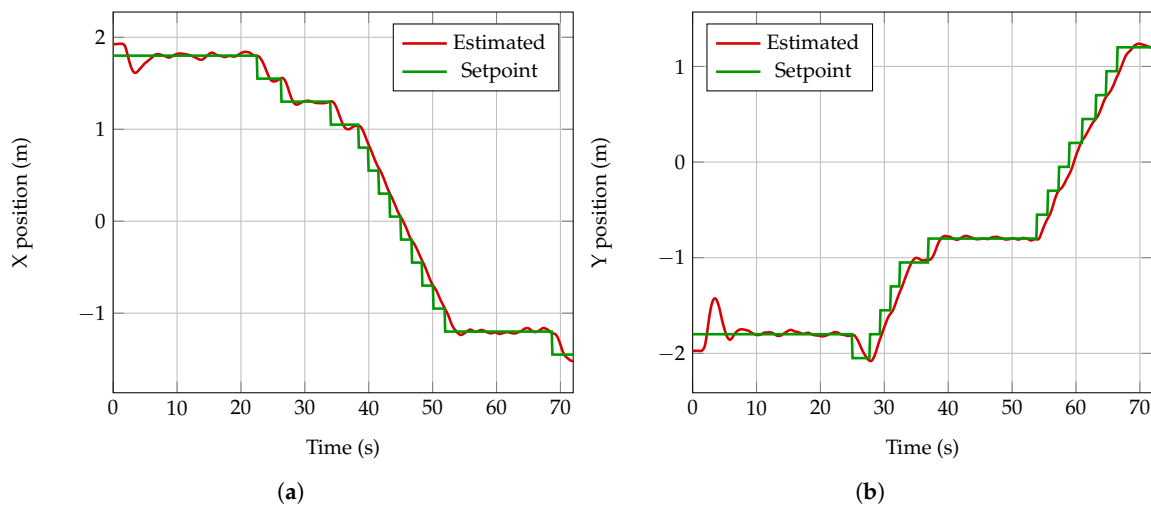


Figure A1. UAV position response for a smooth traversed path for (a) x and (b) y axes.

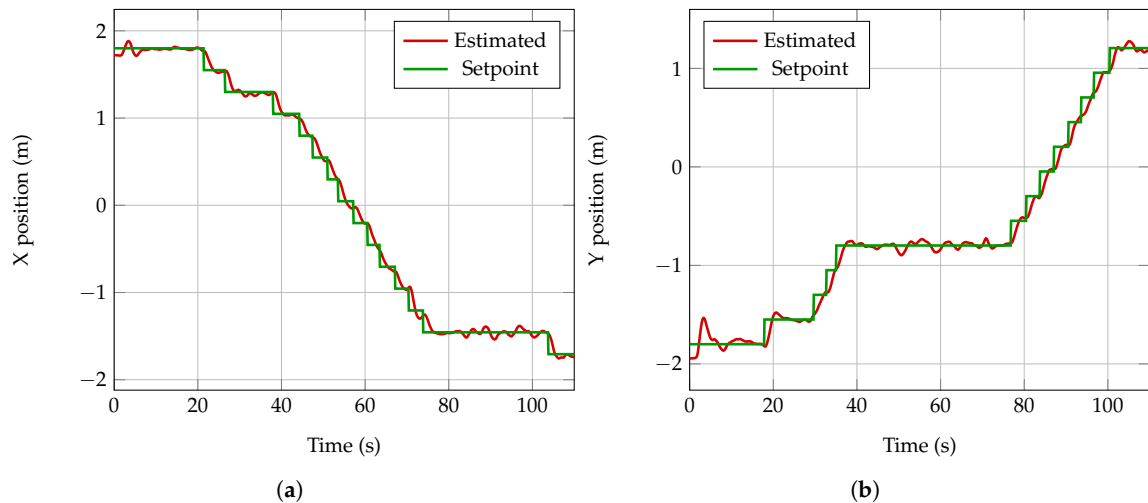


Figure A2. UAV position response for a noisy traversed path for (a) x and (b) y axes.

References

1. Doocy, S.; Daniels, A.; Packer, C.; Dick, A.; Kirsch, T.D. The Human Impact of Earthquakes: A Historical Review of Events 1980–2009 and Systematic Literature Review. *PLoS Curr.* **2013**. doi:10.1371/currents.dis.67bd14fe457f1db0b5433a8ee20fb833.
2. Shapira, S.; Levi, T.; Bar-Dayana, Y.; Aharonson-Daniel, L. The impact of behavior on the risk of injury and death during an earthquake: A simulation-based study. *Nat. Hazards* **2018**, *91*, 1059–1074. doi:10.1007/s11069-018-3167-5.
3. Cobum, A.; Spence, R.J.S.; Pomonis, A. Factors determining human casualty levels in earthquakes: Mortality prediction in building collapse. In Proceedings of the Earthquake Engineering, Tenth World Conference, Madrid, Spain, 19–24 July 1992; IIT Kanpur: Balkema, Rotterdam, 1992; pp. 5989–5994.
4. Pajares, G. Overview and Current Status of Remote Sensing Applications Based on Unmanned Aerial Vehicles (UAVs). *Photogramm. Eng. Remote Sens.* **2015**, *81*, 281–330. doi:10.14358/PERS.81.4.281.
5. Dalamagkidis, K.; Valavanis, K.P.; Piegler, L.A. *On Integrating Unmanned Aircraft Systems into the National Airspace System*; Springer: Dordrecht, The Netherlands, 2012; pp. 1–305. doi:10.1007/978-94-007-2479-2.

6. Erdelj, M.; Natalizio, E. UAV-assisted disaster management: Applications and open issues. In Proceedings of the International Conference on Computing, Networking and Communications, Kauai, HI, USA, 15–18 February 2016; pp. 1–5. doi:10.1109/ICCNC.2016.7440563.
7. Jiménez López, J.; Mulero-Pázmány, M. Drones for Conservation in Protected Areas: Present and Future. *Drones* **2019**, *3*, 10. doi:10.3390/drones3010010.
8. Estrada, M.A.R.; Ndoma, A. The uses of unmanned aerial vehicles –UAV’s- (or drones) in social logistic: Natural disasters response and humanitarian relief aid. *Procedia Comput. Sci.* **2019**, *149*, 375–383. doi:10.1016/j.procs.2019.01.151.
9. Karaca, Y.; Cicek, M.; Tatli, O.; Sahin, A.; Pasli, S.; Beser, M.F.; Turedi, S. The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations. *Am. J. Emerg. Med.* **2018**, *36*, 583–588. doi:10.1016/j.ajem.2017.09.025.
10. Mavroulis, S.; Andreadakis, E.; Spyrou, N.I.; Antoniou, V.; Skourtsos, E.; Papadimitriou, P.; Kassaras, I.; Kaviris, G.; Tselentis, G.A.; Voulgaris, N.; et al. UAV and GIS based rapid earthquake-induced building damage assessment and methodology for EMS-98 isoseismal map drawing: The June 12, 2017 Mw 6.3 Lesvos (Northeastern Aegean, Greece) earthquake. *Int. J. Disaster Risk Reduct.* **2019**, *37*, 101169. doi:10.1016/j.ijdrr.2019.101169.
11. Zhang, R.; Li, H.; Duan, K.; You, S.; Liu, K.; Wang, F.; Hu, Y. Automatic Detection of Earthquake-Damaged Buildings by Integrating UAV Oblique Photography and Infrared Thermal Imaging. *Remote Sens.* **2020**, *12*, 2621. doi:10.3390/rs12162621.
12. Claesson, A.; Svensson, L.; Nordberg, P.; Ringh, M.; Rosenqvist, M.; Djarv, T.; Samuelsson, J.; Hernborg, O.; Dahlbom, P.; Jansson, A.; et al. Drones may be used to save lives in out of hospital cardiac arrest due to drowning. *Resuscitation* **2017**, *114*, 152–156. doi:10.1016/j.resuscitation.2017.01.003.
13. Claesson, A.; Bäckman, A.; Ringh, M.; Svensson, L.; Nordberg, P.; Djärv, T.; Hollenberg, J. Time to Delivery of an Automated External Defibrillator Using a Drone for Simulated Out-of-Hospital Cardiac Arrests vs. Emergency Medical Services. *JAMA* **2017**, *317*, 2332. doi:10.1001/jama.2017.3957.
14. Ballous, K.A.; Khalifa, A.N.; Abdulwadood, A.A.; Al-Shabi, M.; El Haj Assad, M. Medical kit: Emergency drone. In *Unmanned Systems Technology XXII*; Shoemaker, C.M., Muench, P.L., Nguyen, H.G., Eds.; SPIE: Bellingham, WA, USA, 2020; Volume 11425, p. 39. doi:10.1117/12.2566115.
15. Al-Naji, A.; Perera, A.G.; Mohammed, S.L.; Chahl, J. Life Signs Detector Using a Drone in Disaster Zones. *Remote Sens.* **2019**, *11*, 2441. doi:10.3390/rs11202441.
16. Mishra, B.; Garg, D.; Narang, P.; Mishra, V. Drone-surveillance for search and rescue in natural disaster. *Comput. Commun.* **2020**, *156*, 1–10. doi:10.1016/j.comcom.2020.03.012.
17. Stark, B.J.; Chen, Y.Q. Remote Sensing Methodology for Unmanned Aerial Systems. In *Unmanned Aircraft Systems*; Atkins, E.M., Ollero, A., Tsourdos, A., Eds.; Wiley: New York, NY, USA, 2016; Chapter 2, pp. 17–27.
18. Bähnemann, R.; Pantic, M.; Popović, M.; Schindler, D.; Tranzatto, M.; Kamel, M.; Grimm, M.; Widauer, J.; Siegwart, R.; Nieto, J. The ETH-MAV Team in the MBZ International Robotics Challenge. *J. Field Robot.* **2019**, *36*, 78–103. doi:10.1002/rob.21824.
19. Carrio, A.; Sampedro, C.; Rodriguez-Ramos, A.; Campoy, P. A Review of Deep Learning Methods and Applications for Unmanned Aerial Vehicles. *J. Sensors* **2017**, *2017*, 1–13. doi:10.1155/2017/3296874.
20. Bravo, R.Z.B.; Leiras, A.; Cyrino Oliveira, F.L. The Use of UAVs in Humanitarian Relief: An Application of POMDP-Based Methodology for Finding Victims. *Prod. Oper. Manag.* **2019**, *28*, 421–440. doi:10.1111/poms.12930.
21. Kochenderfer, M.J. *Decision Making under Uncertainty: Theory and Application*; MIT Press: Cambridge, MA, USA, 2015; pp. 1–323.
22. Hubmann, C.; Schulz, J.; Becker, M.; Althoff, D.; Stiller, C. Automated Driving in Uncertain Environments: Planning With Interaction and Uncertain Maneuver Prediction. *IEEE Trans. Intell. Veh.* **2018**, *3*, 5–17. doi:10.1109/TIV.2017.2788208.
23. Rizk, Y.; Awad, M.; Tunstel, E.W. Decision Making in Multiagent Systems: A Survey. *Trans. Cogn. Dev. Syst.* **2018**, *10*, 514–529. doi:10.1109/TCDS.2018.2840971.
24. Vanegas, F.; Gonzalez, F. Enabling UAV Navigation with Sensor and Environmental Uncertainty in Cluttered and GPS-Denied Environments. *Sensors* **2016**, *16*, 666. doi:10.3390/s16050666.

25. Silver, D.; Veness, J. Monte-Carlo Planning in Large POMDPs. In *Advances in Neural Information Processing Systems 23*; Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A., Eds.; Curran Associates, Inc.: Vancouver, BC, Canada, 2010; pp. 2164–2172.
26. Kurniawati, H.; Yadav, V. An Online POMDP Solver for Uncertainty Planning in Dynamic Environment. In *Robotics Research. Springer Tracts in Advanced Robotics*; Inaba, M., Corke, P., Eds.; Springer: Cham, Switzerland, 2016; Volume 114, pp. 611–629. doi:10.1007/978-3-319-28872-7_35.
27. Garrido-Jurado, S.; Muñoz-Salinas, R.; Madrid-Cuevas, F.; Marín-Jiménez, M. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.* **2014**, *47*, 2280–2292. doi:10.1016/j.patcog.2014.01.005.
28. Valavanis, K.P.; Vachtsevanos, G.J. Future of Unmanned Aviation. In *Handbook of Unmanned Aerial Vehicles*; Valavanis, K.P., Vachtsevanos, G.J., Eds.; Springer: Dordrecht, The Netherlands, 2015; Chapter 126, pp. 2993–3009. doi:10.1007/978-90-481-9707-1_95.
29. Ragi, S.; Chong, E.K.P. UAV Path Planning in a Dynamic Environment via Partially Observable Markov Decision Process. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 2397–2412. doi:10.1109/TAES.2013.6621824.
30. Ragi, S.; Chong, E.K.P. UAV Guidance Algorithms via Partially Observable Markov Decision Processes. In *Handbook of Unmanned Aerial Vehicles*; Valavanis, K., Vachtsevanos, G., Eds.; Springer: Dordrecht, The Netherlands, 2015; Chapter 73, pp. 1775–1810. doi:10.1007/978-90-481-9707-1_59.
31. Waharte, S.; Trigoni, N. Supporting Search and Rescue Operations with UAVs. In Proceedings of the International Conference on Emerging Security Technologies, Canterbury, UK, 6–7 September 2010; IEEE: Canterbury, UK, 2010; pp. 142–147. doi:10.1109/EST.2010.31.
32. Chen, M.; Frazzoli, E.; Hsu, D.; Lee, W.S. POMDP-lite for Robust Robot Planning under Uncertainty. *Int. Conf. Robot. Autom.* **2016**, 5427–5433. doi:10.1109/ICRA.2016.7487754.
33. Ponzoni Carvalho Chanel, C.; Albore, A.; T’Hooft, J.; Lesire, C.; Teichteil-Königsbuch, F. AMPLE: An anytime planning and execution framework for dynamic and uncertain problems in robotics. *Auton. Robot.* **2019**, *43*, 37–62. doi:10.1007/s10514-018-9703-z.
34. Ong, S.W.C.; Png, S.W.; Hsu, D.; Lee, W.S. *POMDPs for Robotic Tasks with Mixed Observability*; Robotics: Science and Systems V; Robotics: Science and Systems Foundation: Seattle, WA, USA, 2009. doi:10.15607/RSS.2009.V.026.
35. Ilhan, U.; Gardashova, L.; Kilic, K. UAV Using Dec-POMDP Model for Increasing the Level of Security in the Company. *Procedia Comput. Sci.* **2016**, *102*, 458–464. doi:10.1016/j.procs.2016.09.427.
36. Zhao, Y.; Wang, X.; Kong, W.; Shen, L.; Jia, S. Decision-making of UAV for tracking moving target via information geometry. In Proceedings of the Chinese Control Conference, Chengdu, China, 27–29 July 2016; IEEE: Chengdu, China, 2016; pp. 5611–5617. doi:10.1109/ChiCC.2016.7554231.
37. Yang, Q.; Zhang, J.; Shi, G. Path planning for unmanned aerial vehicle passive detection under the framework of partially observable markov decision process. In Proceedings of the Chinese Control and Decision Conference, Shenyang, China, 9–11 June 2018; pp. 3896–3903. doi:10.1109/CCDC.2018.8407800.
38. Chanel, C.; Teichteil-Königsbuch, F.; Lesire, C. Multi-target detection and recognition by UAVs using online POMDPs. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013; AAAI Press: Bellevue, Washington, 2013; pp. 1381–1387.
39. Klimenko, D.; Song, J.; Kurniawati, H. TAPIR: A software Toolkit for approximating and adapting POMDP solutions online. In Proceedings of the Australasian Conference on Robotics and Automation, Melbourne, Australia, 2–4 December 2014; ARAA: Melbourne, Australia, 2014; pp. 1–9.
40. Sandino, J.; Vanegas, F.; Gonzalez, F.; Maire, F. Autonomous UAV Navigation for Active Perception of Targets in Uncertain and Cluttered Environments. In Proceedings of the Aerospace Conference, Big Sky, MT, USA, 7–14 March 2020; IEEE: Big Sky, MT, USA, 2020; pp. 1–12. doi:10.1109/AERO47225.2020.9172808.
41. Dutech, A.; Scherrer, B. Partially Observable Markov Decision Processes. In *Markov Decision Processes in Artificial Intelligence*; Sigaud, O., Buffet, O., Eds.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2013; Chapter 7, pp. 185–228. doi:10.1002/9781118557426.ch7.
42. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robot.*; MIT Press: Cambridge, MA, USA, 2005; pp. 485–542.
43. Papadimitriou, C.H.; Tsitsiklis, J.N. The Complexity of Markov Decision Processes. *Math. Oper. Res.* **1987**, *12*, 441–450. doi:10.1287/moor.12.3.441.

44. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
45. Open Source Robotics Foundation. Robot Operating System. Available online: <https://www.ros.org> (accessed on 30 August 2020).
46. Meier, L.; Honegger, D.; Pollefeys, M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *Proceedings of the International Conference on Robotics and Automation*, Seattle, WA, USA, 26–30 May 2015; pp. 6235–6240. doi:10.1109/ICRA.2015.7140074.
47. Koubaa, A.; Allouch, A.; Alajlan, M.; Javed, Y.; Belghith, A.; Khalgui, M. Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey. *IEEE Access* **2019**, *7*, 87658–87680. doi:10.1109/ACCESS.2019.2924410.
48. Chuanqi, Y. Caffe Implementation of Google MobileNet SSD Detection Network, with Pretrained Weights on VOC0712 and mAP=0.727. Available online: <https://github.com/chuanqi305/MobileNet-SSD> (accessed on 30 August 2020).
49. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the International Conference on Multimedia*, Orlando, FL, USA, 3–7 November 2014; ACM Press: New York, NY, USA, 2014; pp. 675–678. doi:10.1145/2647868.2654889.
50. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. doi:10.1007/s11263-014-0733-5.
51. Chovancová, A.; Fico, T.; Chovanec, L.; Hubinsk, P. Mathematical Modelling and Parameter Identification of Quadrotor (a survey). *Procedia Eng.* **2014**, *96*, 172–181. doi:10.1016/j.proeng.2014.12.139.
52. Bourke, P. Polygons and Meshes. Available online: <http://paulbourke.net/geometry/polygonmesh> (accessed on 25 September 2019).
53. Hornung, A.; Wurm, K.M.; Bennewitz, M.; Stachniss, C.; Burgard, W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robot.* **2013**, *34*, 189–206. doi:10.1007/s10514-012-9321-0.
54. Zhang, D.; Shiguematsu, Y.M.; Lin, J.Y.; Ma, Y.H.; Maamari, M.S.A.; Takanishi, A. Development of a Hybrid Locomotion Robot for Earthquake Search and Rescue in Partially Collapsed Building. In *Proceedings of the International Conference on Mechatronics and Automation*, Tianjin, China, 4–7 August 2019; pp. 2559–2564. doi:10.1109/ICMA.2019.8816327.
55. Arnold, R.D.; Yamaguchi, H.; Tanaka, T. Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *J. Int. Humanit. Action* **2018**, *3*, 18. doi:10.1186/s41018-018-0045-4.
56. Lee, S.; Har, D.; Kum, D. Drone-Assisted Disaster Management: Finding Victims via Infrared Camera and LiDAR Sensor Fusion. In *Proceedings of the 3rd Asia-Pacific World Congress on Computer Science and Engineering*, Nadi, Fiji, 5–6 December 2016; pp. 84–89. doi:10.1109/APWC-on-CSE.2016.025.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Chapter 5

Drone-based Autonomous Motion Planning System for Outdoor Environments under Object Detection Uncertainty

RECENT advances in UAV autonomy have led to their gradual use in remote sensing applications such as precision agriculture, biosecurity, disaster monitoring and surveillance. This chapter presents the third paper published from this research, a modular UAV system design for autonomous navigation and object finding in outdoor environments under partial observability and object detection uncertainty, and capable of operating in small UAVs with resource-constrained hardware. The POMDP problem formulation presented in Chapter 4 is extended here to have a more advanced reward function that determines the behaviour of the UAV for exploration, obstacle avoidance, and inspection of areas with potential presence of objects. The novel flight mode (*hybrid mode*) combines the waypoint navigation of UAVs in outdoor environments, and the autonomy of inspecting delimited regions using *offboard mode*. This paper presents an approach that successfully replicates real-world environments in the Gazebo robotics simulator by fusing collected data of a real scenario from airborne UAV geo-referenced RGB mosaic rasters, and LiDAR point clouds to produce 3D occupancy maps of the virtual environment. Experimental insights from an emulated SAR scenario revealed that increased cognitive power added by the proposed motion planner and flight modes enabled UAVs to collect more accurate victim coordinates compared to the baseline planner. The boosted capability of navigating autonomously and diminishing false positive readings from CNN object detectors improves the robustness of small UAVs in time-critical applications such as SAR.

Statement of Contribution of Co-Authors for Thesis by Published Paper

The authors listed below have certified that:

1. they meet the criteria for authorship and that they have participated in the conception, execution, or interpretation, of at least that part of the publication in their field of expertise;
2. they take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;
3. there are no other authors of the publication according to these criteria;
4. potential conflicts of interest have been disclosed to (a) granting bodies, (b) the editor or publisher of journals or other publications, and (c) the head of the responsible academic unit, and
5. they agree to the use of the publication in the student's thesis and its publication on the [QUT's ePrints site](#) consistent with any limitations set by publisher requirements.

In the case of this chapter:

J. Sandino, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez, "Drone-based autonomous motion planning system for outdoor environments under object detection uncertainty," *Remote Sensing*, vol. 13, no. 21, p. 4481, Nov. 2021. DOI: [10.3390/rs13214481](https://doi.org/10.3390/rs13214481).

Contributor	Statement of Contribution ¹
Juan Sandino QUT Verified Signature Date: <u>2022-05-20</u>	Contributed on: conceptualisation, methodology, software, validation, formal analysis, investigation, data curation, writing-original draft preparation and visualisation.
Frederic Maire	Contributed on: conceptualisation, writing-review and editing, and supervision.
Peter Caccetta	Contributed on: writing-review and editing, supervision, and funding acquisition.
Conrad Sanderson	Contributed on: writing-review and editing, and supervision.
Felipe Gonzalez	Contributed on: conceptualisation, methodology, resources, writing-review and editing, supervision, project administration, and funding acquisition.

Principal Supervisor Confirmation		
I have sighted email or other correspondence from all co-authors confirming their certifying authorship.		
<i>Felipe Gonzalez</i>	QUT Verified Signature	2022-05-20
_____ Name	_____ Signature	_____ Date

¹ Contributions follow the CRediT taxonomy. Further details can be found at <https://credit.niso.org/>.

Article

Drone-Based Autonomous Motion Planning System for Outdoor Environments under Object Detection Uncertainty

 Juan Sandino ^{1,2,*} , Frederic Maire ¹ , Peter Caccetta ² , Conrad Sanderson ^{2,3}  and Felipe Gonzalez ¹ 

¹ School of Electrical Engineering and Robotics, Queensland University of Technology (QUT), 2 George Street, Brisbane, QLD 4000, Australia; f.maire@qut.edu.au (F.M.); felipe.gonzalez@qut.edu.au (F.G.)

² Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Building 101, Clunies Ross Street, Black Mountain, ACT 2601, Australia; peter.caccetta@data61.csiro.au (P.C.); conrad.sanderson@data61.csiro.au (C.S.)

³ Institute for Integrated and Intelligent Systems, Griffith University, 170 Kessels Road, Nathan, QLD 4111, Australia

* Correspondence: sandinoj@qut.edu.au

Abstract: Recent advances in autonomy of unmanned aerial vehicles (UAVs) have increased their use in remote sensing applications, such as precision agriculture, biosecurity, disaster monitoring, and surveillance. However, onboard UAV cognition capabilities for understanding and interacting in environments with imprecise or partial observations, for objects of interest within complex scenes, are limited, and have not yet been fully investigated. This limitation of onboard decision-making under uncertainty has delegated the motion planning strategy in complex environments to human pilots, which rely on communication subsystems and real-time telemetry from ground control stations. This paper presents a UAV-based autonomous motion planning and object finding system under uncertainty and partial observability in outdoor environments. The proposed system architecture follows a modular design, which allocates most of the computationally intensive tasks to a companion computer onboard the UAV to achieve high-fidelity results in simulated environments. We demonstrate the system with a search and rescue (SAR) case study, where a lost person (victim) in bushland needs to be found using a sub-2 kg quadrotor UAV. The navigation problem is mathematically formulated as a partially observable Markov decision process (POMDP). A motion strategy (or policy) is obtained once a POMDP is solved mid-flight and in real time using augmented belief trees (ABT) and the TAPIR toolkit. The system's performance was assessed using three flight modes: (1) mission mode, which follows a survey plan and used here as the baseline motion planner; (2) offboard mode, which runs the POMDP-based planner across the flying area; and (3) hybrid mode, which combines mission and offboard modes for improved coverage in outdoor scenarios. Results suggest the increased cognitive power added by the proposed motion planner and flight modes allow UAVs to collect more accurate victim coordinates compared to the baseline planner. Adding the proposed system to UAVs results in improved robustness against potential false positive readings of detected objects caused by data noise, inaccurate detections, and elevated complexity to navigate in time-critical applications, such as SAR.

Keywords: unmanned aerial system (UAS); unmanned aerial vehicle (UAV); artificial intelligence (AI); embedded systems; machine learning (ML); search and rescue (SAR); computer vision (CV); sequential decision-making; robotics; partially observable Markov decision process (POMDP)



Citation: Sandino, J.; Maire, F.; Caccetta, P.; Sanderson, C.; Gonzalez, F. Drone-Based Autonomous Motion Planning System for Outdoor Environments under Object Detection Uncertainty. *Remote Sens.* **2021**, *13*, 4481. <https://doi.org/10.3390/rs13214481>

Academic Editor: Ming-Der Yang

Received: 6 October 2021

Accepted: 3 November 2021

Published: 8 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The elevated number of stranded people and human loss caused by natural disasters, weather events, crime, and military conflicts is an ever-present issue [1]. According to the Australian Institute of Criminology, an average of 38,000 people are reported missing per year in Australia [2]. Unfortunately, around 2% of them (or 720 persons) are never found, and 2600 are reported (long-term) missing for more than three months. Thus, it is beneficial

to improve search and rescue (SAR) efforts by developing technology that cooperates with first responders to locate as many victims as soon as an emergency situation is declared.

Recent advances in autonomy of unmanned aerial vehicles (UAVs)—or drones—have increased their adoption in time-critical applications, such as disaster monitoring, surveillance, and SAR [3–6]. Compared to manned aircraft, UAVs offer high versatility to scout areas due to their reduced size and cost, extensive payload adaptability, and ease of piloting by incorporating automated tasks, including waypoint navigation, obstacle avoidance, and autonomous take-off and landing [7–9]. In time-critical applications, such as SAR, real-time camera streaming is critical for UAV operators to understand the situation context and manoeuvre the aircraft during the mission [10]. However, this strong reliance in the communication system compromises the usability of UAVs if such systems fail [11]. Furthermore, operators are susceptible to fatigue in the long run by controlling the UAV and photo-interpreting streamed camera frames to find victims [12]. As a result, developing decision-making algorithms onboard UAVs could further expand their applicability in civilian and time-critical applications. UAVs enriched with autonomous onboard object detection for real-time decision making eliminate the dependency of UAV pilots to photo-interpret streamed frames, and in their communication systems to control the aircraft.

Real-world and time-critical UAV applications are full of uncertainties. As shown in Figure 1, uncertainty comes from uncontrolled external factors, such as unknown environment conditions, strong wind gusts, dynamic obstacles, suboptimal survey altitude, and partial observability. Research has demonstrated that applied theory on decision making using partially observable Markov decision processes (POMDPs) can increase cognition capabilities in UAVs for autonomous path planning under uncertainty [13–15]. UAV frameworks for real-time target finding and tracking in cluttered indoor environments under uncertainty [16,17] have been developed by using augmented belief trees (ABT) [18], an online POMDP solver that computes and updates motion policies in real time. Similarly, POMDPs have also been used to solve multi-objective UAV tasks, such as collision avoidance, path planning, and multiple target tracking [19,20]. In the SAR domain, autonomous navigation in humanitarian relief operations have also been tested using POMDPs, with most of the implementations achieving experimental results only in simulation [21,22]. A recent research work towards real-world tests shows a UAV motion planner implementation using the SARSOP POMDP solver [23] onboard a small UAV (UAVs with a maximum take-off weight (MTOW) of 13.5 kg [24]). However, the system was only validated using trivial targets and not with humanoid-shaped mannequins [25].

Uncertainty in time-critical UAV operations is not limited to external factors. Internal UAV systems, which also generate uncertainty include noisy camera frames during streaming, low image resolution, suboptimal camera settings, and inaccurate detection outputs from vision-based object detectors. Research works on onboard decision-making in small UAVs for autonomous navigation under object detection uncertainty from Convolutional Neural Network (CNN) models are scarce. Some of the closest approaches are the studies from Sandino et al. [26,27], which describe a framework and POMDP problem formulation for a SAR application in a global navigation satellite system (GNSS)-denied environment using a sub-2 kg UAV. The framework was tested in simulation and with real flight tests indoors, simulating a collapsed building scenario. Another study presented by Chanel et al. [28] shows an autonomous multiple-car detector using a UAV and a POMDP optimised mid-flight. However, there are limitations with reproducibility; relevant experimental details are not disclosed, such as the UAV frame, companion computer, payload, and computer vision algorithms.

This paper presents an autonomous drone motion planning and object finding system under object detection uncertainty for outdoor environments. The UAV exploration and object inspection tasks are mathematically formulated using a POMDP, which models uncertainty with probabilistic distributions. A case study inspired by SAR is illustrated below, where a small UAV is deployed to locate a missing person (victim) in a bushland

region. Two victim locations across a simulated flight area were evaluated to test the system at various levels of occlusion.

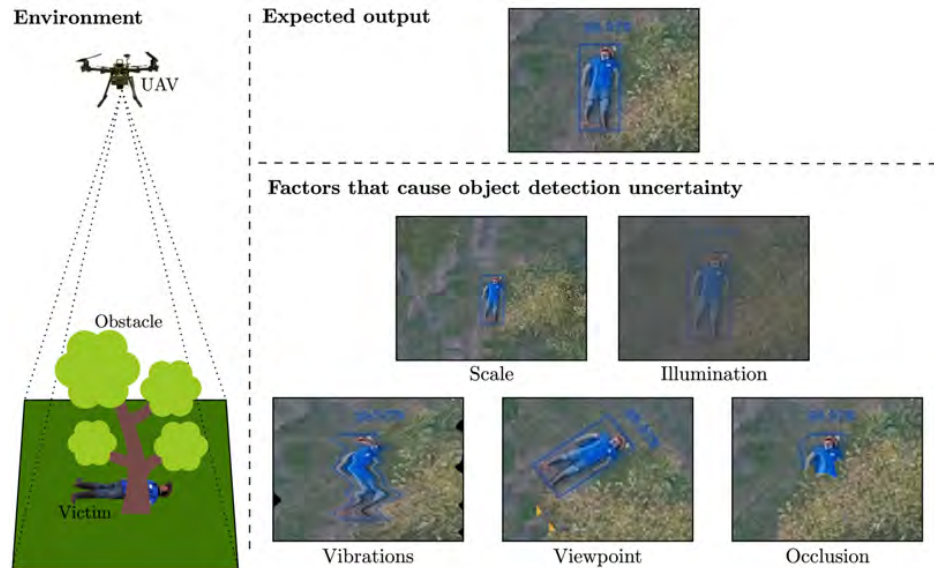


Figure 1. Factors that can cause object detection uncertainty and partial observability during UAV surveys. Incorporating algorithms in decision-making under uncertainty onboard UAVs increase their reliability to explore complex and time-critical environments, such as search and rescue.

This work evaluates the performance of the proposed motion planner by operating the UAV with the following flight modes: (1) mission mode (baseline motion planner), which uses a traditional survey flight plan to explore the area of interest; (2) offboard mode, where the POMDP-based motion planner fully controls the behaviour of the UAV for exploration and inspection of areas from detections of potential victims; and (3) hybrid mode, which uses mission mode to explore the area of interest and triggers offboard mode to inspect dedicated regions from past detections of potential victims. The proposed UAV system design uses hardware in the loop (HIL) to integrate a physical companion computer into the simulation environment. By using HIL to run the decision-making and vision algorithms, it is possible to obtain performance results comparable with those obtained with real flight tests.

2. System Architecture

The system architecture uses a modular design that allocates tasks to dedicated nodes, as shown in Figure 2. This architecture design allows users to collect experimental data in simulated environments to be as close as possible to data collected on real-world tests.

The proposed system architecture categorises its modules into two primary categories, namely the simulator and the companion computer (operating under HIL). The simulator is comprised by items within in the simulated environment including models of the UAV body frame, payloads (i.e., cameras), the victim and relevant obstacles. In addition, the simulator contains plugins to add the UAV's autopilot in the software in the Loop (SIL) mode, and relevant peripheral sensors. The readings of these simulated sensors are used by the autopilot to run its local position estimator based on an extended Kalman filter (EKF). The second category contains the system modules that can be run onboard a UAV's companion computer using HIL, namely, the Computer vision module, decision-making module, mapping module, and motion module. Specific implementations to enable SIL and HIL are covered in detail in Section 4.

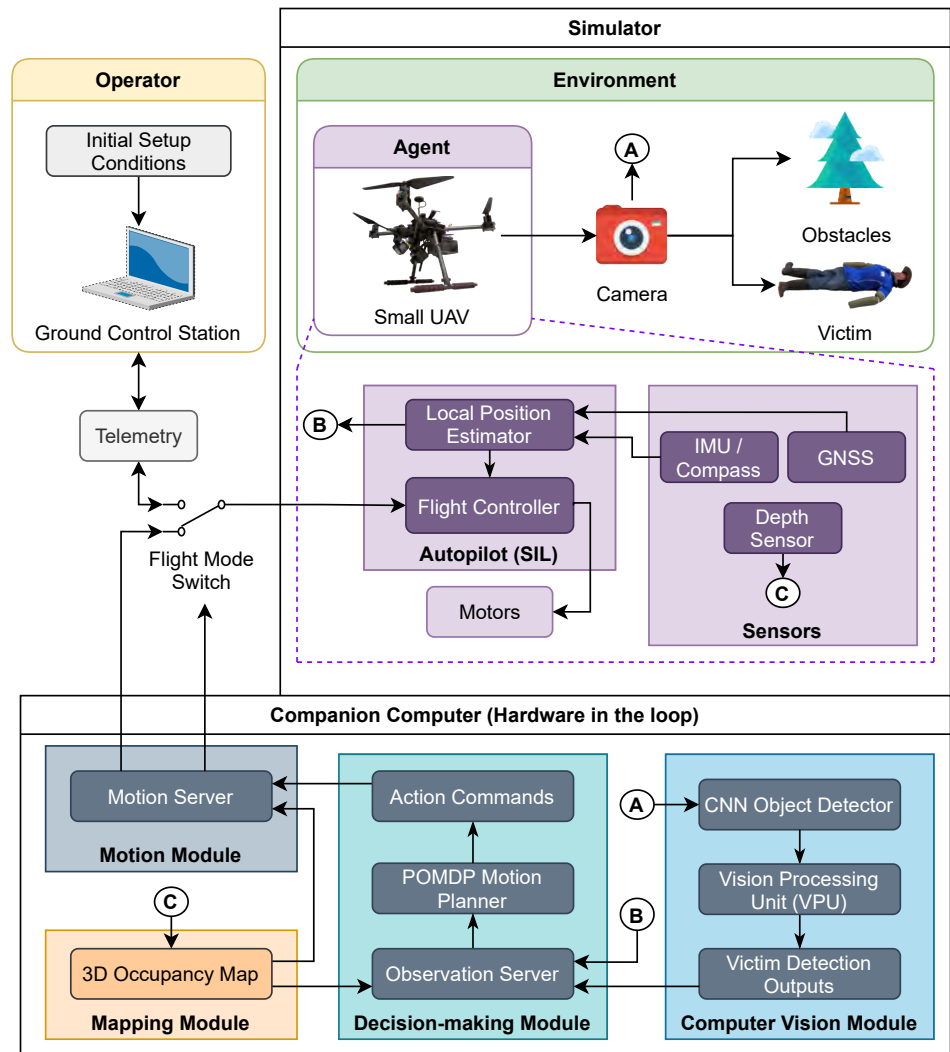


Figure 2. System architecture for applications with rich global navigation satellite system (GNSS) coverage. The environment, UAV, payloads, victim, and obstacles are emulated in a simulator. The computer vision, decision-making, mapping, and motion modules run in hardware in the loop (HIL) using a companion computer.

Each sequential decision process starts by capturing observations of the environment, calculating an action using the motion planner, and then the UAV applying the action. The entire sequence is monitored via telemetry using the ground control station, enabling the operator switch between the autonomous motion planner and manual control if required. Beginning the sequence, environment observations are collected using streamed camera frames, and processed in the computer vision module. This module detects people and their location inside the camera frames with a CNN. As CNNs require mathematical operations, which are computationally expensive in embedded hardware, system resources allocated to run CNN models are delegated to the vision processing unit (VPU). Any positive victim detection outputs are ultimately sent to the decision-making module and managed using the observation server.

The decision-making module is primarily constituted by the POMDP motion planner and the observation server. The observation server handles any data traffic requested by the motion planner, such as local position estimations by the autopilot, any positive

detections by the computer vision module, and 3D occupancy maps by the mapping module. This map is constituted by volumetric occupancy grids and displays the presence and localisation of obstacles in the surveyed environment. In this implementation, the 3D occupancy map is used by the decision-making and motion modules for obstacle avoidance and collision prevention, respectively.

The motion planner features a POMDP to mathematically formulate the sequential decision-making problem. The POMDP-based motion planner outputs action commands to the motion module, which will subsequently translate and send these commands to the flight controller. Lastly, the flight controller adjusts the control signals to the UAV motors to reflect the requested action from the motion planner, and an idle period is granted to the system while the UAV moves to the next location.

3. Motion Planner Design

A decision-making problem for autonomous navigation in UAVs can be formulated as the sequence of actions A that the UAV (or agent) needs to take to accomplish a mission goal. This set of actions are limited to the operational abilities of the UAV within the environment. States S are defined as the set of parameters that outline the system. Every action taken by the UAV causes a change of state $s' \in S$ in the time domain. An illustration of the interaction between the agent and the environment is depicted in Figure 3.

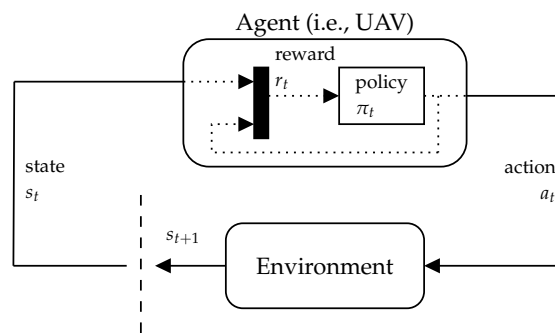


Figure 3. Interaction between the UAV (or agent) and the environment under the framework of Markov decision processes (MDPs).

Each change of state (or action–state chain) is quantified with a reward r . By giving numerical values to every action–state chain, the behaviour of the UAV can be influenced by seeking the highest possible reward when specific desired states are reached (e.g., locating a victim without colliding with nearby obstacles). Simultaneously, action–state combinations that lead to undesirable future states (e.g., colliding with nearby obstacles, or flying beyond the survey geofence) could be quantified with a penalty (or negative reward). Therefore, these problems can be solved by finding the optimal sequential set of actions that a UAV should take to obtain the highest possible accumulated reward as quick as possible [29].

Data received by UAV of the environment (or observations) from its sensors and processed vision outputs (i.e., CNN model detections) in real-world applications are, unfortunately, noisy, incomplete or inaccurate. These imperfections restrict the inference of actual system states, which have inspired the mathematical formulation of POMDPs [30].

This work uses a POMDP to compute a motion policy, which outputs low-level action commands derived using environment observations. A POMDP [31] is a tuple $\langle A, S, O, T, \mathcal{Z}, R, b_0, \gamma \rangle$, where A is a finite set of UAV actions, S is a finite set of states, and O is a finite set of observations. Each time the UAV takes an action $a \in A$ from a state $s \in S$, the probability of moving to a new state $s' \in S$ is defined by a transition function $T(s, a, s') = \mathbb{P}(s' | s, a)$. After an action is taken, the UAV receives an observation $o \in O$ followed by an observation function $\mathcal{Z}(s', a, o) = \mathbb{P}(o | s', a)$. Lastly, the model quantifies

every decision chain (or action taken $a \in A$ from state $s \in S$) with a reward r , calculated using the reward function $R(a, s)$.

In real-world applications, UAVs are limited to capture partial information about the surveyed environment (and the objects to be detected). A POMDP models the uncertainty given by partial observability of its observed states using a probability distribution over the system states. This modelling is known as the belief b , defined as follows:

$$b(H) = \mathbb{P}(s^1 | H), \dots, \mathbb{P}(s^n | H) \quad (1)$$

$$H = a_0, o_1, r_1, \dots, a_{t-1}, o_t, r_t \quad (2)$$

where H is the history of actions, observations, and rewards the UAV has experienced until a time step t .

A POMDP solver starts planning from an initial belief b_0 , which is usually generated using the initial conditions (and assumptions) of the flight mission. When an observation is made, the belief distribution is subsequently updated. The motion policy π of the UAV is represented by mapping belief states to actions $\pi : b \rightarrow A$. An illustration of a belief tree and motion policy in POMDPs is shown in Figure 4.

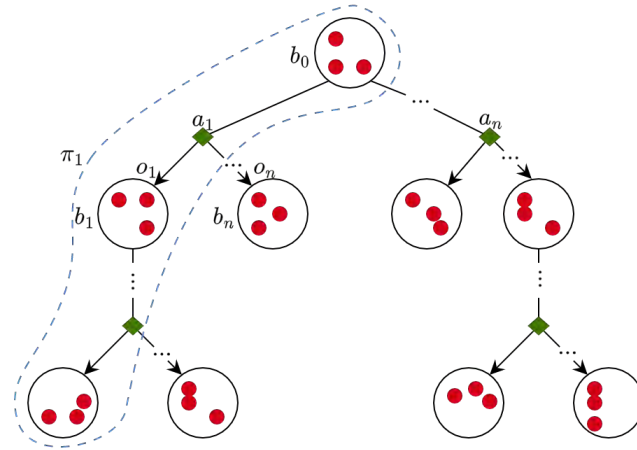


Figure 4. Belief tree and motion policy representation in partially observable MDPs (POMDPs). Each circle illustrates the system belief as the probability distribution over the system states (red dots). When an observation is made, the belief distribution is subsequently updated.

The solution of a POMDP is obtained after finding the optimal policy π^* , which is defined as follows:

$$\pi^* := \arg \max_{\pi} \left(\mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R(S_t, \pi(b_t)) \right] \right), \quad (3)$$

where $\gamma \in [0, 1]$ is the discount factor and defines the relative importance of immediate rewards compared to future rewards.

The solver selected in this work uses augmented belief trees (ABT) [18], which reduce computational demands by reusing past computed policies and updating the optimal policy if changes to the POMDP model are detected. Furthermore, formulated problems with ABT allow declaring continuous values for actions, states, and observations. The following subsections detail the problem assumptions, and the components of the POMDP tuple for exploration and object detection applied on multi-rotor UAVs.

3.1. Assumptions

The following assumptions are applied to the problem formulation for exploration and object detection (i.e., victims) using multi-rotor UAVs:

- An initial 3D occupancy map of the environment is pre-loaded to the UAV before taking-off (further details on how the map is generated can be found in Section 4.4).
- Observations come from real-time streaming of processed camera frames (from the computer vision module), the 3D occupancy map, and the estimated local UAV position from the autopilot.
- The problem formulation is constrained to detect only a single, static victim. However, multiple victims located in the search area could be detected if the motion planner is launched multiple times (see Section 4.5 for further details).
- The motion planner starts once the UAV reaches the initial position waypoint of the flight survey.
- The planner concludes computing motion policies when either: (1) the UAV detects a victim with a high confidence value that exceeds a given threshold (covered in Section 3.7); (2) the UAV covers the entire exploration extent of the search area without finding any victims; or (3) the UAV exceeds a maximum flight timeout.

3.2. Action Commands

This implementation proposes seven low-level UAV actions, which produce position changes in at least one of axes of the world coordinate frame as shown in Table 1. This design allows adding more actions if the application requires it. Possible scenarios include actions to set pitch and roll angles of a camera gimbal, the heading of the UAV (yaw) and actions applied to more than one Cartesian axis (e.g., up and forward, forward and right, etc). Other UAV actions that are executed out of the scope of the POMDP motion planner, such as take-off, return to launch, and land are instead managed by the autopilot.

Table 1. Set of action commands comprised of local position commands referenced to the world coordinate frame. The system keeps records of its current local position at time step k , and calculates δ as the change of position magnitude of coordinates x_u , y_u and z_u from time step k to time step $k + 1$.

$a(k) \in A$	$x_u(k + 1)$	$y_u(k + 1)$	$z_u(k + 1)$
Forward	$x_u(k) + \delta_x$	$y_u(k)$	$z_u(k)$
Backward	$x_u(k) - \delta_x$	$y_u(k)$	$z_u(k)$
Left	$x_u(k)$	$y_u(k) + \delta_y$	$z_u(k)$
Right	$x_u(k)$	$y_u(k) - \delta_y$	$z_u(k)$
Up	$x_u(k)$	$y_u(k)$	$z_u(k) + \delta_z$
Down	$x_u(k)$	$y_u(k)$	$z_u(k) - \delta_z$
Hover	$x_u(k)$	$y_u(k)$	$z_u(k)$

Each time the UAV takes an action, a change δ in UAV local position coordinates is applied. The magnitude for δ_x and δ_y is variable, and its value is defined by the desired overlap of collected camera frames between time steps, as shown below:

$$\delta = l_{\text{FOV}}(1 - \lambda) \quad (4)$$

where l_{FOV} is the length of the projected camera's field of view (FOV), and $\lambda \in [0, 1)$ is the desired overlap value. High values of λ result in smoother motion response and conservative detection of victims in the scene, whereas low values lead to more aggressive UAV response. A detailed description of FOV estimations from vision-based sensors can be found in Section 3.7.

3.3. States

In POMDP formulation for autonomous UAV path-planning, the defined states serve two goals: (1) to model the motion dynamics of the UAV; and (2) to oversee key system

conditions during the flight mission. A system state $s \in S$ for autonomous UAV exploration and object detection is defined as:

$$s = (p_u, f_{\text{crash}}, f_{\text{roi}}, f_{\text{dct}}, p_v, c_v) \quad (5)$$

where p_u is the position of the UAV in the world coordinate frame, f_{crash} is a flag raised when the UAV crashes with an obstacle, f_{roi} is a flag indicating whether the UAV flying beyond the flying limits, f_{dct} is the flag raised if a potential victim is detected by the UAV. If $f_{\text{dct}} = \text{True}$, the position of the victim in the world coordinate frame is given in p_v , with $c_v \in [0, 1]$ providing the corresponding detection confidence. The system reaches a terminal state whenever $c_v \geq \zeta$, where ζ is the confidence threshold. The formulated state space is flexible and can be further extended for two or more victims, as well as additional system conditions if required.

3.4. UAV Motion Model

The transition from current to new states is defined through a simplified version of the motion dynamics of a multi-rotor UAV as follows:

$$p_u(k+1) = p_u(k) + \Delta p_u(k) \quad (6)$$

where $p_u(k)$ is the position of the UAV at time step k , and $\Delta p_u(k)$ is the change in the UAV's position from time step k to time step $k+1$. While this implementation does not incorporate the UAV Euler yaw angles in the action space, Equation (6) can be further expanded by including the multi-rotor rotation matrix [32] as presented in the problem formulation given in [26].

The procedure to model $\Delta p_u(k)$ follows the system identification process presented in [27]. The primary workflow is defined as follows:

1. Track the position responses of the UAV controlled system using a series of step setpoints via a Vicon motion capture system (Vicon, Oxford, UK).
2. Estimate the transfer function of the UAV controlled system using the MATLAB system identification toolbox.
3. Discretise the transfer function using the Tustin approximation method.
4. Obtain the difference equation using the inverse Z transform.

The generic difference equation is defined as:

$$y(k) = \sum_{i=0}^N a_i r(k-i) - \sum_{i=1}^N b_i y(k-i), \quad (7)$$

$$\Delta p_u(k) = y(T_s^{-1}) - y(0), \quad (8)$$

where $y(k)$ is the response of the system, N is the order of the transfer function, a_i and b_i are the numerical constants for each setpoint and response function variables respectively, and T_s is the sampling period. The constants used for the experiments are given in Appendix A.

3.5. Reward Function

Any given reward r to the UAV after taking an action $a \in A$ from state $s \in S$ is calculated using a reward function $R(a, s)$. This function design allows multi-objective task definition, and critically influences the UAV behaviour during flight missions. This work proposes Algorithm 1 as a novel algorithm for $R(a, s)$, which incorporates obstacle avoidance, exploration, and object detection. The proposed algorithm contains several reward variables, whose values are set following an iterative process of the system in the simulator to observe the combination that provides the best behaviour of the UAV to explore the environment and inspect areas with potential victims inside them. Table 2 shows the converged reward values for this implementation.

Algorithm 1 Reward function R for exploration and object detection.

```

1:  $r \leftarrow 0$ 
2: if  $f_{\text{crash}}$  then
3:    $r \leftarrow r_{\text{crash}}$  ▷ UAV crashing cost
4: else if  $f_{\text{roi}}$  then
5:    $r \leftarrow r_{\text{out}}$  ▷ Beyond safety limits cost
6: else if  $f_{\text{det}}$  then
7:    $r \leftarrow r_{\text{dte}}$  ▷ Detected object reward
8:    $r \leftarrow r + \left[ r_{\text{dte}} \cdot \left( 1 - \frac{z_u - z_{\text{min}}}{z_{\text{max}} - z_{\text{min}}} \right) \right]$  ▷ UAV altitude reward
9:   if  $c_v \geq \zeta$  and  $a = \text{Down}$  then
10:     $r \leftarrow r + r_{\text{conf}}$ 
11:   end if
12: else
13:    $r \leftarrow r_{\text{action}}$  ▷ Action cost
14:    $r \leftarrow r - \left[ r_{\text{dte}} \cdot \left( 1 - \frac{z_u - z_{\text{min}}}{z_{\text{max}} - z_{\text{min}}} \right) \right]$  ▷ UAV altitude cost
15:    $r \leftarrow r - \left[ r_{\text{dte}} \cdot \left( 1 - 0.5^{4 \cdot d_v / d_w} \right) \right]$  ▷ Horizontal distance cost
16:    $r \leftarrow r + r_{\text{fov}} \cdot \varepsilon$  ▷ Footprint overlap cost
17: end if
18: return  $r$ 

```

Table 2. Applied reward values to the reward function R , defined in Algorithm 1.

Variable	Value	Description
r_{crash}	-50	Cost of UAV crash
r_{out}	-25	Cost of UAV breaching safety limits
r_{dte}	+25	Reward for detecting potential victim
r_{conf}	+50	Reward for confirmed victim detection
r_{action}	-2.5	Cost per action taken
r_{fov}	-5	Footprint overlapping cost

The reward function R first evaluates any states that will negatively affect the integrity of the UAV. R returns negative reward values if a given action a provokes a crash or lets the UAV fly beyond the safety limits regardless whether the UAV detects a potential victim. This modelling encourages the UAV to apply a policy that approaches a victim without colliding with an obstacle in the process. If the UAV detects a potential victim (Step 6), R calculates a linear function (Step 8), which returns increased values as the UAV gets closer to the minimum allowed altitude. Once the confidence value c_v from the CNN object detector surpasses a pre-defined threshold, a bigger reward is returned as the potential victim is assumed to be found (Step 9 and 10).

In case there are no detections, R applies a set of cost functions to encourage a greedy horizontal exploration of the environment. The first is the inverse function from (Step 8) to encourage the UAV to explore at the maximum possible altitude to maximise the camera's footprint extent. The second is an exponential function (Step 14), which assigns lower cost values the closer the UAV is to the victim. This function calculates the Manhattan distance between the UAV and the victim d_v and the maximum exploration distance d_w , which are defined as follows:

$$d_v = \sum_{i=1}^n |p_i - q_i|, p_i = (x_u, y_u), q_i = (x_v, y_v) \quad (9)$$

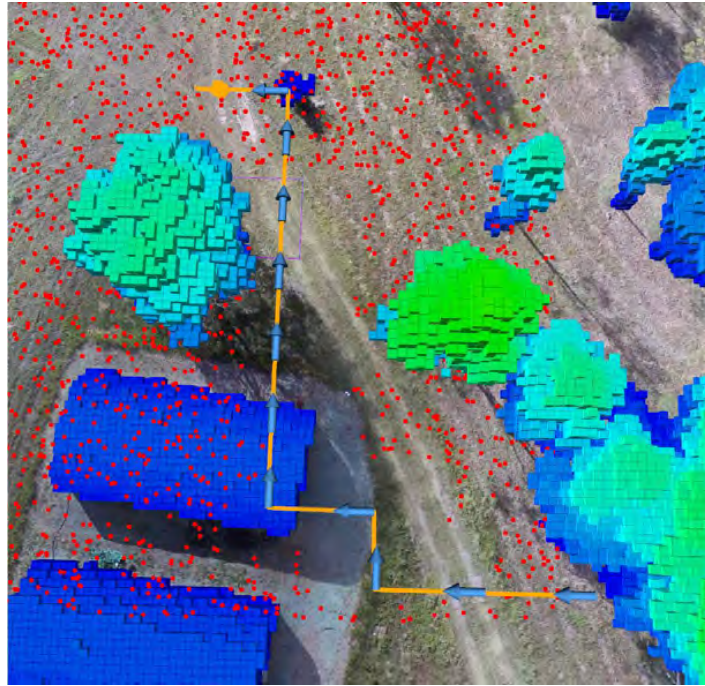
$$d_w = \sum_{i=1}^n |p_i - q_i|, p_i = (x_{\max}, y_{\max}), q_i = (x_{\min}, y_{\min}) \quad (10)$$

Sub-optimal policies could be generated if a set of distributed victim position particles are equidistant from the UAV by using Equation (10). Moreover, redundant traversed paths have been reported in the literature that implemented similar problem formulations [16,27]. The third cost function (Step 16) counteracts the above-mentioned effects by logging the traversed path by the UAV and introducing the concept of exploration footprint. The concept records the accumulated visual footprint of the downward-looking camera into a two-dimensional (2D) map. The map extent equals the extent limits of the surveyed area, and it is updated after calculating the camera's footprint once the UAV collects a new observation. An illustration of the concept is shown in Figure 5.

The overlap ε between the camera's current footprint and its correspondent location in the footprint map is defined as follows:

$$\varepsilon = \frac{\sum_{i=1}^n F_i(p_u)}{n}, \varepsilon \in [0, 1] \quad (11)$$

where $F_i(p_u)$ are the pixel values of the projected FOV in the footprint map, and n is the total number of projected pixels in the footprint. A maximum overlap value of 1 indicates that such action will place the UAV to a fully previously explored area, and, as indicated in Step 16 of Algorithm 1, the whole penalty value r_{fov} will be added to the reward. A minimum value of 0 means that a given action will place the UAV in an unexplored area and no penalty will be added to the reward. Intermediate values of ε represent partial overlapping, adding a partial penalty value r_{fov} to the reward.



(a)

Figure 5. Cont.



Figure 5. Example of a traversed path by the UAV and its corresponding footprint map. (a) Traversed path of the UAV (orange splines) in a simulated environment. The arrows indicate actions taken by the UAV per time step. Environment obstacles are depicted with blue and green blocks, and the believed victim location points are shown using a red point cloud. (b) Accumulated traced footprint using the extent of the camera’s FOV. Explored areas are depicted in black and unexplored areas in white. The higher the overlap of explored areas after taking an action $a \in A$, the higher the footprint overlap cost.

3.6. Observations

A captured observation $o \in O$ —data from UAV sensors about the state of the environment and the UAV itself (illustrated in Figure 2)—is defined as follows:

$$o = (o_{p_u}, o_{\text{dte}}, o_{p_v}, o_{\zeta}, o_{\text{obs}}), \quad (12)$$

where o_{p_u} is the UAV position calculated by the local position estimator of the autopilot, o_{dte} is the flag triggered by the very first detection of a potential victim by the CNN object detector, o_{p_v} and o_{ζ} are the local positions of the victim and the detection confidence, respectively, both of them defined only if there are any positive detections, and o_{obs} is the flag triggered after processing the 3D occupancy map for any obstacles located in front of the UAV. The detection confidence o_{ζ} , defined in Equation (13), is a summary statistic that measures the frequency of positive detections between the last two observation calls:

$$o_{\zeta} = \frac{\sum_{i=1}^n o_{\text{dte}_i}}{n}, \quad (13)$$

where n is the number of processed image frames between observation calls, and o_{dte} is the flag indicating a positive detection per processed frame i .

3.7. Observation Model

An advantage of ABT compared to other solvers is that the probabilistic transition functions T and Z do not require to be explicitly defined, but instead it uses a generative

model. This model generates T and \mathcal{Z} using a modelled observation o given an action a and the next state s' . In this implementation, the variables contained in the generative model are the local position of the UAV s'_{p_u} , the local position of the victim s'_{p_v} , and the detection confidence o_c . The estimation of s'_{p_u} follows the UAV motion estimations detailed in Section 3.4.

Potential victim detections and their subsequent positioning estimations are conditioned by the camera pose at the UAV frame and its projected footprint of the environment. Specifically, if the 2D local position coordinates of the victim $s'_{p_v}(x, y)$ are within the projected footprint limits of the camera, the victim is assumed to be detected. This estimation is done by calculating the sum of angles between a 2D point (i.e., s'_{p_v}) and each pair of points that constitute the footprint boundaries (the footprint rectangular corners) [33]. The 2D projected footprint extent l of a vision-based sensor, illustrated in Figure 6, can be calculated using Equations (14) and (15):

$$l_{\text{top, bottom}} = s'_{p_u}(z) \cdot \tan\left(\alpha \pm \tan^{-1}\left(\frac{h}{2f}\right)\right), \quad (14)$$

$$l_{\text{left, right}} = s'_{p_u}(z) \cdot \tan\left(\pm \tan^{-1}\left(\frac{w}{2f}\right)\right), \quad (15)$$

where s'_{p_u} is the UAV altitude, α and β are the camera's pointing angles from the vertical z and horizontal x axis of the world coordinate frame, w is the lens width, h is the lens height, and f is the focal length.

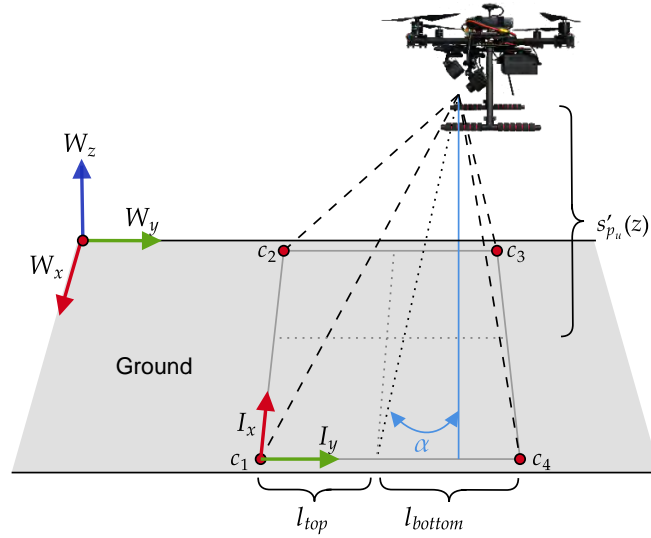


Figure 6. Field of view (FOV) projection and footprint extent of a vision-based sensor. The camera setup on the UAV frame defines α as the pointing angle from the vertical (or pitch) and determines the coordinates of the footprint corners c .

The footprint corners c from the camera's local coordinate frame I are translated to the world's coordinate frame W using the following transformation:

$$\begin{bmatrix} c'(x) \\ c'(y) \end{bmatrix} = \begin{bmatrix} s'_{p_u}(x) \\ s'_{p_u}(y) \end{bmatrix} + \begin{bmatrix} \cos(\varphi_u) & -\sin(\varphi_u) \\ \sin(\varphi_u) & \cos(\varphi_u) \end{bmatrix} \begin{bmatrix} c(x) \\ c(y) \end{bmatrix}, \quad (16)$$

where s'_{p_u} is the next UAV position state, and φ_u is the Euler yaw angle of the UAV. However, as no actions involve adjusting the heading of the UAV mid-flight, and assuming yaw estimation errors are negligible, Equation (16) is simplified as follows:

$$\begin{bmatrix} c'(x) \\ c'(y) \end{bmatrix} = \begin{bmatrix} s'_{p_u}(x) + c(x) \\ s'_{p_u}(y) + c(y) \end{bmatrix}. \quad (17)$$

The detection confidence o_ζ that comes as part of the output data from the CNN object detector is modelled using Equation (18):

$$o_\zeta = \frac{(1 - \zeta_{\min})(d_{uv} - z_{\min} + \zeta_{\min})}{z_{\max} - z_{\min}}, \quad (18)$$

where ζ_{\min} is the minimally accepted confidence threshold, z_{\max} and z_{\min} are the maximum and minimum UAV flying altitudes respectively, and d_{uv} is the Manhattan distance between the UAV and the victim.

4. Experiments

The proposed system is evaluated under the context of SAR to autonomously scout a bushland area and find a lost person (victim). The following subsections describe the environment setup, victim locations, hardware and software used, and tested flight modes to accomplish the mission. The experiments assume the environmental terrain and 3D occupancy map have been generated prior to launching the UAV to survey the area.

4.1. Environment Setup

The system was tested under a simulated replica of the Queensland University of Technology (QUT) Samford Ecological Research Facility (SERF), located at 148 Camp Mountain Road, Samford QLD 4520, Australia. SERF is a 51 hectare property comprising a protected dry sclerophyll forest and grazing zones. From this property, a smaller 2.25 hectare of a mostly flat area was chosen for the tests as it contains an environment with a rich combination of open, mixed, and dense tree canopy areas, as depicted in Figure 7a.

The environment was emulated using the Gazebo Robotics Simulator. High-fidelity terrain textures from SERF were generated after projecting a high-resolution georeferenced red, green, blue (RGB) mosaic onto the world's surface, as illustrated in Figure 7b. Imagery datasets for the mosaic are captured using a DJI Zenmuse XT2 camera (DJI, Shenzhen, China), mounted on a DJI M210 V2 (DJI, Shenzhen, China). The UAV collected the data at an above ground level (AGL) height of 60 m, 70% overlap, and 60% sidelap, at a constant speed of 10 m/s. Orthomosaics have been produced from video files by extracting and georeferencing still images, significantly increasing the area coverage rates.

Relevant obstacles in the environment, such as trees and greenhouses, are placed following their corresponding projections in the world's terrain. In addition, the altitude values of each tree in the scene are set following a LiDAR dataset collected from a Hovermap mapper (Emesent Pty Ltd., QLD, Australia), shown in Figure 7c. The dataset has been captured by mounting the Hovermap into a DJI M600Pro (DJI, Shenzhen, China) at an AGL height of 60 m and constant speed of 5.4 m/s.

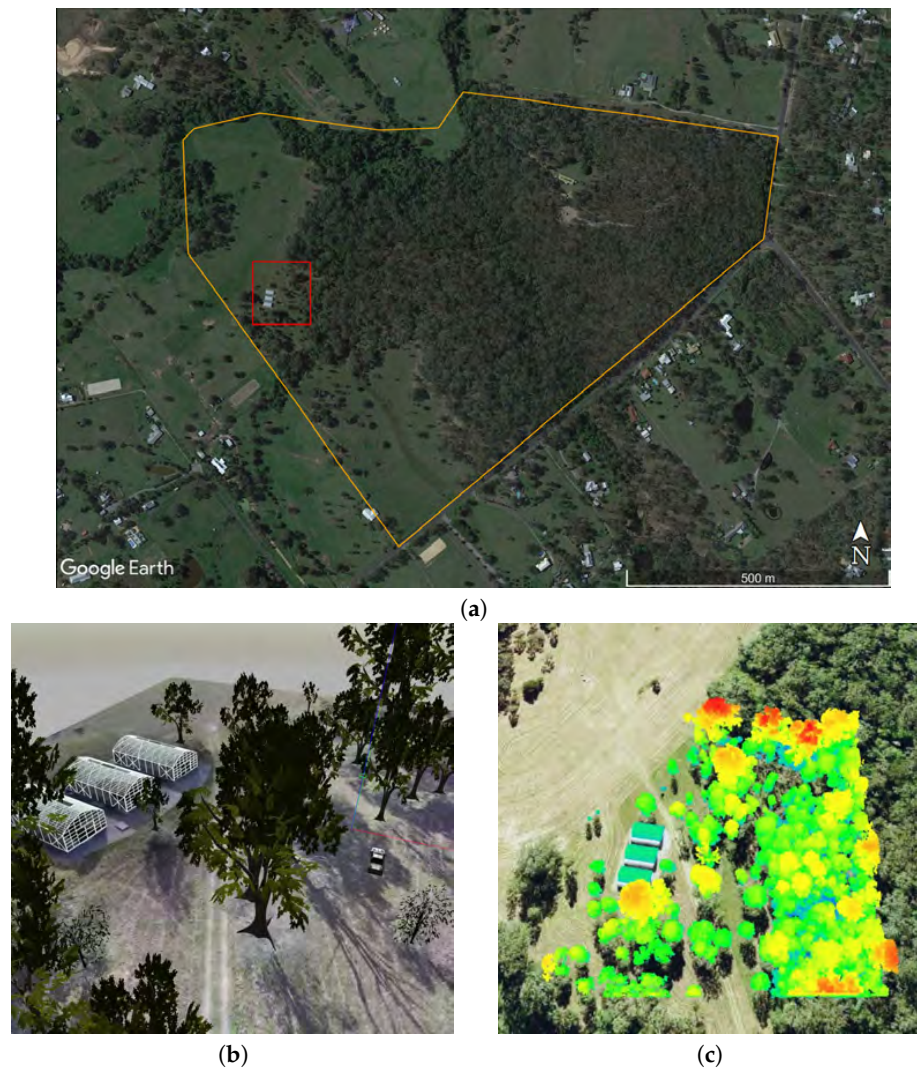


Figure 7. QUT Samford Ecological Research Facility (SERF) virtual environment setup. (a) Property boundaries of SERF (orange) and simulated region of interest (red). (b) Isometric view of the virtual SERF instance in Gazebo using a high-resolution red, green, blue (RGB) mosaic as the terrain texture. (c) Collected georeferenced LiDAR at SERF using a Hovermap mapper (Emesent Pty Ltd., QLD, Australia). LiDAR data are used to set accurate building and tree canopy heights in the simulated environment.

4.2. Victim Locations

Two sets of experiments with a single victim (male adult) at two locations are proposed to validate the system in terms of detection complexity. The first location (L1) setup (latitude 152.87309239; longitude -27.38938274) displays the victim in an area free of obstacles and no visibility challenges from streamed UAV data. The second location (L2) setup (latitude 152.87300287; longitude -27.38966054) features the victim located close to one of the trees where partial or full occlusion may occur as portrayed in Figure 8. For both locations, the victim is static, lying on the ground with a heading of 0° from the north compass.



Figure 8. Top-down visualisation and detection challenges of the victim placed at location L2. (a) Partial occlusion of the victim from the tree branches and leaves. (b) Magnified view of the victim from Figure 8a.

4.3. Hardware

The system architecture shown in this paper (Figure 2) classifies hardware into two primary components: (1) the companion computer, which runs all of the modules as if the system is tested on real flight tests; and (2) the workstation that supports the simulated environment and the SIL implementation of the autopilot.

The companion computer selected for these experiments follows the recommended hardware used by Sandino et al. [27] for onboard navigation using small quadrotor UAVs. The computer vision, decision-making, motion, and 3D occupancy mapping modules run on an UP² (AAEON Technology Inc., New Taipei City, Taiwan). Relevant specifications include a 64-bit quad-core Intel[®] Pentium[®] N4200 processor running at 1.1 GHz, 64 GB eMMC solid state drive, 8 GB DDR3 RAM, four FL110 USB 3.0 connectors, two Ethernet controllers, two high-speed UART controllers, and one mPCIe connector. The mPCIe connector from the UP² is used to plug an AI CORE Movidius Myriad X VPU 2485 and load CNN object detectors and perform real-time inference.

The simulated environment setup, ground control station, and the autopilot (running in SIL mode) are tested under a desktop computer with the following specifications: 64-bit 12-core Intel[®] Core[®] i7-8700 CPU running at 3.2 GHz, 512 GB solid state drive, 32 GB DDR4 RAM, 6 GB NVIDIA GeForce GTX 1060, six (6) USB 3.0 ports, and one Ethernet controller.

4.4. Software and Communications

The presented system applies a set of various software solutions per module. It uses the robot operating system (ROS) [34] as the common middleware for communications and allows the modular design presented in this paper. The software for both the companion computer and desktop runs under Linux Ubuntu 18.04 64-bit operating system.

The CNN model architecture used to detect potential victims in the computer vision module is a Google MobileNet single-shot detector (SSD) [35]. This instance of MobileNet SSD is tuned using pre-trained weights from the PASCAL VOC2012 dataset and deployed using the Caffe framework [36]. The model classifies up to 21 unique objects from the PASCAL dataset, and achieves a mean average precision of 72.7%. The computer vision module only filters and outputs the class *person*, and positive detections with a confidence of at least 60%. The CNN is loaded to the VPU using OpenVINO and OpenCV libraries. The UAV system is not limited to this detector, and other CNN models trained with airborne specific datasets can be implemented instead. Nevertheless, existing limitations from the chosen model (i.e., most of the labelled images were taken from a front-view

camera configuration) aid its intentional usage to demonstrate of the robustness of the motion planner against higher levels of detection uncertainty.

The key nodes of the decision-making module are programmed using the C++ programming language and the TAPIR toolkit [37], which implements the ABT online POMDP solver. The mapping module, which contains 3D occupancy map node for obstacle avoidance, runs under the OctoMap library [38]. A benefit of using OctoMap is the capability of generating and updating occupancy maps from 3D point cloud data such as LiDAR or depth sensors. Therefore, the LiDAR data captured at SERF by Hovermap are loaded to generate an initial occupancy map of the environment, as shown in Figure 9.

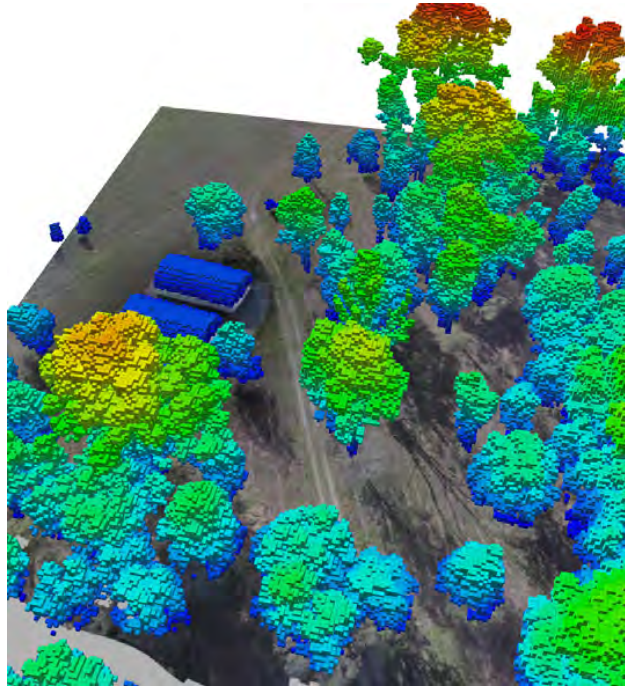


Figure 9. Three-dimensional (3D) occupancy map from the Hovermap LiDAR data of The Samford Ecological Research Facility (SERF). Occupied cells represent the physical location of various tree species and greenhouses present at SERF.

The emulation of the autopilot is performed using the PX4 SIL plugin for Gazebo. This plugin provides a high-fidelity experience of the interface and flight controller behaviour with other modules. The communication interface between the motion module and the PX4 autopilot (with the SIL Gazebo plugin) is based on the second version of the Micro Air Vehicle Link (MAVLink) protocol [39] and applied through MAVROS, a ROS wrapper of MAVLink. Telemetry to the ground control station is achieved using QGroundControl via a UDP connection.

4.5. UAV Flight Modes

Three flight modes are proposed to evaluate the system at each victim location mentioned in Section 4.2. From a SAR application context, the presumed location of the victim is unknown. Hence, the flight modes are designed to scout the entire delimited flying area.

4.5.1. Mission Mode

The first flight mode, denominated here as *mission* mode, is intended to be served as the UAV motion planner baseline for this study. This flight mode uses the traditional survey approach provided by many modern flight controllers. Typically, a sequential list of position waypoints that follows the desired survey pattern by the pilot are generated

by a ground control station software and sent to the UAV autopilot. The flight plan to assess this mode follows the flight parameter values from Table 3, and is compiled using QGroundControl, as shown in Figure 10. This flight mode is triggered with the UAV on the ground and allows the UAV to take off, reach the initial position waypoint, and perform the survey. The mode terminates after completing the survey and manoeuvring the UAV back to its launch position.

Table 3. Flight plan parameters for mission mode.

Property	Value
UAV altitude	20 m
UAV velocity	2 m/s
Camera lens width	1.51 mm
Camera lens height	1.13 mm
Camera focal length	3.6 mm
Image resolution	640 by 480 px
Overlap	30%
Bottom right waypoint	$-27.3897972^\circ, 152.8732300^\circ$
Top right waypoint	$-27.3892651^\circ, 152.8732300^\circ$
Top left waypoint	$-27.3892593^\circ, 152.8728180^\circ$
Bottom left waypoint	$-27.3897858^\circ, 152.8728180^\circ$

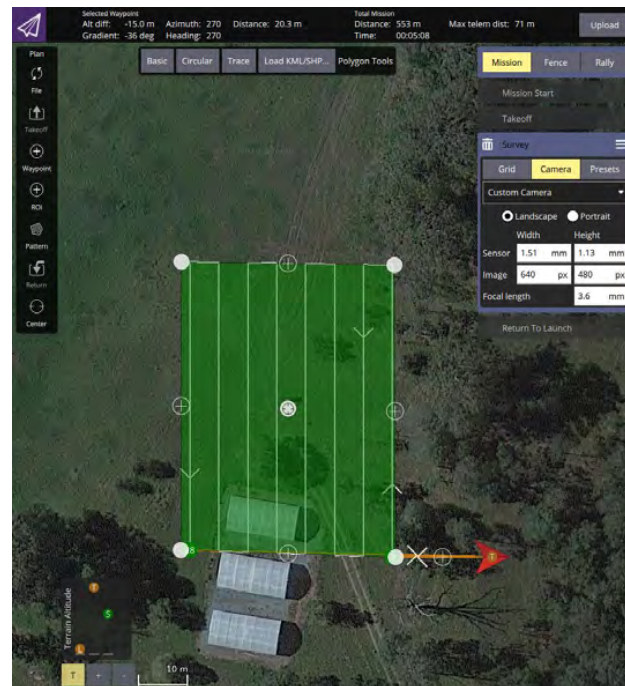


Figure 10. Designed flight plan that the UAV follows in mission mode. The survey is set for a single pass of the setup at a constant height of 20 m, UAV velocity of 2 m/s, and overlap of 30%.

4.5.2. Offboard Mode

The second flight mode, denominated here as *Offboard* mode, runs the POMDP-based motion planner described in Section 3. In concordance to the flight plan design from mission mode, the initial conditions set for the offboard mode and defined variables from Section 3.4 are detailed in Table 4.

Table 4. Initial conditions set to the POMDP motion planner.

Variable	Description	Value
z_{\max}	Maximum UAV altitude	21 m
z_{\min}	Minimum UAV altitude	5.25 m
p_{u0}	Initial UAV position	(-27.3897972° , 152.8732300° , 20 m)
φ_u	UAV Heading	0°
δ_z	UAV climb step	2 m
λ	Frame overlap	30%
α	Camera pitch angle	0°
ζ_{\min}	Minimum detection confidence	30%
ζ	Confidence threshold	85%
γ	Discount factor	0.95
Δt	Time step interval	4 s
t_{\max}	Maximum flying time	8 min

Before the planner is run at the initial UAV location p_{u0} , TAPIR—the toolkit the planner is coded—runs an offline policy solver for two seconds. Then, for every time step, an observation is collected and TAPIR triggers the solver online to update the motion policy with a timeout of 800 ms. The remaining time is an idle period while the UAV moves to the next position following the action taken from the updated policy. Compared to the first flight mode, this flight mode finishes once the maximum flight time is exceeded or if the victim is found (i.e., $\zeta \geq 85\%$).

The initial state belief b_0 of the POMDP model consists of the position of UAV and the victim. The UAV position belief follows a normal distribution with a mean of p_{u0} and standard deviation of 1.5 m. The victim position belief follows a uniform distribution whose limits are equivalent to the survey extent of the flight plan from mission mode, as shown in Figure 11.

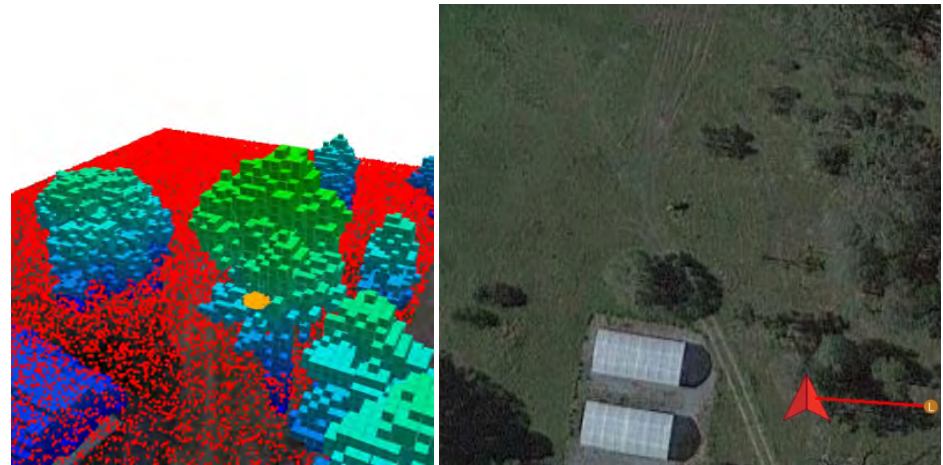


Figure 11. Probabilistic distribution of initial UAV and victim position belief across the survey extent. The UAV position belief (orange points) follows a normal distribution with mean equals to p_{u0} and standard deviation of 1.5 m. The victim position belief (red points) follow a uniform distribution, whose limits are equivalent to the survey extent of the flight plan from mission mode.

4.5.3. Hybrid Mode

The third flight mode, known as *hybrid* mode, combines the methods from mission and offboard modes. This mode splits the overall navigation task into two components: it uses mission mode to skim the survey area and triggers the POMDP motion planner after a first positive detection. The primary difference in the design of the POMDP motion

planner between the offboard and hybrid modes relies on the modelling of the initial belief b_0 . In a hybrid mode, the mean value of the normal distribution of the UAV position belief is now defined as the UAV position coordinate, where the first victim detection takes place. Instead of distributing the victim position belief uniformly across the entire flying area, this mode constraints the search area to the extent of the camera's FOV, as shown in Figure 12.

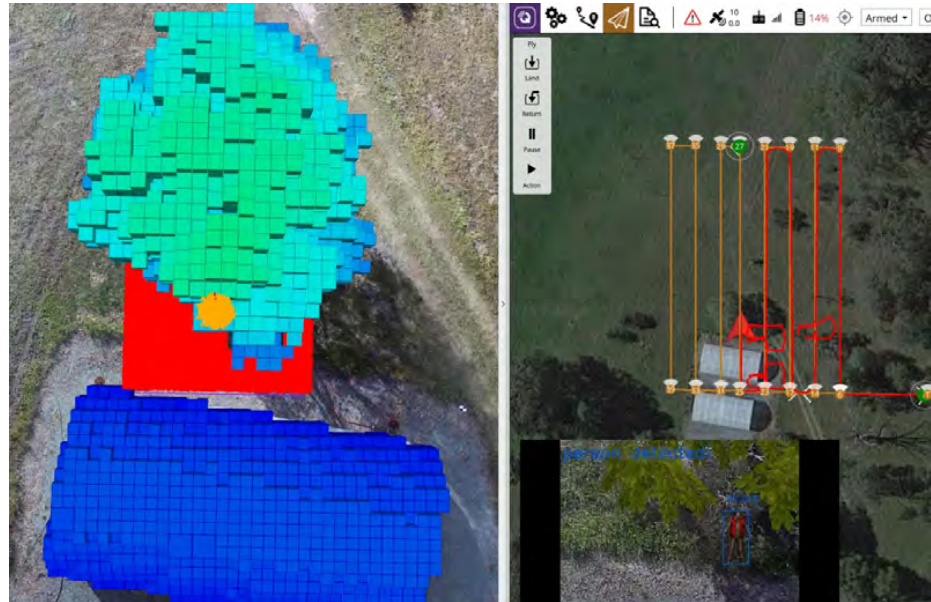


Figure 12. Probabilistic distribution of initial UAV and victim position belief under hybrid flight mode. The UAV position belief (orange points) follows a normal distribution, with a mean equals to the current UAV position where a potential victim is first detected, and standard deviation of 1.5 m. The victim position belief (red points) follow a uniform distribution whose limits are equivalent to the extent of the camera's FOV.

5. Results

Validation of the system through the presented SAR case study contains the following performance metrics: (1) Heatmaps of recorded GNSS coordinates of any detected victims; (2) accuracy of output detections and collisions; (3) speed analysis of POMDP-based motion planner in offboard and hybrid flight modes; and (4) system speed to cover the survey area or until the victim is detected. Each performance indicator is evaluated per flight mode and victim locations (which represent various detection complexities because of occlusion from nearby obstacles). Each setup combination (flight mode and victim location) was tested for 20 iterations. Demonstrative videos of the UAV system operating under mission (Video S1), offboard (Video S2), and hybrid (Video S3) flight modes can be found in the Supplementary Materials section.

A series of heatmaps of GNSS coordinates (from detected victims) returned by mission, offboard and hybrid flight modes are shown in Figures 13–15 respectively.

The distribution of GNSS coordinates in the heatmaps suggests a considerably higher precision when the system runs offboard and hybrid flight modes to report victim locations compared to the baseline (mission) flight mode. The visualisation complexity from the defined victim locations does not indicate any significant differences to the precision outputs of the system. The variable distribution of victim GNSS coordinates in mission mode (Figure 13) was caused by false positive readings from specific object visualisations. For this case study and environment setup, other objects wrongly detected as people included trees, greenhouse rooftops, and to a lesser degree, bare grass. Examples of these detections are displayed in Figure 16.



Figure 13. Heatmaps from recorded GNSS coordinate points from detections in mission mode. The amount of false positive locations were triggered by the CNN model from detecting other objects as people during the survey. (a) First victim location (blue dot) in an open area. (b) Second victim location (blue dot) nearby a tree.



Figure 14. GNSS coordinates' heatmaps of confirmed victim detections by the POMDP-based motion planner in offboard flight mode. (a) First victim location (blue dot) in an open area. (b) Second victim location (blue dot) nearby a tree.



Figure 15. GNSS coordinates' heatmaps of confirmed victim detections by the POMDP-based motion planner in hybrid flight mode. (a) First victim location (blue dot) in an open area. (b) Second victim location (blue dot) nearby a tree.

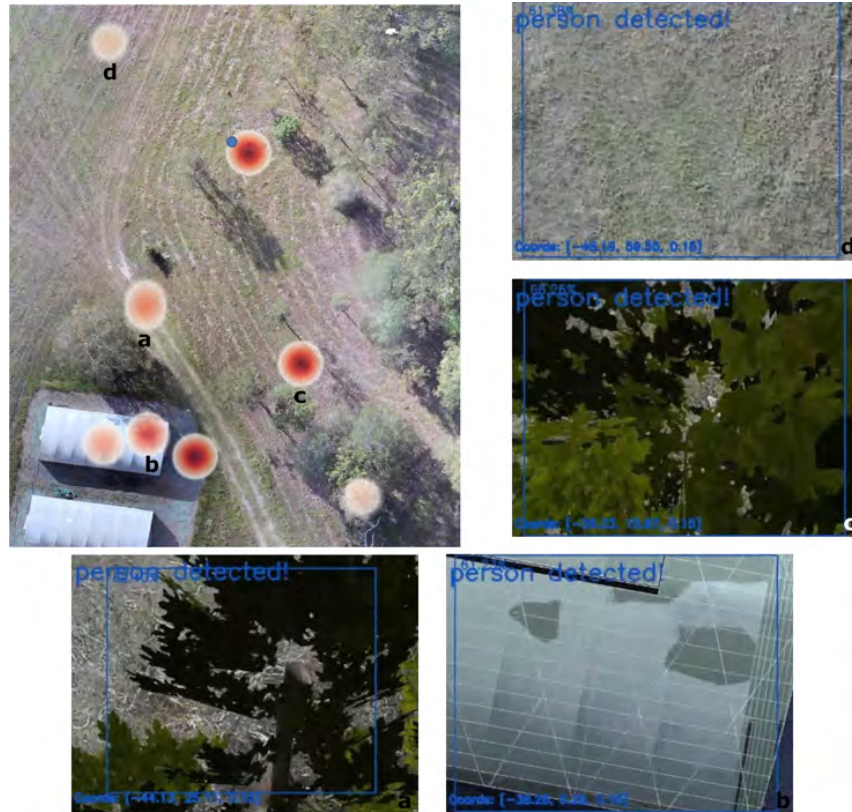


Figure 16. Most common object visualisations that triggered false positive readings of people in mission mode. Other objects wrongly detected as people included trees, greenhouse rooftops, and to a lesser degree, bare grass.

The detection outputs of the CNN model also had an impact in the accuracy metrics of the system when surveying the studied area in mission mode. A summary of the accuracy and collision metrics per flight mode and victim location is shown in Table 5.

Table 5. Accuracy and collision metrics of the system to locate a victim at two locations (L1 and L2). Each setup was evaluated for 20 iterations.

Setup	Detections (%)	Misses (%)	Non-Victims (%)	Collisions (%)
Mission (L1)	20.0	0.0	80.0	0.0
Mission (L2)	17.4	0.0	82.6	0.0
Offboard (L1)	100.0	0.0	0.0	0.0
Offboard (L2)	100.0	0.0	0.0	0.0
Hybrid (L1)	95.0	0.0	0.0	5.0
Hybrid (L2)	90.0	5.0	5.0	0.0

Regardless of the victim location in the surveyed area, the use of the MobileNet SSD CNN model had a negative impact on any flights using mission mode. After testing each setup combination for 20 iterations, the positive detection rate for victims in a trivial location (or Location 1 (L1) as referred in Table 5) did not exceed 20% of the reported detections. In contrast, offboard and hybrid flight modes reported improved detection rates of 100% and 92.5% respectively. An overall slight decrease in the victim detection rates was observed between trivial and complex locations, where up to 5.0% of the iteration runs failed to confirm a victim's location in hybrid mode. The tests reported that 2.5% of

the iterations in hybrid mode (5.0% for Location 1) with the UAV colliding with obstacles. However, taking into account all of the tested runs from offboard and hybrid modes, the overall collision probability experienced using the POMDP-based motion planner is 1.25%.

A speed analysis of the evaluated flight modes tracked the elapsed time from the initial position waypoint until a victim detection was confirmed. The collected data are represented with box plots, as shown in Figure 17.

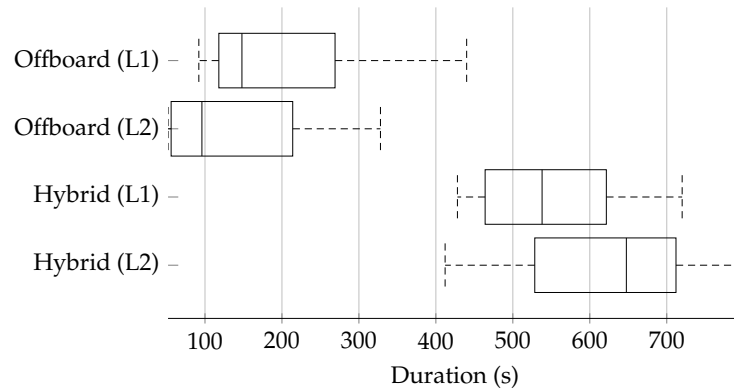


Figure 17. Distribution frequency of the elapsed time until the POMDP-based motion planner confirmed a detected victim in a trivial (L1) and complex (L2) locations.

The data distribution shows a wider range in the top quartile and top whisker on the conducted tests using offboard mode for both victim locations. This distribution anomaly was caused by increased uncertainty given to the solver to compute a motion policy for a higher extent of area to survey compared to hybrid mode. Similarly, a bigger search area resulted in increased combinations of traversed paths to survey and locate the victim. Recorded paths indicated a trend of the solver to navigate towards the centre of the area extent, surrounding trees, as shown in Figure 18a.

Data distribution and traversed paths on tests under hybrid flight mode were more consistent as the exploration extent is constrained to the lawnmower pattern followed in mission mode. However, elapsed times in hybrid mode were considerable higher at both victim locations because of the occurrences of false positives. As first detection readings trigger offboard mode to inspect such locations, the system required additional time to confirm or discard the presence of any victim. An illustration of the traversed path by the UAV in hybrid flight mode is shown in Figure 18c.

The average elapsed time to locate the victim in locations 1 and 2, shown in Table 6, indicates contrasting values between offboard and other flight modes. A trend to detect a victim at location 2 quicker than at location 1 was caused by the proximity of the initial position waypoint to begin the search (bottom right corner of the survey pattern). Overall, the system required 2.3 times more time to survey the flight area in hybrid mode than in mission mode. Nonetheless, using tuned CNN models from UAV image datasets is expected to decrease the instances of false positive readings as it was the case in this work. As the primary contribution of this work is on presenting a fully autonomous robust motion planner system against high levels of object detection uncertainty and partial observability, a pre-trained detector from a PASCAL VOC2012 dataset was purposely used to test the performance of the system.

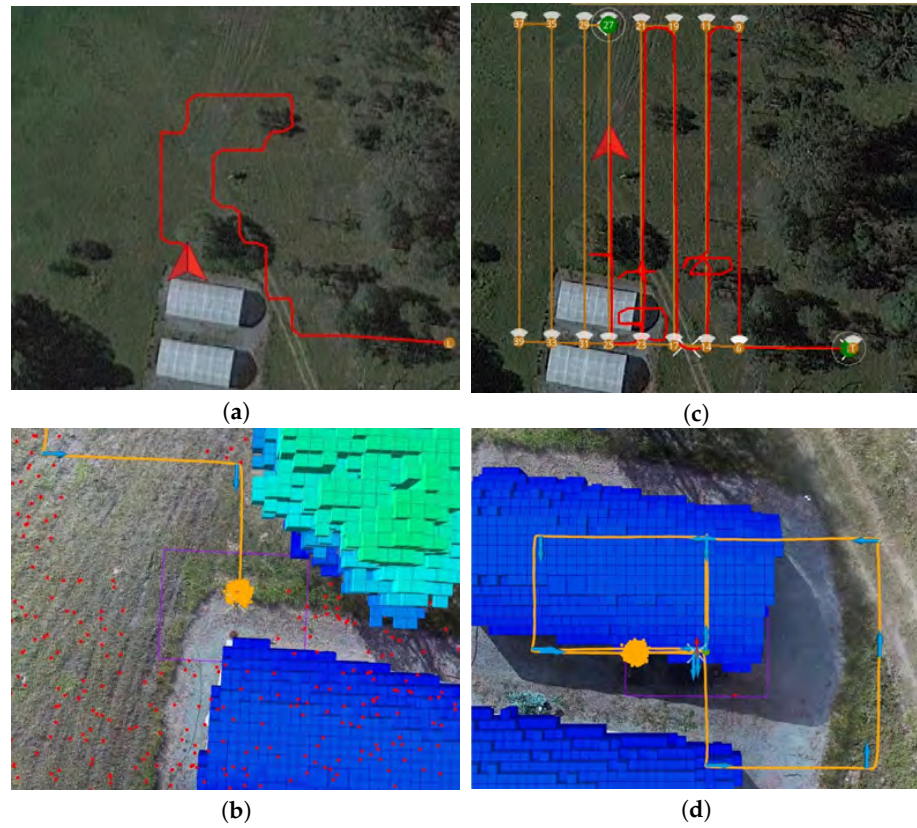


Figure 18. Illustration of two traversed paths using the proposed POMDP-based motion planner. (a) Traversed path of the UAV in offboard mode. (b) Probabilistic belief locations of the UAV (orange) and victim (red), the latter covering the entire surveyed area. (c) Traversed path of the UAV in hybrid mode. (d) Common path pattern by the UAV to confirm or discard positive detections sent by the computer vision module.

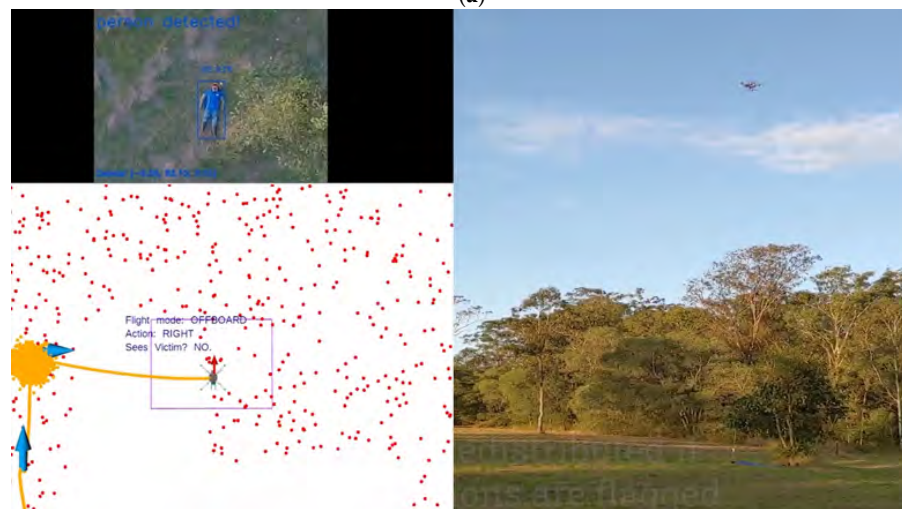
Table 6. Average duration among flight modes to locate the victim. Here, M1, M2, and M3 refer to mission, offboard, and hybrid flight modes, respectively.

Condition	Duration M1 (s)	Duration M2 (s)	Duration M3 (s)
Victim at Location 1	89	197	238
Victim at Location 2	139	144	353
Entire surveyed area	256	—	622

A further preliminary evaluation of the UAV system presented in this paper has been conducted with real flight tests at QUT SERF. The UAV framework has been adopted from the system tested by Sandino et al. [27] for a SAR case study in GNSS-denied environments, which follows a similar system architecture for simulation and real flight tests. The demonstrative flights tested mission, offboard, and hybrid flight modes to locate an adult mannequin placed in a trivial location. As shown in Figure 19, the UAV frame consists of a carbon-fibre Holybro X500 quadrotor, a Pixhawk 4 with GPS and 433 MHz telemetry radio. The tested companion computer in HIL mode (Section 4.3) and software tools (Section 4.4) are integrated to the frame during the tests, as well as a GoPro Hero 9 for real-time streaming of RGB frames. A detailed visual representation of the system can be found in supplementary video S4.



(a)



(b)

Figure 19. Demonstration of the presented UAV system in hardware with real flight tests at QUT SERF. (a) Holybro X500 quadrotor UAV frame kit with UP² as a companion computer and a GoPro Hero 9 as RGB camera. (b) UAV detecting an adult mannequin for the first time while navigating in offboard flight mode.

6. Discussion

The presented system constitutes a step towards fully autonomous navigation onboard small UAVs in real-world environments under uncertainty and partial observability. This research expands the navigation problem of Sandino et al. [26,27], by (1) formulating a more sophisticated reward function for a better discrimination between exploration and object inspection; (2) proposing optimised action commands based on the desired overlap between the camera's FOV and the UAV altitude; and (3) incorporating a new

flight mode (i.e., hybrid mode) to further optimise the exploration capabilities when partial information on the environment is provided beforehand (e.g., UAV navigation in outdoor scenarios, compared to GNSS-denied environments). Furthermore, this work indicates an enhanced design of UAV simulation environments by using realistic representations of ground textures and elevation, obstacle location and 3D occupancy maps from airborne UAV georeferenced mosaic and LiDAR datasets.

The system architecture and POMDP motion planner presented in this paper improve the autonomy capabilities of small UAVs under environment and object detection uncertainty. Under the presence of uncertainty from potential incorrect readings of vision-based detectors, such as CNNs, the standard objective of many existing approaches is to improve the performance metrics of their customised CNN-based object detectors for real-world UAV operations [40]. In contrast, this paper proposes an alternative approach to this uncertainty issue by augmenting the cognitive power onboard UAVs. Using reactive UAV systems that calculate and update a motion policy while interacting with the environment allow higher flexibility to detect objects (i.e., victims) at various levels of object detection uncertainty, regardless if this uncertainty comes from inaccurate detections by the detector model, noise in the data, or sensor malfunctioning. Even though the performance of the object detector model can be enhanced with tailored datasets and fit models from UAV data [41,42], this approach shows the possibility to use UAVs to navigate autonomously and accomplish a victim finding mission using off-the-shelf detectors, such as MobileNet SSDs.

The speed of the UAV system to survey an area is partly conditioned by the flight plan design, and especially the maximum surveying altitude. The maximum altitude is defined by extrinsic and intrinsic factors on a case study basis. Extrinsic factors include the maximum height of surrounding obstacles, lifting restrictions in high-altitude environments, and local flight regulations. Intrinsic factors are defined by hardware and software limitations in the UAV to detect objects (i.e., victims). For instance, the camera lens properties (e.g., sensor width and focal length) define the camera's FOV and, thus, the footprint extent and scale of observed objects. The image resolution of streamed camera frames into the computer vision module also determines the number of given pixels that represent an object to be detected using CNNs [43]. In this implementation, the system could capture positive victim detections using the MobileNet SSD model for a maximum surveying altitude of 20 m (and camera lens properties shown in Table 3). However, other CNN-based object detectors trained with airborne datasets are expected to allow surveys at higher altitudes and object detections with smaller scale visualisations, which translate in reduced duration for the UAV to survey the same area extent [40,43].

The design of the computer vision module is not limited to the MobileNet SSD detector, but it can be extended with other CNN architectures compatible with OpenVINO. Zhang et al. [44], for instance, propose a global density fused CNN model for UAV images that addresses object detection uncertainty factors such as variations in scale, viewpoint, and occlusion. A survey on CNN models for object detection on the VisDrone2019 dataset—an established low-altitude UAV dataset—shows that recent architectures, such as Cascade RCNN, RefineDet, and CornerNet outperform methods such as YOLO, SSD, and faster RCNN [40]. However, low average precision values for the best performing models (i.e., 17.41% on the VisDrone2019 dataset compared to 40.6% on the MS-COCO dataset) implies that further efforts in CNN architectures for object detection from airborne UAV data are required. Future implementations of CNN models discussed above could require a separate study of any model performance impacts caused by the streaming and processing of high-resolution frames onboard sub-2 kg UAVs, and onboard inference of CNN models in embedded computing systems, such as the UP², Jetson Nano, and Edge TPU [45,46].

The UAV system presented in this paper ultimately offers several benefits for first responders in the event of an emergency. The system provides real-time telemetry of camera frames processed onboard the UAV as, thus, prior secured risk and accessibility assessment of the environment, the presence of any found victims, and their visible health conditions. The generated list of GNSS coordinates shown in Figures 13–15 can be further

transferred to the ground team (SAR squads) to update intervention strategy, and other UAVs for air-to-ground first aid deployment. An additional extension of this system is the use of other sensors and computer vision models such as Doppler radars, microphones, and thermal cameras [5,47,48], which provide a better understanding of health conditions from potential victims.

7. Conclusions

This article presented a modular system for autonomous motion planning and object finding, capable of running in resource-constrained hardware onboard small UAVs. The system uses a POMDP to increase autonomy of UAVs for obstacle avoidance, environment exploration, and object inspection in outdoor environments under object detection uncertainty and partial observability. The system was evaluated with an emulated SAR scenario in Gazebo using real-world airborne georeferenced rasters, and airborne LiDAR point clouds to produce 3D occupancy maps. The task consisted of finding a lost person last seen in a bushland in two locations at various levels of occlusion and complex path planning strategies because of nearby obstacles. High-fidelity simulation results were achieved as the proposed system architecture combines SIL to emulate real-behaviour from a PX4 autopilot, and HIL for onboard inference of a MobileNet SSD detector, and the computed motion policy using TAPIR and the ABT solver.

The proposed UAV system was tested in simulation and with preliminary real flight tests with three flight configurations (i.e., mission, offboard and hybrid) and four performance metrics: (1) heatmaps of accumulated GNSS coordinates of the victim; (2) accuracy metrics to detect a victim; (3) speed analysis of proposed flight modes utilising the POMDP-based motion planner; and (4) speed analysis until the victim is found. Heatmaps show that false positive victim detections recorded in mission mode, from objects, such as trees, building structures and grass, were discarded (or rejected) by the UAV after exploring the area using the POMDP-based planner with and without a pre-defined flight plan (i.e., in hybrid and offboard modes respectively). Offboard and hybrid flight modes achieved the highest detection accuracy rates (of 100% and 92.5% respectively), and a slight decrease in detection rates (from 95.0% to 90.0%) was observed between trivial and complex victim locations. The elevated occurrence of false positive detections from the MobileNet SSD detector resulted in increased flight times in offboard and hybrid flight modes. However, the use of other CNN architectures, such as Cascade RCNN, RefineDet, and CornerNet are expected to reduce flight times of the UAV to inspect and reject false positive detections.

Further research avenues include a detailed experimental analysis using real flight tests, static and dynamic obstacles, and with sensors commonly used for SAR applications, such as thermal cameras. Placing the victim at random locations or increasing the number of victims could contribute to further understand the limits of the used POMDP solver. Similar case studies that can also be explored include surveillance for victims, using dynamic camera gimbal configurations and computer vision systems, which provide additional statistics of the victim, such as age, gender, and heartbeat rate. An evaluation of performance impacts in resource-constrained hardware for onboard inference using more complex CNN architectures for UAV object detection, and processing of high-resolution streamed frames, could help understand the limits of the proposed system architecture in sub-2 kg UAVs. Other sequential decision-making algorithms could also be implemented to compare the performance of the ABT solver.

Supplementary Materials: The following video demonstrations are available online: Video S1: system demonstration using mission mode, available at <https://youtu.be/H1qmdo1Qzok> (accessed on 6 November 2021); Video S2: system demonstration using offboard mode, available at <https://youtu.be/ykPI-yimgtQ> (accessed on 6 November 2021); Video S3: system demonstration using hybrid mode, available at <https://youtu.be/cFzx8XtGI2w> (accessed on 6 November 2021); and Video S4: Preview of UAV system running on real hardware at QUT SERF, available at https://youtu.be/U_9LbNXUwV0 (accessed on 6 November 2021).

Author Contributions: Conceptualisation, J.S., F.M., P.C., and F.G.; methodology, J.S., and F.G.; software, J.S.; validation, J.S.; formal analysis, J.S.; investigation, J.S.; resources, F.G.; data curation, J.S.; writing—original draft preparation, J.S.; writing—review and editing, F.M., P.C., C.S., and F.G.; visualisation, J.S.; supervision, F.M., P.C., C.S., and F.G.; project administration, F.G.; funding acquisition, P.C., and F.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The Commonwealth Scientific and Industrial Research Organisation (CSIRO) through the CSIRO Data61 PhD and Top Up Scholarships (Agreement 50061686), the Australian Research Council (ARC) through the ARC Discovery Project 2018 “Navigating under the forest canopy and in the urban jungle” (grant number ARC DP180102250) and the Queensland University of Technology (QUT) through the Higher Degree Research (HDR) Tuition Fee Sponsorship. The APC was funded by the QUT Office for Scholarly Communication (OSC).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Flight logs and recorded GNSS coordinates per flight mode and victim location are available by contacting the corresponding author upon request.

Acknowledgments: The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. The authors would also like to gratefully thank the QUT Research Engineering Facility (REF) team (Dmitry Bratanov, Gavin Broadbent, Dean Gilligan), for collection and processing of airborne LiDAR and imagery datasets that were used as a ground reference for the simulated environment. Special thanks to Hexagon through the Hexagon SmartNet RTK corrections service that enabled high accuracy surveying and positioning data using the EMLID Reach RTK receiver during the experimentation phase.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

2D	two-dimensional
3D	three-dimensional
ABT	augmented belief trees
AGL	above ground level
CNN	convolutional neural network
FOV	field of view
GNSS	global navigation satellite system
HIL	hardware in the loop
LiDAR	light detection and ranging
MAVLink	micro air vehicle link
MDP	Markov decision process
MTOW	maximum take-off weight
POMDP	partially observable Markov decision process
QUT	Queensland University of Technology
RGB	red, green, blue
ROS	robot operating system
SAR	search and rescue
SERF	Samford Ecological Research Facility
SIL	software in the loop
SSD	single shot detector
TAPIR	toolkit for approximating and adapting POMDP solutions in real time
UAV	unmanned aerial vehicle
VPU	vision processing unit

Appendix A. Difference Equation Constants

A Holybro S500 quad-rotor UAV (Holybro, China) was tested in a controlled VICON flying arena for a series of position responses in the x , y and z Cartesian axes. The UAV responses per axis were recorded for increments of 0.5 m every five seconds. The obtained difference equation constants for a sampling period $T_s = 0.1$ s are defined in Tables A1–A3:

Table A1. Difference in equation constants for the x Cartesian axis.

Variable	Value
a_0	+0.012237830217107
a_1	+0.005333276901521
a_2	−0.006904553315587
b_0	−1.871779712793530
b_1	+0.882425299507294

Table A2. Difference in equation constants for the y Cartesian axis.

Variable	Value
a_0	+0.015832293916205
a_1	+0.004320531132859
a_2	−0.011511762783346
b_0	−1.897777836334432
b_1	+0.906553659751270

Table A3. Difference in equation constants for the z Cartesian axis.

Variable	Value
a_0	+0.039080023298553
a_1	−0.074298943390646
a_2	−0.003370614541135
a_3	+0.074305261982209
a_4	−0.035703090165855
b_0	−3.839717616513733
b_1	+5.53607216125848
b_2	−3.55188073812765
b_3	+0.855538823595518

References

1. Australian National Search and Rescue Council. Search and Rescue Operations. In *National Search and Rescue Manual*, 2020 ed.; Number February; Australian Maritime Safety Authority (AMSA): Canberra, Australia, 2020; Volume 2, Chapter 3, pp. 73–353.
2. Bricknell, S. *Missing Persons: Who Is at Risk?* Technical Report; Australian Institute of Criminology: Canberra, Australia, 2017.
3. Hanson, L.; Namuduri, K. Real-World Applications. In *UAV Networks and Communications*; Namuduri, K., Chaumette, S., Kim, J.H., Sterbenz, J.P.G., Eds.; Cambridge University Press: Cambridge, UK, 2017; pp. 194–213, doi:10.1017/9781316335765.010.
4. Pajares, G. Overview and Current Status of Remote Sensing Applications Based on Unmanned Aerial Vehicles (UAVs). *Photogramm. Eng. Remote Sens.* **2015**, *81*, 281–330, doi:10.14358/PERS.81.4.281.
5. Lee, S.; Har, D.; Kum, D. Drone-Assisted Disaster Management: Finding Victims via Infrared Camera and LiDAR Sensor Fusion. In Proceedings of the 3rd Asia-Pacific World Congress on Computer Science and Engineering, Nadi, Fiji, 5–6 December 2016; pp. 84–89, doi:10.1109/APWC-on-CSE.2016.025.
6. Motlagh, N.H.; Bagaa, M.; Taleb, T. UAV-Based IoT Platform: A Crowd Surveillance Use Case. *IEEE Commun. Mag.* **2017**, *55*, 128–134, doi:10.1109/MCOM.2017.1600587CM.
7. Jiménez López, J.; Mulero-Pázmány, M. Drones for Conservation in Protected Areas: Present and Future. *Drones* **2019**, *3*, 10, doi:10.3390/drones3010010.

8. Erdelj, M.; Natalizio, E. UAV-assisted disaster management: Applications and open issues. In Proceedings of the International Conference on Computing, Networking and Communications, Kauai, HI, USA, 15–18 February 2016; pp. 1–5, doi:10.1109/ICCNC.2016.7440563.
9. Pensieri, M.G.; Garau, M.; Barone, P.M. Drones as an Integral Part of Remote Sensing Technologies to Help Missing People. *Drones* **2020**, *4*, 15, doi:10.3390/drones4020015.
10. Mayer, S.; Lischke, L.; Woźniak, P.W. Drones for Search and Rescue. In *Proceedings of the 1st International Workshop on Human-Drone Interaction*; Ecole Nationale de l'Aviation Civile [ENAC]: Glasgow, UK, 2019; pp. 1–7.
11. Valavanis, K.P.; Vachtsevanos, G.J. Future of Unmanned Aviation. In *Handbook of Unmanned Aerial Vehicles*; Valavanis, K.P., Vachtsevanos, G.J., Eds.; Springer: Dordrecht, The Netherlands, 2015; Chapter 126, pp. 2993–3009, doi:10.1007/978-90-481-9707-1_95.
12. Weldon, W.T.; Hupy, J. Investigating Methods for Integrating Unmanned Aerial Systems in Search and Rescue Operations. *Drones* **2020**, *4*, 38, doi:10.3390/drones4030038.
13. Chen, M.; Frazzoli, E.; Hsu, D.; Lee, W.S. POMDP-lite for Robust Robot Planning under Uncertainty. In Proceedings of the International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 5427–5433, doi:10.1109/ICRA.2016.7487754.
14. Ilhan, U.; Gardashova, L.; Kilic, M. UAV Using Dec-POMDP Model for Increasing the Level of Security in the Company. *Procedia Comput. Sci.* **2016**, *102*, 458–464, doi:10.1016/j.procs.2016.09.427.
15. Ponzoni Carvalho Chanel, C.; Albore, A.; T'Hooft, J.; Lesire, C.; Teichteil-Königsbuch, F. AMPLE: An anytime planning and execution framework for dynamic and uncertain problems in robotics. *Auton. Robot.* **2019**, *43*, 37–62, doi:10.1007/s10514-018-9703-z.
16. Vanegas, F.; Gonzalez, F. Uncertainty based online planning for UAV target finding in cluttered and GPS-denied environments. In Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016; pp. 1–9, doi:10.1109/AERO.2016.7500566.
17. Vanegas, F.; Campbell, D.; Eich, M.; Gonzalez, F. UAV based target finding and tracking in GPS-denied and cluttered environments. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, Daejeon, Korea, 9–14 October 2016; pp. 2307–2313, doi:10.1109/IROS.2016.7759360.
18. Kurniawati, H.; Yadav, V. An Online POMDP Solver for Uncertainty Planning in Dynamic Environment. In *Robotics Research. Springer Tracts in Advanced Robotics*; Inaba, M., Corke, P., Eds.; Springer: Cham, Switzerland, 2016; Volume 114, pp. 611–629, doi:10.1007/978-3-319-28872-7_35.
19. Ragi, S.; Chong, E.K.P. UAV Path Planning in a Dynamic Environment via Partially Observable Markov Decision Process. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 2397–2412, doi:10.1109/TAES.2013.6621824.
20. Ragi, S.; Chong, E.K.P. UAV Guidance Algorithms via Partially Observable Markov Decision Processes. In *Handbook of Unmanned Aerial Vehicles*; Valavanis, K., Vachtsevanos, G., Eds.; Springer: Dordrecht, The Netherlands, 2015; Chapter 73, pp. 1775–1810, doi:10.1007/978-90-481-9707-1_59.
21. Waharte, S.; Trigoni, N. Supporting Search and Rescue Operations with UAVs. In Proceedings of the International Conference on Emerging Security Technologies, Canterbury, UK, 6–7 September 2010; pp. 142–147, doi:10.1109/EST.2010.31.
22. Bravo, R.Z.B.; Leiras, A.; Cyrino Oliveira, F.L. The Use of UAVs in Humanitarian Relief: An Application of POMDP-Based Methodology for Finding Victims. *Prod. Oper. Manag.* **2019**, *28*, 421–440, doi:10.1111/poms.12930.
23. Kurniawati, H.; Hsu, D.; Lee, W.S.; Wee, D.H.; Lee, S. SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces. In Proceedings of the Robotics: Science and Systems IV, Zurich, Switzerland, 25–28 June 2008; pp. 1–8.
24. Dalamagkidis, K.; Valavanis, K.P.; Piegl, L.A. *On Integrating Unmanned Aircraft Systems into the National Airspace System*; Springer: Dordrecht, The Netherlands, 2012; pp. 1–305, doi:10.1007/978-94-007-2479-2.
25. Gupta, A.; Bessonov, D.; Li, P. A decision-theoretic approach to detection-based target search with a UAV. In Proceedings of the International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 24–28 September 2017; pp. 5304–5309, doi:10.1109/IROS.2017.8206423.
26. Sandino, J.; Vanegas, F.; Gonzalez, F.; Maire, F. Autonomous UAV Navigation for Active Perception of Targets in Uncertain and Cluttered Environments. In Proceedings of the 2020 IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2020; pp. 1–12, doi:10.1109/AERO47225.2020.9172808.
27. Sandino, J.; Vanegas, F.; Maire, F.; Caccetta, P.; Sanderson, C.; Gonzalez, F. UAV Framework for Autonomous Onboard Navigation and People/Object Detection in Cluttered Indoor Environments. *Remote Sens.* **2020**, *12*, 3386, doi:10.3390/rs12203386.
28. Chanel, C.; Teichteil-Königsbuch, F.; Lesire, C. Multi-target detection and recognition by UAVs using online POMDPs. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013; pp. 1381–1387.
29. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018; pp. 1–548.
30. Dutech, A.; Scherrer, B. Partially Observable Markov Decision Processes. In *Markov Decision Processes in Artificial Intelligence*; Sigaud, O., Buffet, O., Eds.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2013; Chapter 7, pp. 185–228, doi:10.1002/9781118557426.ch7.
31. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005; pp. 485–542.
32. Chovancová, A.; Fico, T.; Chovanec, L.; Hubinsk, P. Mathematical Modelling and Parameter Identification of Quadrotor (a survey). *Procedia Eng.* **2014**, *96*, 172–181, doi:10.1016/j.proeng.2014.12.139.

33. Bourke, P. Polygons and Meshes. 1997. Available online: <http://paulbourke.net/geometry/polygonmesh> (accessed on 6 October 2021).
34. Open Source Robotics Foundation. Robot Operating System. 2018. Available online: <https://www.ros.org> (accessed on 6 October 2021).
35. Chuanqi, Y. Caffe Implementation of Google MobileNet SSD Detection Network, with Pretrained Weights on VOC0712 and mAP = 0.727. 2020. Available online: <https://github.com/chuanqi305/MobileNet-SSD> (accessed on 6 October 2021).
36. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; ACM Press: New York, NY, USA, 2014; pp. 675–678, doi:10.1145/2647868.2654889.
37. Klimenko, D.; Song, J.; Kurniawati, H. TAPIR: A software Toolkit for approximating and adapting POMDP solutions online. In Proceedings of the Australasian Conference on Robotics and Automation, Melbourne, Australia, 2–4 December 2014; pp. 1–9.
38. Hornung, A.; Wurm, K.M.; Bennewitz, M.; Stachniss, C.; Burgard, W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robot.* **2013**, *34*, 189–206, doi:10.1007/s10514-012-9321-0.
39. Koubaa, A.; Allouch, A.; Alajlan, M.; Javed, Y.; Belghith, A.; Khalgui, M. Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey. *IEEE Access* **2019**, *7*, 87658–87680, doi:10.1109/ACCESS.2019.2924410.
40. Mittal, P.; Singh, R.; Sharma, A. Deep learning-based object detection in low-altitude UAV datasets: A survey. *Image Vis. Comput.* **2020**, *104*, 104046, doi:10.1016/j.imavis.2020.104046.
41. Lygouras, E.; Santavas, N.; Taitzoglou, A.; Tarchanidis, K.; Mitropoulos, A.; Gasteratos, A. Unsupervised Human Detection with an Embedded Vision System on a Fully Autonomous UAV for Search and Rescue Operations. *Sensors* **2019**, *19*, 3542, doi:10.3390/s19163542.
42. Bejiga, M.; Zeggada, A.; Nouffidj, A.; Melgani, F. A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery. *Remote Sens.* **2017**, *9*, 100, doi:10.3390/rs9020100.
43. Zhang, H.; Sun, M.; Li, Q.; Liu, L.; Liu, M.; Ji, Y. An empirical study of multi-scale object detection in high resolution UAV images. *Neurocomputing* **2021**, *421*, 173–182, doi:10.1016/j.neucom.2020.08.074.
44. Zhang, R.; Shao, Z.; Huang, X.; Wang, J.; Li, D. Object Detection in UAV Images via Global Density Fused Convolutional Network. *Remote Sens.* **2020**, *12*, 3140, doi:10.3390/rs12193140.
45. Mandel, N.; Milford, M.; Gonzalez, F. A Method for Evaluating and Selecting Suitable Hardware for Deployment of Embedded System on UAVs. *Sensors* **2020**, *20*, 4420, doi:10.3390/s20164420.
46. Faniadis, E.; Amanatiadis, A. Deep Learning Inference at the Edge for Mobile and Aerial Robotics. In Proceedings of the 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics, Abu Dhabi, United Arab Emirates, 4–6 November 2020; pp. 334–340, doi:10.1109/SSRR50563.2020.9292575.
47. Arnold, R.D.; Yamaguchi, H.; Tanaka, T. Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *J. Int. Humanit. Action* **2018**, *3*, 18, doi:10.1186/s41018-018-0045-4.
48. Al-Naji, A.; Perera, A.G.; Mohammed, S.L.; Chahl, J. Life Signs Detector Using a Drone in Disaster Zones. *Remote Sens.* **2019**, *11*, 2441, doi:10.3390/rs11202441.

Chapter 6

Reducing Object Detection Uncertainty from RGB and Thermal Data for UAV Outdoor Surveillance

BENEFITS of using UAVs have improved response efforts in time-critical applications as they provide a flexible and affordable eye in the sky safely and quickly. Recent advances in computer vision and machine learning have enabled the automated detection, localisation, and quantification of objects or persons of interest from streamed camera frames.

The research presented in this chapter is the fourth paper published from this research and extends the UAV navigation problem for outdoor environments presented in Chapter 5 with a framework implementation and comprehensive real flight tests using a sub 2 kg quadrotor UAV. The framework for autonomous navigation in outdoor scenarios under uncertainty was tested with a SAR case study to locate a person last seen in a bushland. The scalability of the framework is presented with preliminary experimental results using thermal imagery and a custom CNN object detector, and demonstrate greater flexibility of the UAV to interact with the environment and obtain clearer visualisations of any potential victims using RGB and thermal cameras.

Statement of Contribution of Co-Authors for Thesis by Published Paper

The authors listed below have certified that:

1. they meet the criteria for authorship and that they have participated in the conception, execution, or interpretation, of at least that part of the publication in their field of expertise;
2. they take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;
3. there are no other authors of the publication according to these criteria;
4. potential conflicts of interest have been disclosed to (a) granting bodies, (b) the editor or publisher of journals or other publications, and (c) the head of the responsible academic unit, and
5. they agree to the use of the publication in the student's thesis and its publication on the [QUT's ePrints site](#) consistent with any limitations set by publisher requirements.

In the case of this chapter:

J. Sandino, P. Caccetta, C. Sanderson, F. Maire, and F. Gonzalez, "Reducing object detection uncertainty from rgb and thermal data for UAV outdoor surveillance," in *Aerospace Conference, Big Sky, MT, USA: IEEE, Mar. 2022*, (accepted).

Contributor	Statement of Contribution ¹
Juan Sandino QUT Verified Signature Date: <u>2022-05-20</u>	Contributed on: conceptualisation, methodology, software, validation, formal analysis, investigation, data curation, writing-original draft preparation, and visualisation.
Peter Caccetta	Contributed on: writing-review and editing, supervision, and funding acquisition.
Conrad Sanderson	Contributed on: writing-review and editing, and supervision.
Frederic Maire	Contributed on: writing-review and editing, and supervision.
Felipe Gonzalez	Contributed on: conceptualisation, methodology, resources, writing-review and editing, supervision, project administration, and funding acquisition.

Principal Supervisor Confirmation		
I have sighted email or other correspondence from all co-authors confirming their certifying authorship.		
<i>Felipe Gonzalez</i>	QUT Verified Signature	<u>2022-05-20</u>
Name	Signature	Date

¹ Contributions follow the CRediT taxonomy. Further details can be found at <https://credit.niso.org/>.

Reducing Object Detection Uncertainty from RGB and Thermal Data for UAV Outdoor Surveillance

Juan Sandino
Queensland University of Technology
Brisbane, Australia
j.sandino@qut.edu.au

Peter A. Caccetta
Data61 / CSIRO, Australia
Peter.Caccetta@data61.csiro.au

Conrad Sanderson
Data61 / CSIRO, Australia
Conrad.Sanderson@data61.csiro.au

Frederic Maire
Queensland University of Technology
Brisbane, Australia
frederic.maire@qut.edu.au

Felipe Gonzalez
Queensland University of Technology
Brisbane, Australia
felipe.gonzalez@qut.edu.au

Abstract—Recent advances in Unmanned Aerial Vehicles (UAVs) have resulted in their quick adoption for wide a range of civilian applications, including precision agriculture, biosecurity, disaster monitoring and surveillance. UAVs offer low-cost platforms with flexible hardware configurations, as well as an increasing number of autonomous capabilities, including take-off, landing, object tracking and obstacle avoidance. However, little attention has been paid to how UAVs deal with object detection uncertainties caused by false readings from vision-based detectors, data noise, vibrations, and occlusion. In most situations, the relevance and understanding of these detections are delegated to human operators, as many UAVs have limited cognition power to interact autonomously with the environment. This paper presents a framework for autonomous navigation under uncertainty in outdoor scenarios for small UAVs using a probabilistic-based motion planner. The framework is evaluated with real flight tests using a sub 2 kg quadrotor UAV and illustrated in victim finding Search and Rescue (SAR) case study in a forest/bushland. The navigation problem is modelled using a Partially Observable Markov Decision Process (POMDP), and solved in real time onboard the small UAV using Augmented Belief Trees (ABT) and the TAPIR toolkit. Results from experiments using colour and thermal imagery show that the proposed motion planner provides accurate victim localisation coordinates, as the UAV has the flexibility to interact with the environment and obtain clearer visualisations of any potential victims compared to the baseline motion planner. Incorporating this system allows optimised UAV surveillance operations by diminishing false positive readings from vision-based object detectors.

critical applications such as surveillance, disaster monitoring, and Search and Rescue (SAR) [1–4]. UAVs offer unique benefits such as compact sizes and low cost to scout outdoor and indoor environments, real-time telemetry and camera streaming to monitor challenging and otherwise inaccessible environments, extensive payload adaptability, and extensive possibilities to augment navigation capabilities through software [5–8].

One critical challenge in deploying UAVs and robots in general into real-world and time-critical applications is the ever-presence of uncertainty. Factors that cause uncertainty are diverse, and they can be classified as external or internal. External factors come from sources beyond the scope of the UAV, such as poor weather and illumination conditions, strong gusts, unknown situational-awareness of surveyed environments, and partial observability. Internal factors include sub-optimal camera calibration settings, low image resolution, noisy camera frames during streaming, or imperfect detection outputs from computer vision detectors. As shown in Figure 1, uncertainty sources that are poorly managed can compromise the behaviour of UAVs and the flight mission itself [9]. Thus, it is essential to incorporate cognitive capabilities in UAVs to broaden their use in more real-world scenarios [10].

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. FRAMEWORK DESIGN.....	2
3. PLANNER DESIGN.....	4
4. EXPERIMENTS.....	6
5. RESULTS AND DISCUSSION.....	8
6. CONCLUSIONS.....	12
APPENDIX.....	13
ACKNOWLEDGMENTS.....	13
REFERENCES.....	13
BIOGRAPHY.....	15

1. INTRODUCTION

Recent advances in autonomous navigation of Unmanned Aerial Vehicles (UAVs)—also known as drones—have resulted in their gradual adoption in a set of civilian and time-



Figure 1. Unmanned aerial vehicle (UAV) navigating in environments under uncertainty and partial observability. A small UAV with autonomous decision-making should be able to plan sequential sets of actions for optimal navigation trajectories, despite limitations from imperfect sensor data.

The elevated number of stranded people and human loss is a

problem that is far from solved [11]. In Australia alone, an average of 38,000 people per year are reported missing and around 2% of them (or 720 persons) are never located [12]. In the event of an emergency—where time management plays a critical factor in the success of the rescue operation—the goal is to identify and locate as many victims as quick as possible. Thus, UAV technology for autonomous navigation and victim detection in challenging environments could assist first-responders in locating as many victims as soon as possible.

Research works on applied decision-making theory in UAVs is extensive and indicates that using Partially Observable Markov Decision Processes (POMDPs) onboard UAVs can increase their cognitive capabilities for autonomous navigation and object detection under uncertainty [13–15]. UAV frameworks for object detection and tracking have been tested in cluttered indoor environments and in the absence of Global Navigation Satellite System (GNSS) coverage [8, 16–18]. POMDPs have also been applied to solve multi-objective problems in UAVs, addressing tasks such as path planning, multiple object detection and tracking, and collision prevention [19, 20].

In time-critical applications such as SAR, real-time camera streaming is critical to comprehend the context of the environment [21]. However, drone pilots have a strong reliance on their communication systems to control most UAVs. If communication systems fail, the usability of the UAV could be seriously compromised [22]. Many approaches of POMDPs applied in UAVs for humanitarian relief operations have been tested in simulation [23, 24] and very few systems have been evaluated with real flight test using trivial targets [25].

Research efforts on onboard decision-making under object detection uncertainty from Convolutional Neural Network (CNN) models are scarce. Research conducted by Sandino et al. [8, 18] described a framework and POMDP problem formulation for a SAR application in GNSS-denied environments with a sub 2 kg UAV. However, the framework was only tested in cluttered indoor scenarios.

This paper describes a modular UAV framework for autonomous onboard navigation in outdoor environments under uncertainty. The framework design aims to reduce levels of object detection uncertainty using a POMDP-based motion planner, which allows the UAV to interact with the environment to obtain better visual representations of detected objects. CNN-based computer vision inference and motion planning can be executed in resource-constrained hardware onboard small UAVs. The framework is tested with real flight tests with a simulated SAR mission, which consisted of finding an adult mannequin in an open area and close to a tree. Three flight modes are proposed to evaluate the feasibility of the framework for real-world SAR operations.

This paper extends the research studies of Sandino et al. [8, 18, 26] with the following contributions: (1) an extension of their evaluated UAV framework—originally designed for navigation in GNSS-denied environments—for outdoor missions with GNSS signal coverage, and the design of a novel flight mode; (2) an additional validation of preliminary results of their proposed UAV framework with comprehensive real flight tests; and (3) a scalability approach of the framework by adapting a thermal camera and a custom object detector to locate victims using their heat signatures.

The rest of the manuscript is structured as follows: Section 2 details the UAV framework design for autonomous object

detection in uncertain outdoor environments; Section 3 summarises the implemented probabilistic-based motion planner using a POMDP; The design of conducted experiments using real flight tests is presented in Section 4; Obtained results and discussion of performance indicators are provided in Section 5; In Section 6, conclusions and future avenues for research are discussed.

2. FRAMEWORK DESIGN

The framework follows a modular system architecture for autonomous navigation onboard small UAVs as illustrated in Figure 2. This design extends an existing UAV framework for autonomous navigation in cluttered environments under object detection uncertainty, tested in simulation and with real flight tests in a sub 2 kg quadcopter [18].

Figure 2 illustrates the physical environment (or world) composed by the UAV frame and any attached payloads (i.e., RGB or thermal cameras), the victim and obstacles. Acquired camera frames represent the visual interface (also called observations) of the surveyed environment by the UAV. The UAV also contains the *autopilot*, which translates high-level action commands into low-level signals that control the UAV motors. The last hardware component of the UAV frame is a companion computer, which is allocated to execute software algorithms in dedicated modules for computer vision, mapping, and real-time path planning. Action commands from the planner are managed by the motion module, which interfaces with the flight controller of the autopilot.

The following subsections discuss each of the proposed framework components. The UAV framework used in this work is not limited to the hardware and software discussed below. Other UAV frame designs, payloads, autopilots, vision-based object detectors, planners, and software toolkits can also be implemented.

UAV Airframe and Payloads

The UAV airframe which offered the best combination between payload adaptability, size, and endurance for this research is a Holybro X500 quadrotor kit (Holybro, China). As shown in Figure 3, key components utilised from the kit include a Pixhawk 4 autopilot, Pixhawk 4 GNSS receiver, 2216 KV880 brushless motors, 22.86 cm plastic propellers, and a 433 MHz Telemetry Radio. With dimensions of 41 cm × 41 cm × 30.0 cm, the UAV carries a four cell 5000 mAh LiPo battery, for an approximate flight autonomy of 12 min.

The companion computer is an Intel UP², chosen for its price tag, number of peripherals and Central Processing Unit (CPU) architecture. Key specifications include a 64-bit quad-core CPU at 1.1 GHz, 64 GB eMMC SSD, 8 GB DDR3 RAM, four FL110 USB 3.0 connectors, two High-Speed UART controllers, and one mPCIe connector.

The proposed framework was tested using two Red, Green, Blue (RGB) cameras, namely an Arducam B019701 and a GoPro Hero 9. Thermal imagery is sourced from a FLIR Tau 2 connected to a ThermalCapture device for real-time frame streaming. The cameras, which can be interchangeably used in the proposed framework, are mounted onto an anti-vibration bracket, pointing to the ground and in parallel to Earth's nadir, as seen in Figure 4. Core properties for the cameras can be found in Table 4 in the Appendix.

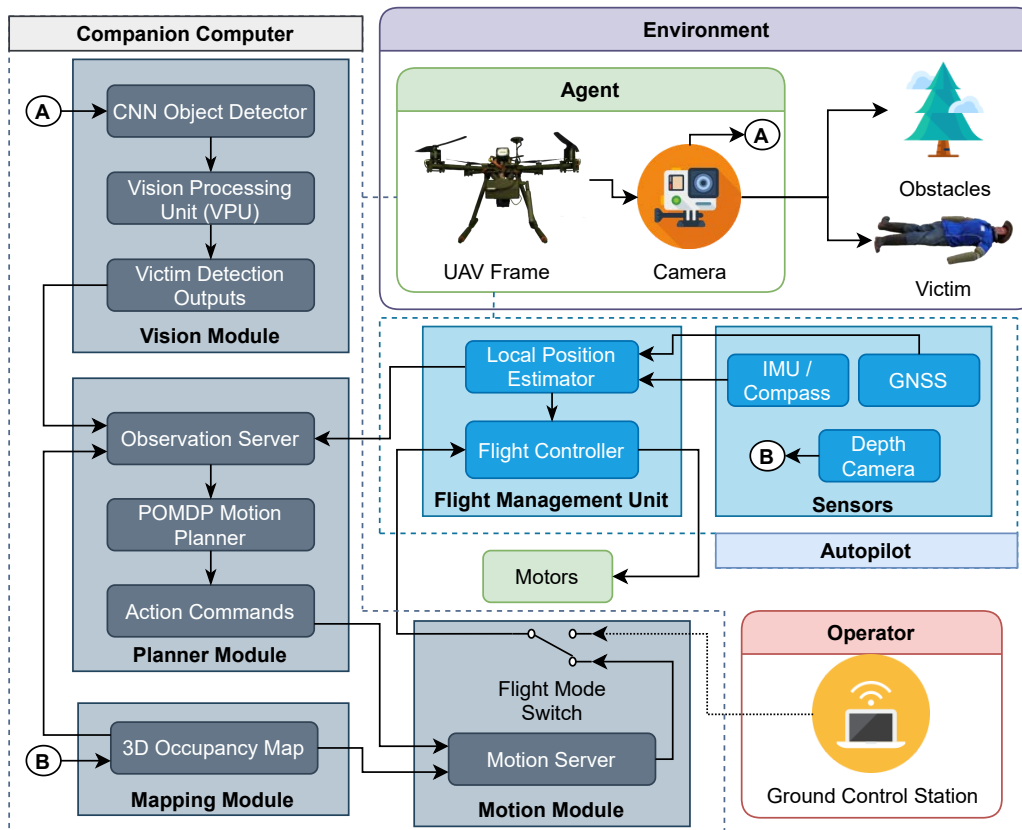


Figure 2. Modular system architecture for autonomous navigation onboard UAVs in uncertain outdoor environments. The framework portrays the physical environment (or world) composed of the UAV frame, attached payloads, world obstacles, and the victim. A companion computer is attached to the UAV to execute software algorithms in dedicated modules for computer vision, mapping, real-time path planning, and a motion server that interfaces the companion computer with the UAV autopilot.

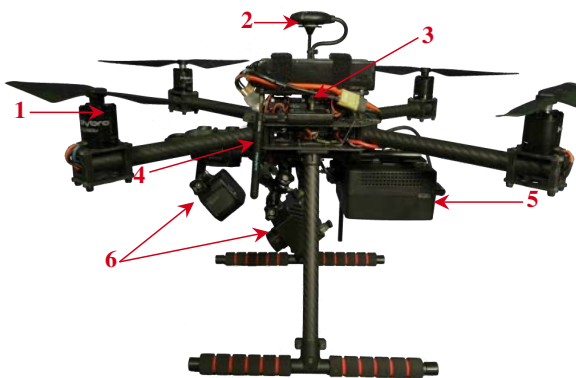


Figure 3. Framework implementation in a sub 2 kg quadrotor UAV. Primary components include: 1) a carbon fibre Holybro X500; 2) a Pixhawk 4 GNSS receiver; 3) a Pixhawk 4 autopilot; 4) a 433 MHz telemetry radio; 5) an Intel UP² (companion computer); and 6) payload.



Figure 4. GoPro Hero 9 mounted onto an anti-vibration bracket, pointing to the ground and in parallel to Earth's nadir.

Vision Module

This module consists of a deep learning object detector processing raw frames from the GoPro Hero 9 camera. Taking

into account the performance limitations of running deep learning models in resource-constrained hardware, a Vision Processing Unit (VPU) is installed in the companion com-

puter. Convolutional operations that normally run onboard a CPU or GPU are allocated to the VPU for inference of CNN models in resource-constrained hardware. In this implementation, the selected VPU is an Intel Movidius Myriad X, which is connected to the companion computer via the mPCIe slot. The detection module is programmed in Python and uses the OpenVINO library to optimise code instructions to load CNN models into the VPU.

The deep learning model architecture used to detect victims is an off-the-shelf Google MobileNet Single-Shot Detector (SSD) [27]. This model is deployed in Caffe [28] and tuned with pre-trained weights from the PASCAL VOC2012 dataset [29], scoring a mean average precision of 72.7%. The dataset covers up to 21 class objects (including persons). However, only positive detections for the class person are evaluated. Acquired camera frames are fitted into the input layer of the neural network (i.e., MobileNet SSD model) by downsizing the frames to 300×300 pixels.

Mapping Module

The Mapping module manages 3D occupancy maps, which are constituted by volumetric occupancy grids and displays the presence and localisation of objects in the surveyed environment. In this implementation, the 3D Occupancy Map are requested by the Motion and Planner modules to evaluate the presence of obstacles at selected position coordinates in the world coordinate frame. The maps are created through the use of the Octomap library [30].

Planner Module

The planner module computes the motion policy of the UAV and contains three primary components: (1) an observation server, which handles raw observations from the vision module (i.e., detected victims, confidence and victim coordinates), local position estimations of the UAV, and the state of the 3D occupancy map; (2) the POMDP motion planner, which calls the observation server every time the planner requires new observations; and (3) action commands computed by the motion policy of the planner and are read by the motion server. Complete details of the POMDP planner design can be found in Section 3.

Communication Interface

The UAV framework runs on open-source software tools. The companion computer runs under Linux Ubuntu 18.04 LTS O.S. and the Robot Operating System (ROS) melodic. ROS is a middleware to communicate between the nodes of each module (following the architecture design from Figure 2). The Pixhawk 4 autopilot is powered by PX4, which communicates with the companion computer through MAVROS, a ROS implementation of the MAVLink protocol, which is industry standard for UAV communication and control [31]. The POMDP solver implementation, which is described in Section 3, also contains a ROS implementation to maximise the use of visualisation, telemetry and recording tools from ROS.

3. PLANNER DESIGN

This approach formulates the decision-making problem as a POMDP. The planner transmits UAV position commands to the motion planner derived from environment observations. The discussion presented in this section is adapted from [26] and only essential parts are shown in this paper for completeness.

With a taken action $a \in A$, the UAV receives an observation $o \in O$ encoded by the observation function $\mathcal{Z}(s', a, o) = \mathbb{P}(o | s', a)$. Every decision chain is then quantified with an estimated reward r , calculated using the reward function $R(a, s)$. A POMDP uses a probability distribution over the system states to model uncertainty of its observed states. This modelling is called the belief $b(H) = \mathbb{P}[s^1 | H], \dots, \mathbb{P}[s^n | H]$, where H is the history of actions, observations and rewards the UAV has accumulated until a time step t , or $H = a_0, o_1, r_1, \dots, a_{t-1}, o_t, r_t$.

The motion policy π of the UAV is represented by mapping belief states to actions $\pi : b \rightarrow A$. A POMDP is solved after finding the optimal policy π^* , calculated as follows:

$$\pi^* := \arg \max_{\pi} \left(\mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R(S_t, \pi(b_t)) \right] \right), \quad (1)$$

where $\gamma \in [0, 1]$ is the discount factor and defines the relative importance of immediate rewards compared to future rewards. A given POMDP solver starts planning from an initial belief b_0 , which is usually generated using the initial conditions (and assumptions) of the flight mission.

Assumptions

In this implementation, the formulated problem for exploration and object detection (i.e., victims) using multi-rotor UAVs in outdoor environments assumes:

- An initial 3D occupancy map of the environment is pre-loaded to the planner before the UAV takes off.
- Observations come from processed camera frames (by the Vision module), the 3D occupancy map (by the Mapping Module), and the estimated local UAV position (by the Autopilot).
- Only a single, and static victim can be detected at the same time. If more victims appear on processed camera frames, the planner will only read data from the victim with the highest detection confidence values.
- The motion planner starts once the UAV reaches known position setpoint (i.e., at one of the corners of the surveyed area).
- The planner stops computing a motion policy once: 1) the UAV detects a victim whose detection confidence surpasses a set threshold; 2) the UAV explores the whole search area extent without finding any victims; or 3) the UAV exceeds the maximum flight time on air (because of low levels of battery power).

Actions

UAV actions are defined by seven position commands, namely *forward*, *backward*, *left*, *right*, *up*, *down*, and *hover*. UAV actions that are not included in the action space but are managed by the autopilot instead include *arm*, *disarm*, *take-off*, *return to launch and land*. Each action updates the position set point of the UAV in the world coordinate frame by calculating and applying a change of position δ .

The magnitude for δ_x and δ_y depends on the estimated overlap value between camera frame observations, calculated using Equation (2):

$$\delta = l_{\text{FOV}}(1 - \lambda), \quad (2)$$

where l_{FOV} is the length of the projected camera's Field of View (FOV), and $\lambda \in [0, 1)$ is the desired overlap value.

States

A system state $s \in S$ is defined as:

$$s = (p_u, f_{\text{roi}}, f_{\text{dct}}, p_v, c_v) \quad (3)$$

where p_u is the position of the UAV in the world coordinate frame, f_{crash} is a flag raised when the UAV crashes with an obstacle, f_{roi} is a flag indicating whether the UAV is flying beyond the flying limits, f_{dct} is the flag raised if a potential victim is detected by the UAV. If $f_{\text{dct}} = \text{True}$, the position of the victim in the world coordinate frame is given in p_v , with detection confidence $c_v \in [0, 1]$. The system reaches a terminal state whenever $c_v \geq \zeta$, where ζ is the confidence threshold.

Transition Function

The motion dynamics of a multi-rotor UAV defines the transition from current to new states:

$$p_u(k+1) = p_u(k) + \Delta p_u(k) \quad (4)$$

where $p_u(k)$ is the position of the UAV at time step k , and $\Delta p_u(k)$ is the position change of the UAV between time steps. This formulation does not contain any actions for heading changes. However, Equation (4) can be expanded if required by adding the rotation matrix in multi-copters [32]. An illustration of a problem formulation including the rotation matrix can be found in [8].

Reward Function

The expected reward r after taking an action $a \in A$ from state $s \in S$ is calculated using the reward function $R(a, s)$ defined in Algorithm 1 and Table 1. This function critically influences the UAV behaviour during flight missions, and its definition allows multi-objective task definition. A complete discussion on the design considerations of the reward function can be found in [26].

Table 1. Applied reward values to the reward function R , defined in Algorithm 1.

Variable	Value	Description
r_{crash}	-50	Cost of UAV crash
r_{out}	-25	Cost of UAV breaching safety limits
r_{dct}	+25	Reward for detecting potential victim
r_{conf}	+50	Reward for confirmed victim detection
r_{action}	-2.5	Cost per action taken
r_{fov}	-5	Footprint overlapping cost

The order of the steps from Algorithm 1 classifies high-level tasks into two components. The first one is object detection and starts by evaluating any states which will negatively affect the integrity of the UAV, followed by states indicating positive victim detections. If the UAV detects a potential victim (Step 6), R calculates a linear function (Step 8) which returns increased reward values as the UAV gets closer to the minimum allowed altitude. A higher reward value is returned if a potential victim is confirmed (Step 9 and 10).

Algorithm 1 Reward function R for exploration and object detection in outdoor environments.

```

1:  $r \leftarrow 0$ 
2: if  $f_{\text{crash}}$  then
3:    $r \leftarrow r_{\text{crash}}$  ▷ UAV crashing cost
4: else if  $f_{\text{roi}}$  then
5:    $r \leftarrow r_{\text{out}}$  ▷ Beyond safety limits cost
6: else if  $f_{\text{dct}}$  then
7:    $r \leftarrow r_{\text{dct}}$  ▷ Detected object reward
8:    $r \leftarrow r + \left[ r_{\text{dct}} \cdot \left( 1 - \frac{z_u - z_{\text{min}}}{z_{\text{max}} - z_{\text{min}}} \right) \right]$  ▷ UAV altitude reward
9:   if  $c_v \geq \zeta$  and  $a = \text{Down}$  then
10:     $r \leftarrow r + r_{\text{conf}}$ 
11:   end if
12: else
13:    $r \leftarrow r_{\text{action}}$  ▷ Action cost
14:    $r \leftarrow r - \left[ r_{\text{dct}} \cdot \left( 1 - \frac{z_u - z_{\text{min}}}{z_{\text{max}} - z_{\text{min}}} \right) \right]$  ▷ UAV altitude cost
15:    $r \leftarrow r - \left[ r_{\text{dct}} \cdot \left( 1 - 0.5^{4 \cdot d_v / d_w} \right) \right]$  ▷ Horizontal distance cost
16:    $r \leftarrow r + r_{\text{fov}} \cdot \varepsilon$  ▷ Footprint overlap cost
17: end if
18: return  $r$ 

```

The second component of the algorithm addresses exploration. In case there are no detections, R applies a set of cost functions to encourage a greedy horizontal exploration of the environment. An exponential function in Step 14 calculates the Manhattan distance between the UAV and the victim d_v and the maximum exploration distance d_w which are defined as follows:

$$d_v = \sum_{i=1}^n |p_i - q_i|, p_i=(x_u, y_u), q_i=(x_v, y_v) \quad (5)$$

$$d_w = \sum_{i=1}^n |p_i - q_i|, p_i=(x_{\text{max}}, y_{\text{max}}), q_i=(x_{\text{min}}, y_{\text{min}}) \quad (6)$$

The overlap ε between the camera's current footprint and its correspondent location in the footprint map is defined as follows:

$$\varepsilon = \frac{\sum_{i=1}^n F_i(p_u)}{n}, \varepsilon \in [0, 1] \quad (7)$$

where $F_i(p_u)$ are the pixel values of the projected FOV in the footprint map, and n is the total number of projected pixels in the footprint. A maximum overlap value of 1 indicates that such action will place the UAV to a fully previously explored area, and, as indicated in Step 16 of Algorithm 1, the whole penalty value r_{fov} will be added to the reward. A minimum value of 0 means that a given action will place the UAV in an unexplored area and no penalty will be added to the reward. Intermediate values of ε represent partial overlapping, adding a partial penalty value r_{fov} to the reward.

Observations

An observation $o \in O$ is defined as follows:

$$o = (o_{p_u}, o_{\text{dct}}, o_{p_v}, o_{\zeta}, o_{\text{obs}}), \quad (8)$$

where o_{p_u} is the estimated position of the UAV by the autopilot; o_{dtc} is the flag triggered by potential victim detections received by the CNN model; o_{p_v} and o_ζ are the local position of the victim and the detection confidence respectively, both of them defined only if there are any positive detections; and o_{obs} is the flag triggered after processing the 3D occupancy map for any obstacles located in front of the UAV.

The detection confidence o_ζ measures the frequency of positive detections between the last two observation calls:

$$o_\zeta = \frac{\sum_{i=1}^n o_{\text{dtc}_i}}{n}, \quad (9)$$

where n is the number of segmented frames between observation calls, and o_{dtc} is the flag indicating a positive detection per processed frame i .

Observation Model

This implementation uses Augmented Belief Trees (ABT) [33], an online POMDP solver that contains a model that generates T and \mathcal{Z} using a modelled observation o given an action a and the next state s' . The variables contained in the generative model are the local position of the UAV s'_{p_u} , the local position of the victim s'_{p_v} and the detection confidence o_ζ .

Potential victim detections and their subsequent positioning estimations are conditioned by the camera pose at the UAV frame and its projected footprint of the environment. Specifically, if the 2D local position coordinates of the victim $s'_{p_v}(x, y)$ are within the projected footprint limits of the camera, the victim is assumed to be detected. This estimation is done by calculating the sum of angles between a 2D point (i.e., s'_{p_v}) and each pair of points that constitute the footprint boundaries (the footprint rectangular corners) [34]. The 2D projected footprint extent l of a vision-based sensor, illustrated in Figure 5, can be calculated using Equations (10) and (11):

$$l_{\text{top, bottom}} = s'_{p_u}(z) \cdot \tan\left(\alpha \pm \tan^{-1}\left(\frac{h}{2f}\right)\right), \quad (10)$$

$$l_{\text{left, right}} = s'_{p_u}(z) \cdot \tan\left(\pm \tan^{-1}\left(\frac{w}{2f}\right)\right), \quad (11)$$

where s'_{p_u} is the UAV altitude, α and β are the camera's pointing angles from the vertical z and horizontal x axis of the World coordinate frame, w is the lens width, h is the lens height, and f is the focal length.

The footprint corners c from the camera's local coordinate frame I are translated to the world's coordinate frame W using the following transformation:

$$\begin{bmatrix} c'(x) \\ c'(y) \end{bmatrix} = \begin{bmatrix} s'_{p_u}(x) \\ s'_{p_u}(y) \end{bmatrix} + \begin{bmatrix} \cos(\varphi_u) & -\sin(\varphi_u) \\ \sin(\varphi_u) & \cos(\varphi_u) \end{bmatrix} \begin{bmatrix} c(x) \\ c(y) \end{bmatrix}, \quad (12)$$

where s'_{p_u} is the next UAV position state, and φ_u is the Euler

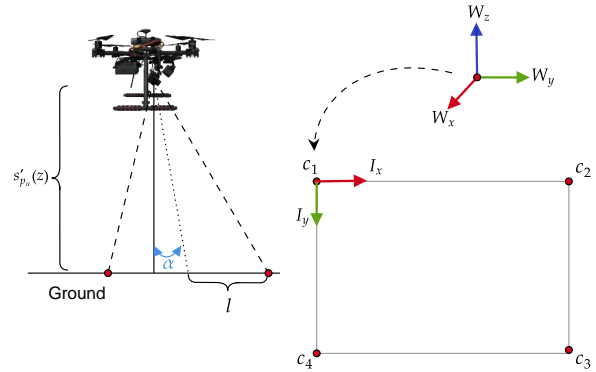


Figure 5. Field of View (FOV) projection and footprint extent of a vision-based sensor. The camera setup on the UAV frame defines α as the pointing angle from the vertical (or pitch) and determines the coordinates of the footprint corners c .

yaw angle of the UAV. However, as no actions involve adjusting the heading of the UAV mid-flight, and assuming yaw estimation errors are negligible, Equation (12) is simplified as follows:

$$\begin{bmatrix} c'(x) \\ c'(y) \end{bmatrix} = \begin{bmatrix} s'_{p_u}(x) + c(x) \\ s'_{p_u}(y) + c(y) \end{bmatrix}. \quad (13)$$

The detection confidence o_ζ that comes as part of the output data from the CNN object detector is modelled using Equation (14):

$$o_\zeta = \frac{(1 - \zeta_{\min})(d_{uv} - z_{\min} + \zeta_{\min})}{z_{\max} - z_{\min}}, \quad (14)$$

where ζ_{\min} is the minimally accepted confidence threshold, z_{\max} and z_{\min} are the maximum and minimum UAV flying altitudes respectively, and d_{uv} is the Manhattan distance between the UAV and the victim.

4. EXPERIMENTS

This research validated the proposed UAV framework with real flight tests on the sub 2 kg quadcopter shown in Section 2. The tests were designed under a ground SAR application context, specifically, to locate a lost person last seen around a forest/bushland area. The subsections below present the location of conducted flights, environment setup, proposed flight modes for data collection and tuned hyperparameters of the online POMDP solver.

Location and Environment Setup

Flight tests were conducted at the Samford Ecological Research Facility (SERF), 148 Camp Mountain Road, Samford QLD 4520, Australia. As shown in Figure 6, the 51 hectare property contains protected Dry Sclerophyll forest and grazing zones, where the latter ones were utilised to fly the UAV.

The delimited flying area covers a mostly flat grazing zone featuring buffel grassland, a five-metre tree, and a car pur-

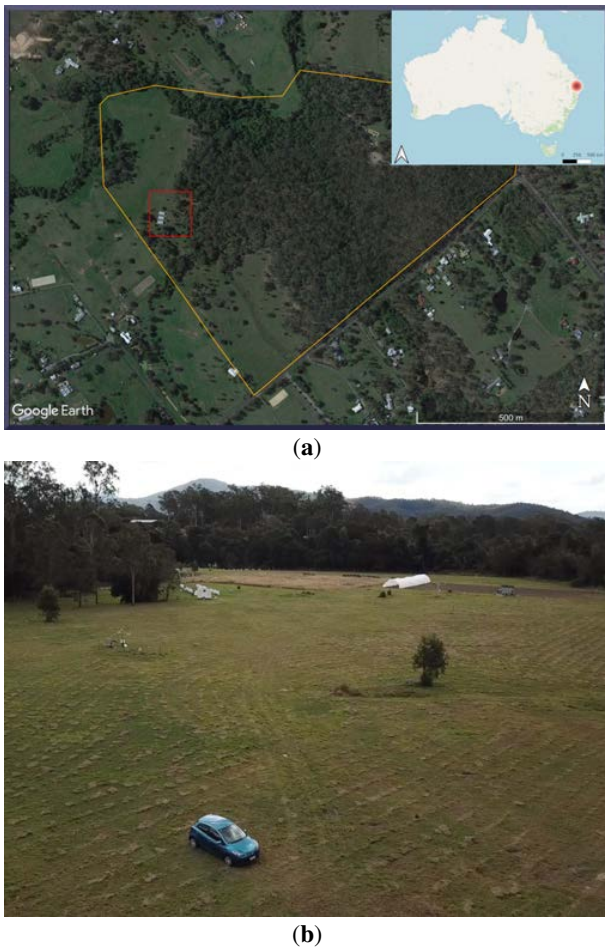


Figure 6. Location of conducted flight tests at the Samford Ecological Research Facility (SERF), QLD, Australia. (a) SERF and surveyed area boundary extents (orange and red blobs respectively). (b) Aerial footage of surveyed area displaying buffel grassland and obstacles.

posedly placed as an additional obstacle. Flight tests were conducted between the 19th July 2021 and the 8th of September 2021, in a rich range of illumination and weather conditions. Weather conditions included tests under clear and partly cloudy skies, with calm and gusty winds from 6 km/h up to 24 km/h respectively. The range of recorded temperatures ranged from 14°C to 25°C.

This implementation employed a static adult mannequin posing as the victim to be found for safety reasons. The mannequin was placed at two predefined locations, as depicted in Figure 7. The first location—referred from here as Location 1, or L1—is a trivial setup with the mannequin free of any nearby obstacles and entire visibility from downward-looking cameras. The second location—referred from here as Location 2, or L2—introduces a complex setup as the mannequin is placed nearby a tree which causes partial occlusion during the flight tests.

Flight Modes

The proposed UAV system is evaluated by collecting data using three flight modes: *mission*, *offboard*, and *hybrid*. The

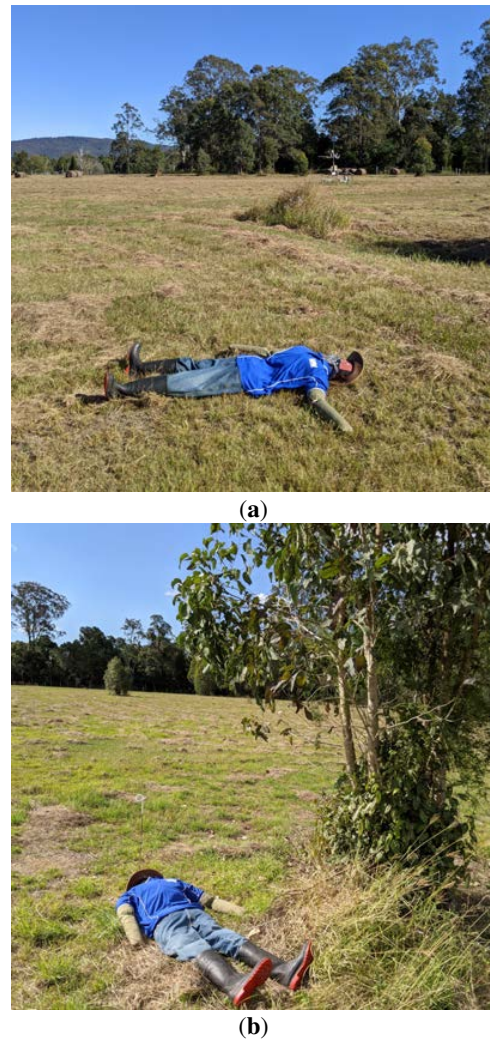


Figure 7. Adult mannequin placed in the surveyed area as the victim to be found. (a) Trivial victim location (L1) with the mannequin fully exposed in the environment. (b) Complex victim location (L2) with the mannequin partly occluded by a five-metre tree.

survey extent for these tests is delimited by a 6 m × 60 m rectangular area, drafted in QGroundControl. Specific details of the survey pattern can be found in Figure 15 and Table 4 from the Appendix. A diagram illustrating the functionality of tested flight modes is shown in Figure 8.

Mission Mode—When *mission* mode is activated, the UAV automatically follows a list of position and velocity waypoints which define the survey plan previously drafted in QGroundControl and uploaded to the autopilot before starting the flight operation. This flight mode is traditionally supported in many autopilots, and its out-of-the-box implementation serves as the planner baseline of this research. While mission mode is operated in the UAV, the object detector is running in parallel to record any positive detections while the UAV is navigating in the environment and completing the survey.

Offboard mode—*Offboard* mode offers autonomous navigation without a predefined survey plan of the environment.

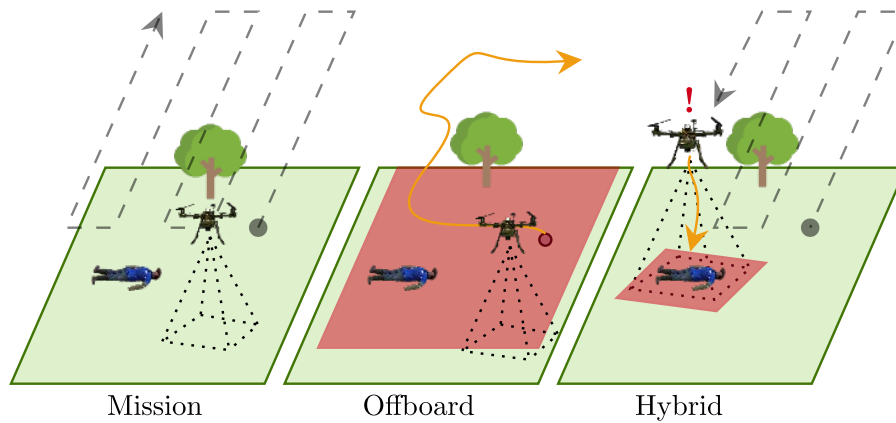


Figure 8. Executed flight modes for exploration and object detection in outdoor environments. Mission mode is the baseline motion planner and lets the UAV survey the SAR emulated area by following a lawn-mower pattern. Offboard mode runs the POMDP motion planner by populating an initial victim position belief across the entire flying area. Hybrid mode extends the functionality of mission mode by running the POMDP motion planner to inspect the area delimited by the camera's FOV.

This flight mode internally executes the POMDP-based motion planner described in Section 3 by declaring as flight parameters the initial position waypoint where the UAV should begin the survey, and the global coordinates of the survey extents. The list of parameters can be found in Table 5 from the Appendix.

Hybrid Mode—This paper proposes the fusion of the provided capabilities between mission and offboard modes, in a flight mode denominated *hybrid*. The aim of this flight mode is to take advantage of the initial awareness and survey coverage coming from mission mode in outdoor environments with GNSS signal coverage, and the autonomous navigation capabilities of offboard mode. Instead of running the POMDP-based motion planner covering the entire extent of the surveyed area, in hybrid mode the survey extent is only limited by the extent of the camera's FOV. Once a first detection is received from the vision module, this flight mode triggers offboard mode, boots the motion planner and passes action commands to the autopilot until the POMDP solver reaches a terminal state (i.e. the UAV discards or confirms a victim). Afterwards, the UAV resumes its survey by triggering back mission mode. The process repeats itself with new detection outputs until the UAV completes the survey in mission mode.

POMDP solver

The navigation problem modelled as a POMDP is solved in real time through the use of the TAPIR toolkit [35]. TAPIR is coded using the C++ programming language and encapsulates the Augmented Belief Trees (ABT) solver [33] to calculate and update the motion policy online. ABT reduces computational demands by reusing past computed policies and updating the optimal policy if changes to the POMDP model are detected. Furthermore, formulated problems with ABT allow declaring continuous values for actions, states, and observations.

Once the motion server calls the motion planner after a first victim detection is received by the object detector, TAPIR is booted by calculating an offline policy for four seconds. Afterwards, the observation server retrieves an observation,

updates the motion policy and takes the action that returns the highest expected reward. An idle period of 3.4 seconds is applied for the UAV to reach the desired position coordinate, and then, the process repeats itself by requesting a new observation from the observation server. The loop is broken once the detection confidence ζ exceeds a threshold. Specific parameters from the TAPIR toolkit and ABT solver are shown in Table 5 from the Appendix.

5. RESULTS AND DISCUSSION

The proposed UAV framework is evaluated through the performance indicators listed as follows: 1) Successful detections per flight mode; 2) Spatial distribution of recorded GNSS coordinates via heatmaps; 3) Elapsed time taken by the UAV to locate the victim per location; and 4) Scalability test using thermal imagery. Real flight demonstrations of the UAV framework can be found at https://youtu.be/U_9LbNXUwV0.

Accuracy metrics of victim detections were recorded using three variables: True Positives (TP), False Positives (FP), and False Negatives (FN). TP is defined here as the relative number of flight runs where the victim was successfully detected at the true location. FP is the relative number of flights which recorded victim locations in other areas than the true position of the victim. FN is the relative number of flights that did not detect the victim at their real location. In this context, a given flight test could report false positive detections and still detect the victim at the real location. A summary table of collected metrics is depicted in Table 2.

Accuracy metrics of the proposed framework provided contrasting results at the tested victim locations. On flight tests with the victim placed in a trivial location (i.e., L1), the UAV achieved 100% of positive victim detections in mission mode, and 20% of those recorded GNSS coordinates of false victim locations. For tests in offboard and hybrid flight modes, the true positive rates decreased in comparison with mission mode. However, both setups achieved flight tests without any false positive readings. An illustration of the spatial

Table 2. Accuracy metrics of the system to locate a victim at two locations (L1 and L2). Here, TP are true positives, FP are false positives, and FN are false negatives.

Flight Mode	Runs	TP (%)	FP (%)	FN (%)
Mission (L1)	5	100.0	20.0	0.0
Offboard (L1)	7	57.1	0.0	42.9
Hybrid (L1)	5	80.0	0.0	20.0
Mission (L2)	5	60.0	0.0	40.0
Offboard (L2)	5	80.0	0.0	20.0
Hybrid (L2)	7	71.4	28.6	28.6

distribution of recorded GNSS coordinates per flight mode for L1 is shown in Figure 10.

Flight tests with the mannequin located in a complex location (i.e., L2) showed an overall improvement in TP rates for offboard and hybrid modes compared to the baseline planner (mission mode). For mission and offboard modes, there were no flights which reported FP detections even though the flight setup, payload, and object detector remained the same while testing the framework with the mannequin located in L1. Nevertheless, 28.6% of the flights in hybrid mode reported false positive victim locations. The rate of false positives for all the flight tests were caused by limitations from the object detector by recording other objects as humans (i.e., from the car placed in the flight area). Conversely, most of the false negative records during the flight tests occurred from excessive vibration in the UAV frame caused by strong winds, as shown in Figure 9.

A visual analysis of the spatial distribution of recorded GNSS coordinates during the flight tests is performed using heatmaps. The heatmaps, illustrated in Figures 10 and 11, indicate a reduction in victim location uncertainty after operating the proposed UAV framework using offboard and hybrid flight modes.

The presented POMDP-based motion planner from Sandino et al.[18]—scaled in this work for UAV navigation in outdoor environments—contributed on the reduction of object detection uncertainty in flight tests using offboard and hybrid modes. An example of how the UAV inspects an area to confirm whether a victim is truly located after receiving an initial detection is shown in Figure 12. In the first time steps (Figure 12a), a low confidence value of 19.51% is retrieved because of the few number of pixels representing the mannequin while surveying at 16 m, and partial occlusion from the nearby tree. After taking actions commands from the POMDP policy computed in the planner module, the UAV is positioned closer to the mannequin (i.e., 10 m) and with a better viewpoint of the scene, retrieving a confidence value of 90.0% (Figure 12b). The traversed path by the UAV also suggests the capability of the UAV to adapt (or update) its motion policy while it interacts with the environment and receives new observations. Adjustments in the motion policy also occur from uncertainty sources, such as unexpected strong wind currents, oscillating GNSS signal errors, illumination changes, and false detections by the CNN model.

This work also studied the speed of the proposed framework to find victims using mission, offboard, and hybrid flight modes. As presented in Table 3, two primary insights are

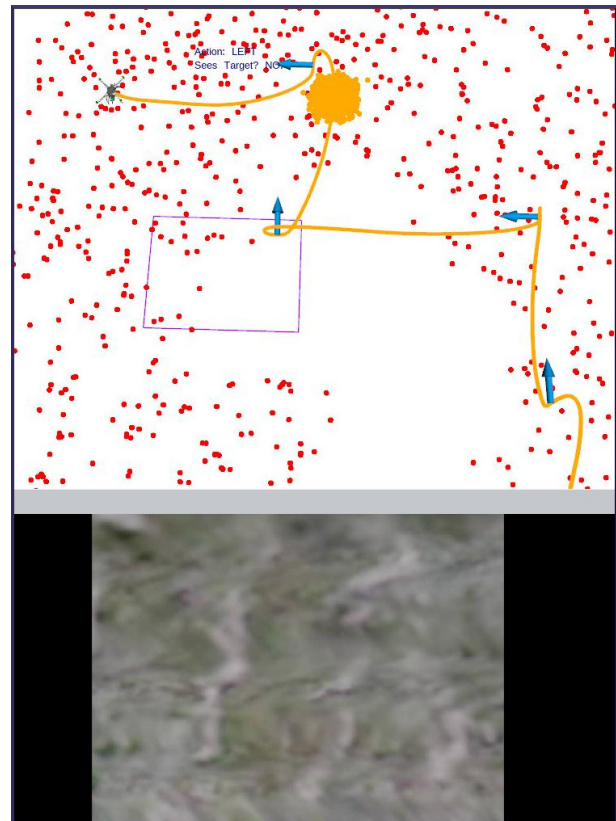


Figure 9. Strong winds distorting RGB streaming in offboard mode. The top image illustrates the traversed path of the UAV and the probability distribution of the UAV and victim locations (orange and red points respectively). The bottom image shows the latest (distorted) streamed frame from the RGB camera.

observed in elapsed times between victim locations and flight modes. The first observed trend defines higher values of standard deviation (especially in offboard mode) in flight tests with the victim at location L1. These values were caused by slight inconsistencies in the traversed path from the motion planner, as seen in Figure 13. From the surveyed area, Location 1 is closer to the survey limits, whereas Location 2 is placed closed the upper centre. The motion planner lets the UAV move towards the centre of the surveyed area before exploring its corners. Despite test runs in offboard mode providing higher overall accuracy values than tests in hybrid mode, surveyed patterns could become suboptimal if a victim is located close to the boundaries of the delimited flying area.

The second observed trend is that hybrid mode recorded longer times for detecting and confirming the victim, regardless of its location in the surveyed area. This impact highly depends on the recall properties of the vision-based object detector. The number of false positive outputs while the UAV explores the environment defines the number of inspections, which will increase the flight time until the survey is complete. Using other object detectors tuned from airborne UAV datasets is expected to reduce the survey duration in hybrid mode, as these models should provide higher recall values than the MobileNet SSD detector implemented in this paper.

Limitations in the implemented object detector to output



(a)



(b)



(c)

Figure 10. Heatmaps of recorded GNSS coordinates in a trivial victim location (L1) using (a) mission, (b) offboard, and (c) hybrid flight modes.

positive detections have conditioned the maximum altitude for the surveys, and consequently, impacted the overall speed of the system to complete the mission. Other restrictions are defined by the gimbal configuration, which scopes the footprint of the camera's FOV, and the image resolution of streamed frames. Therefore, reduced times to accomplish the



(a)



(b)



(c)

Figure 11. Heatmaps of recorded GNSS coordinates in a complex victim location (L2) using (a) mission, (b) offboard, and (c) hybrid flight modes.

victim finding mission can be accomplished by improvements in the object detector, higher image resolution and oblique camera angles that provide a wider viewpoint of the scene. Nonetheless, such improvements might involve an increment in computational power and further research should evaluate any impacts from better detectors and processing of high-

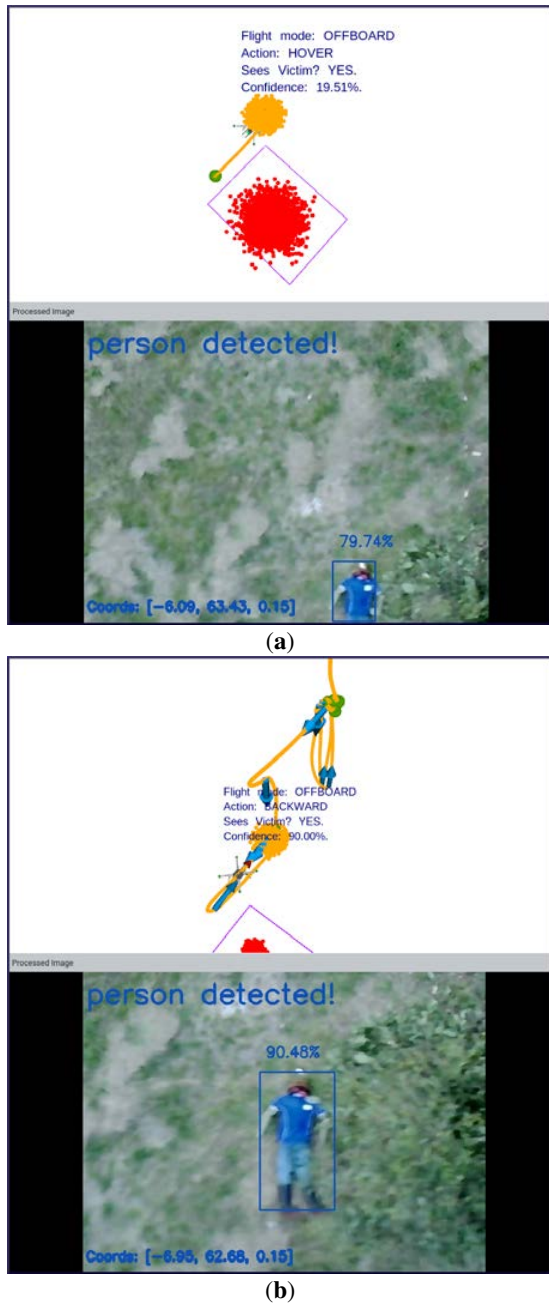


Figure 12. Reduction of object detection uncertainty from RGB camera after executing the motion policy. (a) Initial detection of a potential victim with a confidence value of 19.51%. (b) Increased confidence value (90.0%) at subsequent detections after the UAV executes actions from the computed motion policy and gets closer to the victim.

resolution frames in embedded systems such as the UP².

The scalability test of the presented framework was achieved with preliminary flight tests using thermal imagery. The modular components replaced for this trial consisted of the UAV payload, the vision-based detector, and the flight plan generated by QGroundControl to match the desired overlap

Table 3. Elapsed time by the UAV to locate a victim at two locations (L1 and L2) per flight mode. Here, SD stands for Standard Deviation and SE stands for Standard Error.

Flight Mode	Mean (s)	SD (s)	SE (s)
Mission	169.67	–	–
Offboard (L1)	146.24	128.70	64.35
Hybrid (L1)	392.39	130.20	65.10
Offboard (L2)	148.30	25.58	12.79
Hybrid (L2)	289.26	76.84	34.36

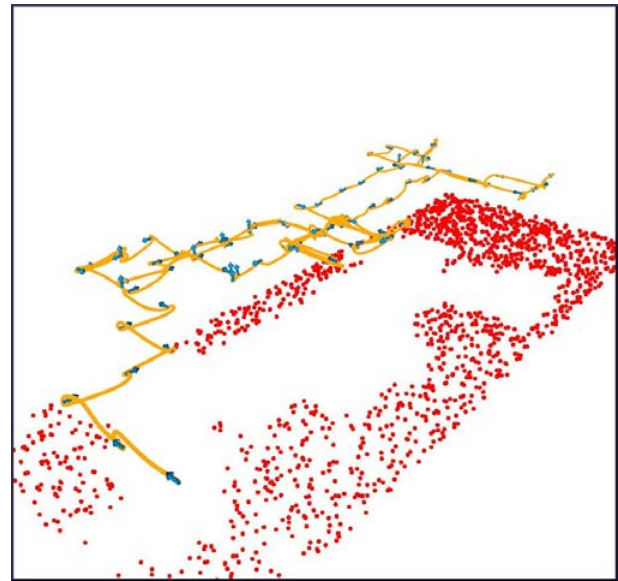


Figure 13. Example traversed path in offboard mode while no victims are found. The UAV moves towards the centre of the surveyed area before exploring its corners.

values previously tested with the RGB camera. The object detector is a custom implementation of a TinyYoloV2 model architecture from Microsoft Azure Custom Vision services. A total of 5175 labelled images were used in the training process. The thermal dataset contains images of adults with a rich set of posing configurations: adults lying on the ground and waving their arms; adults standing up waving their arms; altitude and camera viewpoint variations from the UAV, ranging from 5 to 40 m, and gimbal angles of 45° and 0° from Nadir.

Thermal preview tests were conducted between 5:30 a.m. and 7:30 a.m. from the 25th August 2021 to the 23rd September 2021, with ambient temperatures between 7°C and 14°C. An overview of the system navigating in hybrid mode to increase values of object detection confidence from streamed thermal frames can be found in Figure 14. In this instance, the UAV starts inspecting the area covered by the FOV of the thermal camera (Figure 14a). In subsequent time steps, the aircraft gets closer to have a better visual of the scene and confirm the presence of the victim by retrieving higher detection confidence values (Figure 14b). A demonstrative preview video of flight operations with thermal imagery can be found at <https://youtu.be/yIPNwNYtAo>.

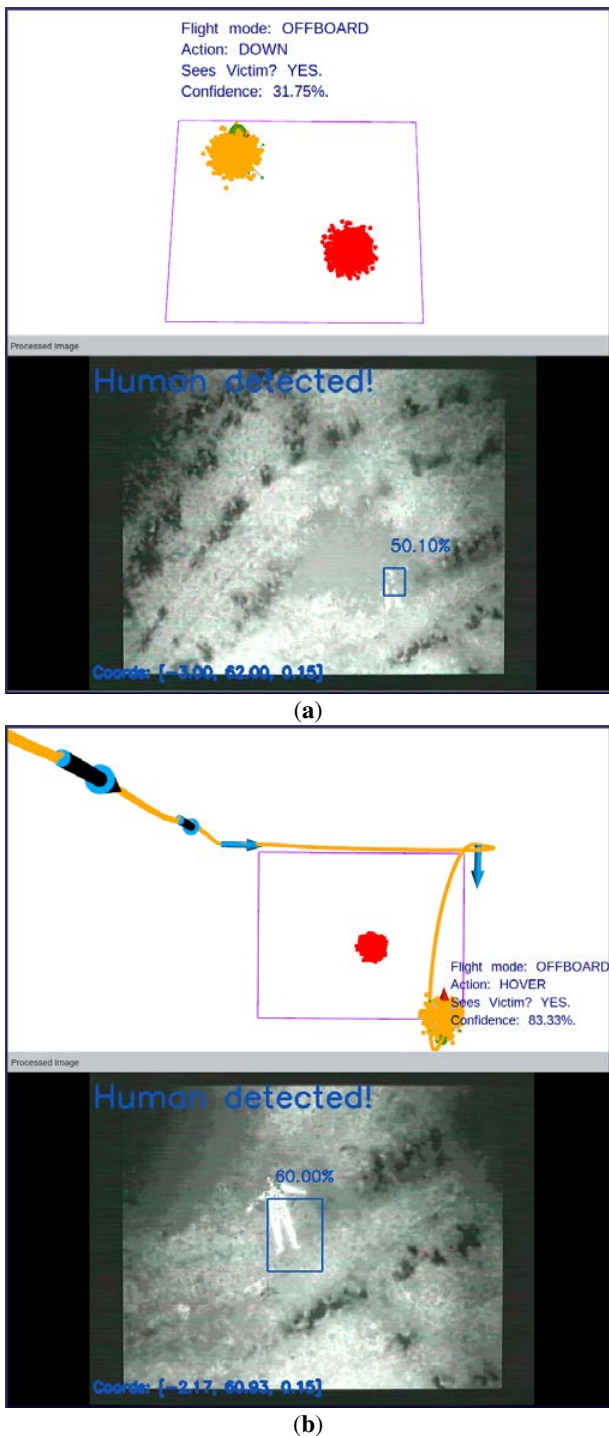


Figure 14. Reduction of object detection uncertainty from thermal camera after executing the motion policy. (a) Initial detection of a potential victim with a confidence value of 31.75%. (b) Increased confidence value (83.33%) at subsequent detections after the UAV gets closer to the victim.

The proposed UAV framework constitutes a novel approach to autonomous navigation for exploration and target finding

under uncertainty. Real-world environments are full of uncertainties such as illumination conditions, strong wind currents, collision from static and dynamic obstacles, occlusion, and limitations in object detectors, which can negatively affect the success of the flight mission. While many approaches handle object detection uncertainty by fine-tuning CNN models with labelled airborne datasets [36], this research suggests augmenting the cognition power onboard UAVs from imperfect sensor and detection output observations. Moreover, augmenting autonomy capabilities in small UAVs might open more approaches to automated surveying in outdoor environments by using a swarm of UAVs which will not require permanent supervision by pilots to photo-interpret streamed camera frames to identify and locate potential victims.

Several benefits can be offered through the use of the framework in time-critical applications to SAR squads. A first assessment of the accessibility conditions of the surveyed environment, identification, localisation, quantification and conditions of victims and external hazards might be obtained rapidly thanks to real-time telemetry of processed camera frames. After completing a flight, the list of GNSS coordinates (depicted in Figures 10 and 11) can be shared to SAR squads to coordinate better response strategies, as well as launching additional UAVs for critical deployment of medicines, food or water.

6. CONCLUSIONS

This paper discussed a modular UAV framework for autonomous onboard navigation in outdoor environments under uncertainty. The system showed how levels of object detection uncertainty were substantially reduced by calculating a motion policy using an online POMDP solver and interacting with the environment to obtain better visual representations of potential detected targets. CNN-based computer vision inference and motion planning can be executed in resource-constrained hardware onboard small UAVs. The framework design was validated with real flight tests with a simulated SAR mission, which consisted in finding an adult mannequin in an open area and close to a tree. Collected performance indicators from three flight modes suggest that the system reduces levels of object detection uncertainty in outdoor environments whether information about the surveyed environment is available. This framework was also extended by adapting the payload with a thermal camera and converting a customised people detector from thermal imagery in the vision module.

The presented framework extends the contributions of [8, 26] by: (1) extending their tested UAV framework in GNSS-denied environments for outdoor missions with GNSS signal coverage with a novel flight mode (i.e., hybrid mode); (2) further validating preliminary results of the presented framework using real flight tests; and (3) demonstrating scalability opportunities of the modular framework design by adapting a thermal camera and custom object detector to locate victims using their heat signatures.

Future work should evaluate the performance of the UAV system using tailored object detectors from different network architectures than MobileNet SSD and high-resolution streamed frames. Further assessments with multiple victims and more complex environment configurations (e.g. slope terrain, type and number of obstacles) are encouraged. A comparison study of various online POMDP solvers, or other motion planners, could improve understanding the limits

Table 4. Flight plan parameters based on RGB camera properties.

Property	Value
UAV altitude	16 m
UAV velocity	2 m/s
Lens width	2.06 mm
Lens height	1.52 mm
Camera focal length	4.7 mm
Image resolution	640 by 480 px
Overlap	30%
Bottom right waypoint	-27.3892746°, 152.8730164°
Top right waypoint	-27.3887138°, 152.8730164°
Top left waypoint	-27.3887138°, 152.8727722°
Bottom left waypoint	-27.3892765°, 152.8727722°

of using the ABT solver and POMDP solvers for motion planning under uncertainty.

APPENDIX

The survey design and flight plan parameters for conducted tests in *mission* and *hybrid* flight modes are shown in Figure 15 and Table 4, respectively.

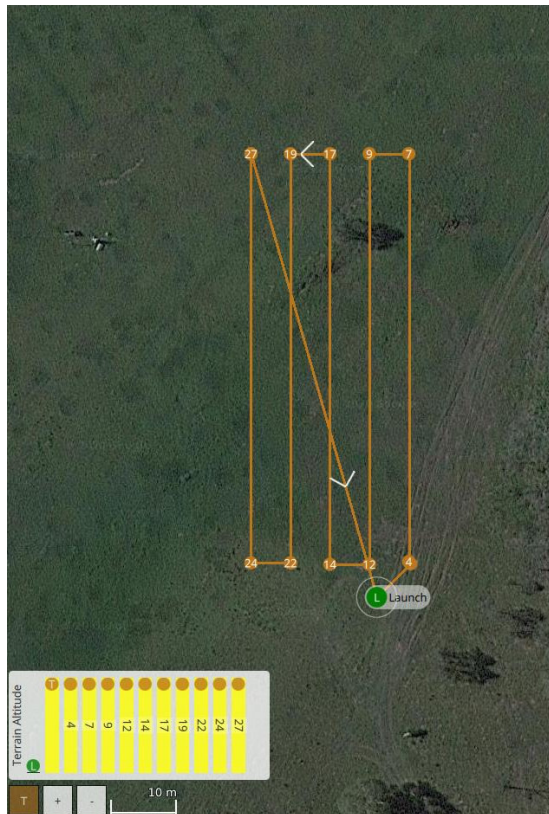


Figure 15. Flight plan following a lawnmower pattern using the RGB camera properties from Table 4.

The set of hyper-parameters used in TAPIR and initial con-

ditions to operate the UAV using offboard and hybrid modes are shown in Table 5.

ACKNOWLEDGMENTS

The authors wish to express their gratitude to Sharlene Lee-Jendili for her implementation of the thermal-based people detector used in this project. The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. Special thanks to the Samford Ecological Research Facility (SERF) team (Marcus Yates and Lorrelle Allen) for their continuous assistance and equipment provided during the flight tests. The authors would also like to gratefully thank the QUT Research Engineering Facility (REF) team (Dr Dmitry Bratanov, Gavin Broadbent, Dean Gilligan) for their technical support that made possible conducting the flight tests.

This research was funded by the Commonwealth Scientific and Industrial Research Organisation (CSIRO) through the CSIRO Data61 PhD and Top Up Scholarships (Agreement 50061686); the Australian Research Council (ARC) through the ARC Discovery Project 2018 “Navigating under the forest canopy and in the urban jungle” (grant number ARC DP180102250); and the Queensland University of Technology (QUT) through the Higher Degree Research (HDR) Tuition Fee Sponsorship. Special thanks to Hexagon through the Hexagon SmartNet RTK corrections service that enabled high accuracy surveying and positioning data using the EMLID Reach RTK receiver during the experimentation phase.

REFERENCES

- [1] L. Hanson and K. Namuduri, “Real-world applications,” in *UAV Networks and Communications*, K. Namuduri, S. Chaumette, J. H. Kim, and J. P. G. Sterbenz, Eds. Cambridge, UK: Cambridge University Press, 2017, pp. 194–213.
- [2] S. Lee, D. Har, and D. Kum, “Drone-assisted disaster management: Finding victims via infrared camera and lidar sensor fusion,” in *3rd Asia-Pacific World Congress on Computer Science and Engineering*. Nadi, Fiji: IEEE, Dec. 2016, pp. 84–89.
- [3] G. Pajares, “Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs),” *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 4, pp. 281–330, Apr. 2015.
- [4] N. H. Motlagh, M. Bagaa, and T. Taleb, “UAV-based iot platform: A crowd surveillance use case,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, Feb. 2017.
- [5] J. Jiménez López and M. Mulero-Pázmány, “Drones for conservation in protected areas: Present and future,” *Drones*, vol. 3, no. 1, p. 10, Jan. 2019.
- [6] M. Erdelj and E. Natalizio, “UAV-assisted disaster management: Applications and open issues,” in *International Conference on Computing, Networking and Communications*. Kauai, HI, USA: IEEE, Feb. 2016, pp. 1–5.
- [7] M. G. Pensieri, M. Garau, and P. M. Barone, “Drones as an integral part of remote sensing technologies to help missing people,” *Drones*, vol. 4, no. 2, p. 15, Apr. 2020.
- [8] J. Sandino, F. Vanegas, F. Gonzalez, and F. Maire, “Autonomous UAV navigation for active perception of

Table 5. Set of hyper-parameters used in TAPIR and initial conditions to operate the UAV in offboard and hybrid flight modes.

Variable	Description	Value
z_{\max}	Maximum UAV altitude	16 m
z_{\min}	Minimum UAV altitude	5.25 m
p_{u0}	Initial UAV position	(-27.3897972°, 152.8732300°, 20 m)
φ_u	UAV Heading	0°
δ_z	UAV climb step	2 m
λ	Frame overlap	40%
α	Camera pitch angle	0°
ζ_{\min}	Minimum detection confidence	10%
ζ	Confidence threshold	85%
γ	Discount factor	0.95
Δt	Time step interval	4 s
t_{\max}	Maximum flying time	10 min

targets in uncertain and cluttered environments,” in *Aerospace Conference*. Big Sky, MT, USA: IEEE, Mar. 2020, pp. 1–12.

- [9] S. Macdonald and A. Stevens, “How to explore planets with drones,” *Astronomy & Geophysics*, vol. 59, no. 3, pp. 3.18–3.22, Jun. 2018.
- [10] M. Hassanalian, D. Rice, and A. Abdelkefi, “Evolution of space drones for planetary exploration: A review,” *Progress in Aerospace Sciences*, vol. 97, pp. 61–105, Feb. 2018.
- [11] Australian National Search and Rescue Council, “Volume 2 - search and rescue operations,” in *National Search and Rescue Manual*, 2020th ed. Canberra, Australia: Australian Maritime Safety Authority (AMSA), 2020, no. February, ch. 3, pp. 73–353.
- [12] S. Bricknell, “Missing persons: Who is at risk?” Australian Institute of Criminology, Canberra, Australia, Tech. Rep., 2017.
- [13] U. Ilhan, L. Gardashova, and K. Kilic, “UAV using dec-POMDP model for increasing the level of security in the company,” *Procedia Computer Science*, vol. 102, pp. 458–464, 2016.
- [14] M. Chen, E. Frazzoli, D. Hsu, and W. S. Lee, “POMDP-lite for robust robot planning under uncertainty,” in *International Conference on Robotics and Automation*. Stockholm, Sweden: IEEE, Feb. 2016, pp. 5427–5433.
- [15] C. Ponzoni Carvalho Chanel, A. Albore, J. T’Hooft, C. Lesire, and F. Teichteil-Königsbuch, “Ample: an anytime planning and execution framework for dynamic and uncertain problems in robotics,” *Autonomous Robots*, vol. 43, no. 1, pp. 37–62, Jan. 2019.
- [16] F. Vanegas and F. Gonzalez, “Uncertainty based online planning for UAV target finding in cluttered and GPS-denied environments,” in *Aerospace Conference*, vol. 2016-June. Big Sky, MT, USA: IEEE, Mar. 2016, pp. 1–9.
- [17] F. Vanegas, D. Campbell, M. Eich, and F. Gonzalez, “UAV based target finding and tracking in GPS-denied and cluttered environments,” in *International Conference on Intelligent Robots and Systems*. Daejeon, South Korea: IEEE/RSJ, Oct. 2016, pp. 2307–2313.
- [18] J. Sandino, F. Vanegas, F. Maire, P. Caccetta, C. Sander-son, and F. Gonzalez, “UAV framework for autonomous onboard navigation and people/object detection in cluttered indoor environments,” *Remote Sensing*, vol. 12, no. 20, p. 3386, Oct. 2020.
- [19] S. Ragi and E. K. P. Chong, “UAV path planning in a dynamic environment via partially observable markov decision process,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2397–2412, Oct. 2013.
- [20] —, “UAV guidance algorithms via partially observable markov decision processes,” in *Handbook of Unmanned Aerial Vehicles*, K. Valavanis and G. Vachtsevanos, Eds. Dordrecht: Springer Netherlands, 2015, ch. 73, pp. 1775–1810.
- [21] S. Mayer, L. Lischke, and P. W. Woźniak, “Drones for search and rescue,” in *1st International Workshop on Human-Drone Interaction*. Glasgow, United Kingdom: Ecole Nationale de l’Aviation Civile [ENAC], May 2019, pp. 1–7.
- [22] K. P. Valavanis and G. J. Vachtsevanos, “Future of unmanned aviation,” in *Handbook of Unmanned Aerial Vehicles*, K. P. Valavanis and G. J. Vachtsevanos, Eds. Dordrecht: Springer Netherlands, 2015, ch. 126, pp. 2993–3009.
- [23] S. Waharte and N. Trigoni, “Supporting search and rescue operations with UAVs,” in *International Conference on Emerging Security Technologies*. Canterbury, UK: IEEE, Sep. 2010, pp. 142–147.
- [24] R. Z. B. Bravo, A. Leiras, and F. L. Cyrino Oliveira, “The use of UAVs in humanitarian relief: An application of POMDP-based methodology for finding victims,” *Production and Operations Management*, vol. 28, no. 2, pp. 421–440, Feb. 2019.
- [25] A. Gupta, D. Bessonov, and P. Li, “A decision-theoretic approach to detection-based target search with a UAV,” in *International Conference on Intelligent Robots and Systems*. Vancouver, BC, Canada: IEEE, Sep. 2017, pp. 5304–5309.
- [26] J. Sandino, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez, “Drone-based autonomous motion planning system for outdoor environments under object detection uncertainty,” *Remote Sensing*, vol. 13, no. 21, p.

4481, Nov. 2021.

- [27] Y. Chuanqi, “Caffe implementation of google mobilenet ssd detection network, with pre-trained weights on voc0712 and map=0.727.” <https://github.com/chuanqi305/MobileNet-SSD>, 2020. [Online]. Available: <https://github.com/chuanqi305/MobileNet-SSD>
- [28] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *International Conference on Multimedia*. Orlando, Florida, USA: ACM Press, Nov. 2014, pp. 675–678.
- [29] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
- [30] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “Octomap: an efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.
- [31] A. Koubaa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, “Micro air vehicle link (mavlink) in a nutshell: A survey,” *IEEE Access*, vol. 7, pp. 87 658–87 680, Jun. 2019.
- [32] A. Chovancová, T. Fico, L. Chovanec, and P. Hubinsk, “Mathematical modelling and parameter identification of quadrotor (a survey),” *Procedia Engineering*, vol. 96, pp. 172–181, 2014.
- [33] H. Kurniawati and V. Yadav, “An online POMDP solver for uncertainty planning in dynamic environment,” in *Robotics Research. Springer Tracts in Advanced Robotics*, M. Inaba and P. Corke, Eds. Springer, Cham, 2016, vol. 114, pp. 611–629.
- [34] P. Bourke, “Polygons and meshes,” <http://paulbourke.net/geometry/polygonmesh>, 1997. [Online]. Available: <http://paulbourke.net/geometry/polygonmesh>
- [35] D. Klimenko, J. Song, and H. Kurniawati, “Tapir: A software toolkit for approximating and adapting POMDP solutions online,” in *Australasian Conference on Robotics and Automation*. Melbourne, Australia: ARAA, Dec. 2014, pp. 1–9.
- [36] P. Mittal, R. Singh, and A. Sharma, “Deep learning-based object detection in low-altitude UAV datasets: A survey,” *Image and Vision Computing*, vol. 104, p. 104046, Dec. 2020.

BIOGRAPHY

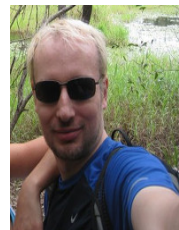


Juan Sandino is a research engineer at the Queensland University of Technology (QUT) Centre for Robotics (QCR) and CSIRO Data61, Brisbane, Australia. He holds a BEng (Mechatronics) and is undertaking a PhD in robotics and autonomous systems at QUT, Brisbane, Australia. His primary interests comprise autonomous decision-making in small Unmanned Aerial Vehicles (UAVs), machine learning and computer vision applied on UAVs for remote sensing, with a focus on hyperspectral

and high-resolution image processing. Juan has worked for research projects in biosecurity, environment monitoring and time-critical applications such as land search and rescue to find lost people in collapsed buildings and bushlands.



Peter Caccetta is a Senior Principal Research Scientist with the Commonwealth Scientific and Industrial Research Organisation. He received his Degree in Electrical Engineering from the University of Western Australia in 1989, and in 1997 his Ph.D. in Computing Science from Curtin University of Technology, Perth, Western Australia. He originally joined the CSIRO Division of Mathematics and Statistics in 1991, and subsequently worked on the development of broad-scale land cover mapping and monitoring technologies, including for continental monitoring of forests, monitoring of agricultural systems, urban and other areas. His current interests include research and development of quantitative methods and algorithms for monitoring natural and built environments using data acquired from various sensors and platforms.



Conrad Sanderson (M’01) received the Ph.D. degree from Griffith University, Nathan, QLD, Australia, in 2003, and the M.B.A. degree from the University of Queensland, Brisbane, QLD, Australia, in 2012. He has been involved in speech recognition and language translation with NICTA, Queensland Research Laboratory, QLD, Australia, in audio-visual biometrics with the IDIAP Research Institute, Martigny, Switzerland, in computer vision for military applications with the University of Adelaide, Adelaide, SA, Australia, and in natural language processing, bioinformatics, and automated surveillance with NICTA, Queensland Research Laboratory, QLD, Australia. His work on innovative surveillance technologies has led to several industry awards. He is currently a Research Leader with National ICT Australia Ltd. His current research interests include adaptation and commercialization of research outcomes toward industrial use, and machine learning, pattern recognition, and predictive analytics. Dr. Sanderson has served as a reviewer for major international conferences and scientific journals.



Frederic Maire received the M.Sc. degree in pure mathematics and computer science engineering in 1989 and the Ph.D. degree in discrete mathematics from the Université Pierre et Marie Curie, Paris 6, France, in 1993. He is currently a Senior Lecturer with the School of Electrical Engineering and Robotics, Queensland University of Technology, Brisbane, Australia. His research interests include computer vision and robotics.



***Felipe Gonzalez** is a Professor at the School of Electrical Engineering and Robotics (EER), Engineering Faculty, and CI at the QUT Centre for Robotics (QCR), with a passion for innovation in the fields of aerial robotics and automation. Gonzalez's interest is in creating aerial robots, drones or UAVs that possess a high level of cognition using efficient on-board computer algorithms*

using advanced optimisation and game theory approaches that assist us to understand and improve our physical and natural world. Professor Gonzalez leads the Airborne Sensing Lab at QUT. He is the co-author of several books in UAV based remote sensing and UAV based design based on evolutionary optimisation and game strategies and as of 2021 has published nearly 155 refereed papers.

Chapter 7

Autonomous Mapping of Desiccation Cracks via a Probabilistic-based Motion Planner Onboard UAVs

UNDERSTANDING the history and potential liveable conditions in planets such as Mars have been of interest for researchers. The identification and localisation of desiccation cracks in ancient water bodies such as lakes, rivers, and seas are a possible source to investigate past life forms. Small UAVs play a key role in the identification and mapping of desiccation cracks, normally located in remote complex environments that are difficult to identify with the naked eye.

This chapter presents the complete scalability example of the UAV framework (established in paper 4, Chapter 6) validated with real flight tests using two desiccation crack patterns distributed across an open area. It is this fifth paper that demonstrates a novel CNN model for real-time image segmentation of desiccation cracks, and a new payload to acquire RGB frames using an OAK-D camera. Experimental data indicates the autonomous decision-making UAV system has the capacity to detect and map desiccation cracks in the surveyed area.

Statement of Contribution of Co-Authors for Thesis by Published Paper

The authors listed below have certified that:

1. they meet the criteria for authorship and that they have participated in the conception, execution, or interpretation, of at least that part of the publication in their field of expertise;
2. they take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;
3. there are no other authors of the publication according to these criteria;
4. potential conflicts of interest have been disclosed to (a) granting bodies, (b) the editor or publisher of journals or other publications, and (c) the head of the responsible academic unit, and
5. they agree to the use of the publication in the student's thesis and its publication on the [QUT's ePrints site](#) consistent with any limitations set by publisher requirements.

In the case of this chapter:

J. Sandino, J. Galvez-Serna, N. Mandel, F. Vanegas, and F. Gonzalez, "Autonomous mapping of desiccation cracks via a probabilistic-based motion planner onboard UAVs," in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2022, (accepted).

Contributor	Statement of Contribution ¹
Juan Sandino QUT Verified Signature Date: <u>2022-05-20</u>	Contributed on: conceptualisation, data curation, formal analysis, investigation, methodology, software, validation, visualisation, and writing-original draft.
Julian Galvez-Serna	Contributed on: data curation, investigation, visualisation, and writing-review and editing.
Nicolas Mandel	Contributed on: data curation, formal analysis, investigation, visualisation, and writing-review and editing.
Fernando Vanegas	Contributed on: investigation, supervision, and writing-review and editing.
Felipe Gonzalez	Contributed on: conceptualisation, funding acquisition methodology, project administration, resources, supervision, and writing-review and editing.

Principal Supervisor Confirmation		
I have sighted email or other correspondence from all co-authors confirming their certifying authorship.		
<i>Felipe Gonzalez</i>	QUT Verified Signature	<u>2022-05-20</u>
Name	Signature	Date

¹ Contributions 0mm follow the CRediT taxonomy. Further details can be found at <https://credit.niso.org/>.

Autonomous Mapping of Desiccation Cracks via a Probabilistic-based Motion Planner Onboard UAVs

Juan Sandino, Julian Galvez-Serna, Nicolas Mandel, Fernando Vanegas and Felipe Gonzalez
 QUT Centre for Robotics
 Queensland University of Technology
 2 George St, Brisbane City
 QLD 4000, Australia

{j.sandino, galvezse, nicolas.mandel, f.vanegasalvarez, felipe.gonzalez}@qut.edu.au

Abstract—Studies of past life forms on other planets are possible through the identification and localisation of desiccation cracks in ancient water bodies such as lakes, rivers and seas. Unmanned aerial vehicles (UAVs) are increasingly being used as a viable remote sensing solution for planetary exploration, as desiccation cracks are difficult to identify with the naked eye and are normally located in complex and unreachable environments. However, most UAVs have a strong reliance on human operators through their communication systems, as UAVs have limited onboard decision-making capabilities for autonomous navigation in such environments. UAV navigation in real-world scenarios is also challenging as data captured from their sensors is imperfect, and outputs from computer vision systems are, sometimes, inaccurate. These sensory and onboard vision limitations cause partial observability of the state of surveyed environments, inducing uncertainty in optimal path planning. This paper proposes a UAV system for autonomous onboard navigation, identification, and mapping of desiccation cracks for planetary exploration. The navigation problem is mathematically formulated as a partially observable Markov decision process (POMDP), where a motion strategy can be obtained by solving the POMDP in real time using the augmented belief tree (ABT) solver. The framework discussed in this work is validated with real flight tests using two desiccation crack patterns distributed across the surveyed area. Real-time segmentation from streamed camera frames of desiccation cracks is achieved through inference onboard the aircraft using a ResNet18 Convolutional Neural Network (CNN) model, and an OpenCV AI Kit (OAK)-D camera. Results from real flight tests indicate that the system can reduce levels of object detection uncertainty to locate and map desiccation cracks in environments under partial observability. The system design allows further adaptation for similar time-critical applications requiring increased levels of UAV autonomy in unstructured environments under uncertainty and partial observability, such as humanitarian relief, wildlife monitoring, and surveillance.

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. FRAMEWORK DESIGN.....	2
3. POMDP MOTION PLANNER DESIGN.....	4
4. EXPERIMENTS.....	6
5. RESULTS AND DISCUSSION.....	7
6. CONCLUSIONS.....	12
APPENDIX.....	12
ACKNOWLEDGMENTS.....	12
REFERENCES.....	12
BIOGRAPHY.....	14

1. INTRODUCTION

Recent efforts in planetary exploration are focused on finding environments that suggest the presence of past or present life forms. These traces of life forms are denominated *biosignatures*. Literature suggests that some biosignatures are likely to be found in environments with the past or present presence of water, as they allow organisms to grow and be preserved in rock records (or fossils) [1]. An increasingly popular type of sedimentary structure is the inspection of mud cracks (or desiccation cracks) from past aquatic environments such as dry lake beds. Environments on Earth with some present biosignature indicators are located in Pilbara, WA, Australia. An illustration of biosignature fossils and desiccation cracks is shown in Figure 1.



(a)



(b)

Figure 1. Environments with indicators of water-related life signatures (or biosignatures). (a) Ancient conical stromatolite fossils at Pilbara, WA, Australia. (b) Desiccation crack formation that may contain biosignatures.

Advances in unmanned aerial vehicle (UAV) technology for civilian applications have resulted in their gradual adoption in fields such as biosecurity, search and rescue (SAR), and wildlife monitoring [2–6]. Popularity of UAVs is also increasing in applications for planetary exploration, owing to their extensive payload adaptability, autonomous flying control capabilities, and flexibility to explore challenging and otherwise inaccessible environments. A recent case study is, for instance, the successful deployment and remote operation of the National Aeronautics and Space Administration (NASA) Ingenuity helicopter [7] on Mars for photogrammetry [8, 9]. Furthermore, the development and use of DragonFly—a UAV deployed to survey Saturn’s moon Titan—shows new opportunities to expand the use of UAVs in a new set of planetary exploration applications [10, 11].

One of the ever-present challenges in autonomous UAV navigation for real-world and planetary exploration missions is dealing with uncertainty. Sources of uncertainty come from external and internal factors that could compromise the behaviour of UAVs and the flight mission itself [12]. Some external factors include strong wind currents, solar radiation, unknown environmental conditions, and partial observability. Internal factors include noisy camera frames during streaming, suboptimal camera settings and calibration, low image resolution, and imperfect detection outputs from computer vision detectors. These factors cause partial observability of the actual state of the surveyed environment, targets of interest, or the UAV itself, which increases the decision-making complexity to undertake optimal sequences of actions and accomplish the mission. It is essential, therefore, that UAVs should incorporate cognitive capabilities for autonomous navigation in challenging planetary environments such as Mars, Titan, and Venus [13].

Literature on applied decision-making theory in UAVs is extensive and suggests that the mathematical modelling of navigation problems using partially observable Markov decision processes (POMDPs) can increase cognitive capabilities of UAVs in environments under uncertainty [14–16]. UAV frameworks for object detection and tracking have been tested in cluttered indoor environments and in the absence of global navigation satellite system (GNSS) coverage [6, 17–19]. Moreover, POMDPs have also been applied to solve multi-objective problems in UAVs, capable of addressing tasks such as path planning, multiple object detection and tracking, and collision prevention [20, 21].

Research works on applied decision-making theory for planetary exploration, and specifically, for the detection and mapping of desiccation cracks, are scarce. Recent advances in platforms such as Ingenuity have contributed to algorithms for flight control and vision-based navigation using feature detectors and data fusion [22]. However, the UAV uses a vision-based navigation algorithm limited to follow position waypoints from a pre-defined flight plan transmitted from Earth. Another significant contribution discussed a collaborative robotics approach for path planning on Mars in simulation, where aerial footage from the UAV could reduce localisation uncertainty in a ground robot for optimal navigation [23].

This paper presents a UAV framework and motion planner for autonomous navigation in planetary exploration environments. The UAV system is evaluated with a case study of biosignatures search through the localisation, segmentation and mapping of desiccation cracks. This work discusses the design and implementation of a novel flight mode that aug-

ments the capabilities of traditional mission surveys in small UAVs. The flight mode inspects dedicated areas to reduce levels of object detection and segmentation uncertainty with the execution of an online POMDP solver that calculates a motion policy in challenging environments. This policy allows the UAV to interact with the environment and obtain improved visualisations of potential detected objects.

This investigation extends the work of Sandino et al. [6, 24, 25] with the following novel contributions; (1) an extension and scalability study of their UAV framework tested in an application beyond the scope of SAR operations; (2) additional validations of their framework from simulation by executing real flight tests; and (3) an evaluation of scalability capabilities by adapting a different payload (an OpenCV AI Kit (OAK)-D camera), and a custom ResNet18 Convolutional Neural Network (CNN) model for detection and mapping of desiccation cracks into the existing framework design;

The rest of the manuscript is structured as follows: Section 2 details the design of the UAV framework for autonomous detection and mapping of desiccation cracks; Section 3 provides an overview of the probabilistic-based motion planner using a POMDP; The design of conducted experiments using real flight tests is presented in Section 4; Obtained results and discussion of performance indicators are provided in Section 5; In Section 6, conclusions and future avenues for research are discussed.

2. FRAMEWORK DESIGN

This implementation uses a modular framework design applicable for sub 2 kg UAVs. The components of this architecture are adapted from a tested framework design by Sandino et al. [19]. The original study validated the framework in cluttered environments, absent from GNSS signal and with a case study of SAR. In contrast, the design proposed in this paper extends the scope of that framework implementation with a novel application of autonomous navigation and mapping of desiccation cracks for planetary exploration. A high-level concept of the framework architecture is shown in Figure 2.

The proposed framework design allocates key components as modules, where the *Vision* and *Planner* modules run onboard a companion computer, and the low-level flight controller and associated drivers in an autopilot. Starting from the physical UAV frame, raw camera frames of the environment (which may contain areas of potential desiccation cracks) are captured with a vision-based sensor (i.e., an OAK-D camera). Streamed frames—also called observations—are visual representations of the environment and are processed in the *Vision* module. The *Vision* module contains a CNN-based semantics detector which processes input frames and returns segmentation outputs. Computational load in the computer while performing CNN model inference onboard the UAV is allocated to a *Vision Processing Unit (VPU)*.

Collected observations from the *Vision* module and the autopilot data (e.g., local and global position estimations of the UAV) are parsed into the *Observation Server* at the *Planner* module. This server is iteratively called by the POMDP-based planner while estimating and updating a motion policy mid-flight. Action commands returned by the planner are received by the *Motion Server*, which interfaces with the flight controller of the autopilot. The discussion below covers in detail each one of the components of the UAV framework.

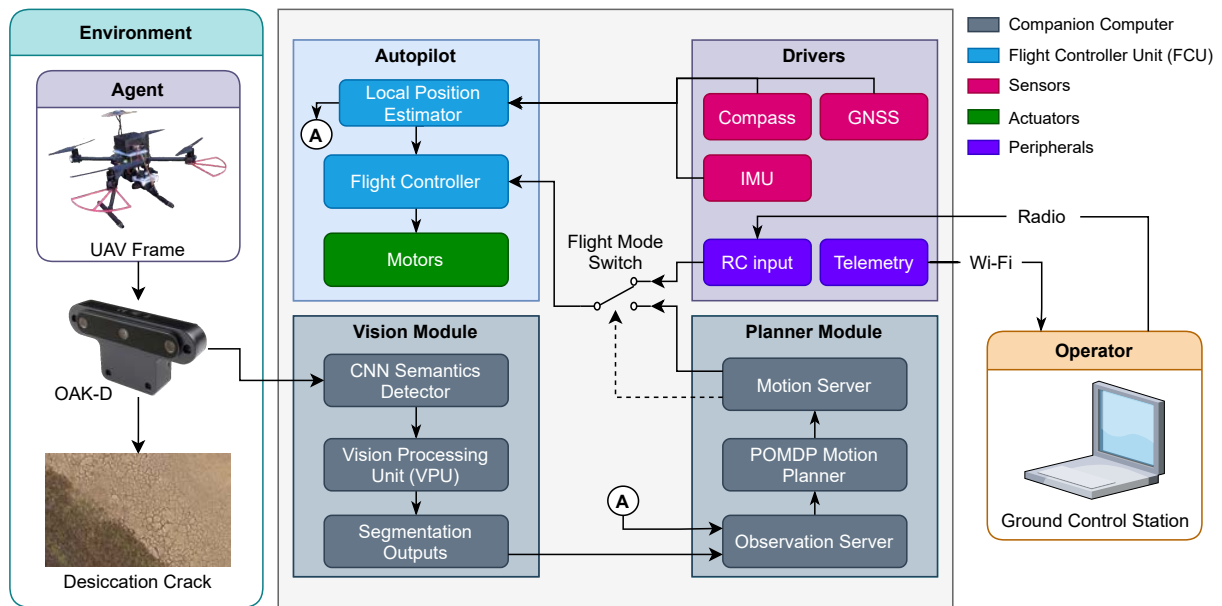


Figure 2. Framework architecture for autonomous navigation onboard unmanned aerial vehicles (UAVs) in uncertain outdoor environments. The design allocates key components as modules. The *Vision* and *Planner* modules run onboard a companion computer, and the flight controller and associated drivers in an autopilot.

UAV and Sensors

This research converged on conducting the flight tests using a Holybro X500 quadrotor kit (Holybro, China)—depicted in Figure 3—as it offers the best combination among customisation of mounted payloads, weight, size, and endurance. Essential components added to the frame include a Pixhawk 4 GNSS receiver, a Pixhawk 4 autopilot, and a 433 MHz telemetry radio. Four-cell 5000 mAh LiPo batteries are used to power the aircraft and achieve an approximate flight autonomy of 12 minutes, for an overall weight of 1.92 kg.

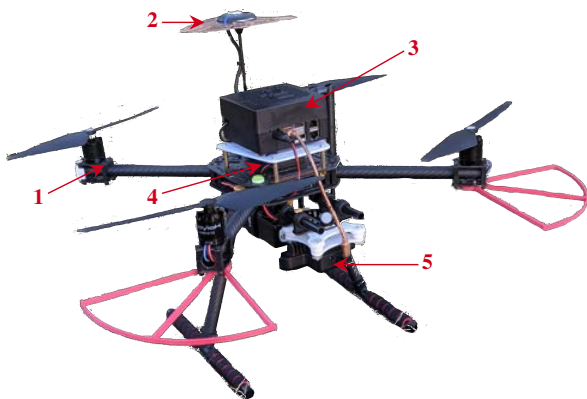


Figure 3. Sub 2 kg UAV framework implementation for autonomous detection and mapping of desiccation cracks. The UAV is composed of: 1) a Holybro X500 drone kit; 2) a Pixhawk 4 GNSS receiver; 3) an Intel UP² companion computer; 4) a Pixhawk 4 autopilot; and 5) an OpenCV AI Kit (OAK)-D sensor.

The vision-based payload is an OAK-D camera, which has an in-built red, green, blue (RGB) sensor, and two high-speed monochrome sensors to offer depth data (through internal

image processing). The camera is mounted to an anti-vibration mount and positioned down-facing and in parallel to Earth’s nadir, as depicted in Figure 4. Core properties of these sensors can be found in Table 3 of the paper’s Appendix.

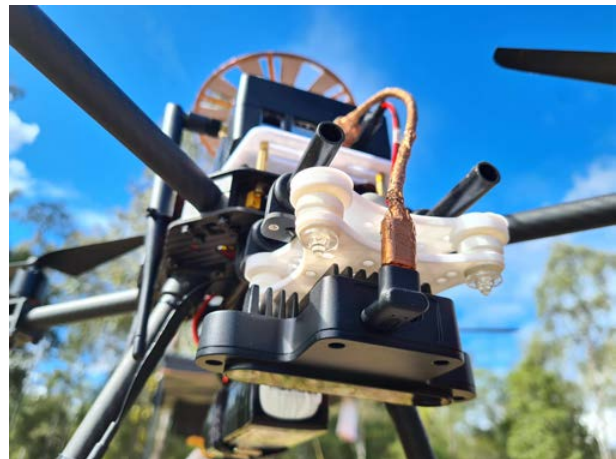


Figure 4. OAK-D camera mounted to an anti-vibration mount, down-facing and in parallel to Earth’s nadir.

Computational resources to execute the POMDP-based planner and computer vision are allocated to an Intel UP² onboard the small UAV. This companion computer features an Intel 64-bit quad-core CPU at 1.1 GHz, 8 GB DDR3 RAM, 64 GB eMMC SSD, four FL110 USB 3.0 connectors, and two High-Speed UART controllers.

Semantic Segmentation Model

Semantic segmentation is performed onboard the UAV using a custom implementation of a ResNet18 CNN architecture [26, 27]. Segmentation is chosen over detection in this

application as it allows fine-grained pixel-level information, which is essential for mapping. In this implementation, the network receives RGB image frames with a resolution of 192×256 pixels, and returns single-channel segmented frames of 96×128 pixels. Further details about the model architecture, image datasets, labelling and training procedure and evaluations can be found in [28]. Figure 5 shows an example image of the detector during the experiments, with regions depicted in red indicating the detected desiccation crack.



(a)



(b)

Figure 5. Inference preview of ResNet18 CNN model for real-time semantic segmentation of desiccation cracks [28]. (a) Retrieved (i.e., raw) OAK-D camera frame. (b) Output frame of segmented desiccation cracks (in red) and other classes (in black).

The tuned CNN model is loaded into an Intel® Movidius™ Myriad™ X VPU, embedded in the OAK-D camera for real-time inference onboard the UAV. This portability process required the use of the Intel OpenVINO toolkit to convert and optimise the CNN model into code instructions compatible with the VPU. The Vision module is programmed using the Python programming language, and the DepthAI and OpenCV libraries [29]. With the model loaded in the VPU and connected to the UP² via USB 3.0, the system achieves up to 24 frames per second (FPS) of instantaneous semantic segmentation.

Communication Interface

The UAV framework runs using open-source software tools. The companion computer runs Linux Ubuntu 18.04 LTS and the robot operating system (ROS), melodic version. ROS is a middleware to communicate between the nodes of each module (following the architecture design from Figure 2). The Pixhawk 4 autopilot is powered by the PX4 firmware and communicates with the companion computer through MAVROS via UART. MAVROS is a ROS implementation of the MAVLink protocol, which is industry standard for UAV communication and control [30].

3. POMDP MOTION PLANNER DESIGN

The POMDP-based planner translates UAV position commands derived from environment observations. The discussion presented in this section is heavily based on the problem formulation of UAV navigation under object detection uncertainty from Sandino et al. [24], and only essential and adapted components to detect and map desiccation cracks are shown below for completeness.

A POMDP [31] is a tuple $\langle A, S, T, O, \mathcal{Z}, R, b_0, \gamma \rangle$, where A, S, O are a finite set of UAV actions, states, and observations respectively. Whenever the UAV (also known as the *agent* in POMDP theory) takes an action $a \in A$ from a state $s \in S$, the transition probability to a new state $s' \in S$ is defined by a transition function $T(s, a, s') = \mathbb{P}(s' | s, a)$.

With a taken action $a \in A$, the UAV receives an observation $o \in O$ encoded by the observation function $\mathcal{Z}(s', a, o) = \mathbb{P}(o | s', a)$. Every decision chain is then quantified with a reward r , calculated using the reward function $R(a, s)$. Taking into account that collected data by the UAV sensors and vision systems is imperfect, partial observability about the state of the environment, the UAV itself and objects of interest is always present in real-world applications. As a result, a POMDP uses probability distributions over the system states to model uncertainty of its observed states. This modelling is denominated the belief $b(H) = \mathbb{P}[s^1 | H], \dots, \mathbb{P}[s^n | H]$, where H is the history of actions, observations and rewards the UAV has accumulated until a time step t , or $H = a_0, o_1, r_1, \dots, a_{t-1}, o_t, r_t$.

The motion policy π of the UAV is represented by mapping belief states to actions $\pi : b \rightarrow A$. A POMDP is solved after finding the optimal policy π^* , calculated as follows:

$$\pi^* := \arg \max_{\pi} \left(\mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R(S_t, \pi(b_t)) \right] \right), \quad (1)$$

where $\gamma \in [0, 1]$ is the discount factor and defines the relative importance of immediate rewards compared to long-term rewards. A given POMDP solver starts planning from an initial belief b_0 , which is usually defined from the initial conditions (and assumptions) of the flight mission using probabilistic distributions.

Assumptions

The formulated problem for exploration and object segmentation (i.e., mapping of desiccation cracks) using multi-rotor UAVs assumes:

- Observations come from segmented camera frames (from

the Vision module), and the estimated local UAV position (from the autopilot).

- Only a single desiccation crack can be detected at the same time. If more crack blobs are detected and depicted, the biggest blob will be chosen, and the rest are discarded.
- The position coordinate of a desiccation crack is defined by calculating the centroid of its segmented blob.
- The navigation task assumes the UAV is positioned at a known position setpoint inside the flight survey.
- The task finishes computing a motion policy once: 1) the UAV detects a crack blob whose detection confidence surpasses a set threshold; 2) the UAV explores the whole search area extent without finding any crack patterns; or 3) the UAV exceeds the maximum flight time on air (because of low levels of battery power).

Actions

The finite set of actions are defined by seven UAV position commands, namely *forward*, *backward*, *left*, *right*, *up*, *down*, and *hover*. UAV actions that are not included in the action space but are managed by the autopilot instead include, but are not limited to, *arm*, *disarm*, *take-off*, *return to launch and land*. Each action updates the position set point of the UAV in the world coordinate frame by calculating and applying a change of position δ .

The magnitude for δ_x and δ_y depends on the estimated overlap value between camera frame observations, calculated using Equation (2):

$$\delta = l(1 - \lambda), \quad (2)$$

where l is the projected length of the camera's Field of View (FOV), and $\lambda \in [0, 1]$ is the desired overlap.

States

A system state $s \in S$ is defined as:

$$s = (p_u, f_{\text{roi}}, f_{\text{dct}}, p_c, c_c) \quad (3)$$

where p_u is the position of the UAV in the world coordinate frame, f_{roi} is a flag indicating whether the UAV flying beyond the flying limits, f_{dct} is the flag raised if a potential desiccation crack is detected. If $f_{\text{dct}} = \text{True}$, the position of the crack blob in the world coordinate frame is given in p_c , with $c_c \in [0, 1]$ providing the corresponding detection confidence. The system reaches a terminal state whenever $c_c \geq \zeta$, where ζ is the confidence threshold.

Transition Function

The motion dynamics of a multi-rotor UAV defines the transition from current to new states:

$$p_u(k+1) = p_u(k) + \Delta p_u(k) \quad (4)$$

where $p_u(k)$ is the position of the UAV at time step k ; and $\Delta p_u(k)$ is the position change of the UAV between time steps. This formulation does not contain any actions for heading changes. However, Equation (4) can be expanded if required by adding the rotation matrix in multi-copters [32]. An illustration of a problem formulation including the rotation matrix can be found in [6].

Reward Function

The expected reward r after taking an action $a \in A$ from state $s \in S$ is calculated using the reward function $R(a, s)$ defined in Algorithm 1 and Table 1. This function critically influences the UAV behaviour during flight missions, and its definition allows multi-objective task definition. A grid search technique was applied for a few initial iterations of the system in simulation to obtain the applied reward values of Table 1. Complete details on the design considerations of the reward function can be found in [24].

Algorithm 1 Reward function R for exploration and inspection of desiccation cracks.

```

1:  $r \leftarrow 0$ 
2: if  $f_{\text{crash}}$  then
3:    $r \leftarrow r_{\text{crash}}$  ▷ UAV crashing cost
4: else if  $f_{\text{roi}}$  then
5:    $r \leftarrow r_{\text{out}}$  ▷ Beyond safety limits cost
6: else if  $f_{\text{dct}}$  then
7:    $r \leftarrow r_{\text{dct}}$  ▷ Detected object reward
8:    $r \leftarrow r + \left[ r_{\text{dct}} \cdot \left( 1 - \frac{z_u - z_{\text{min}}}{z_{\text{max}} - z_{\text{min}}} \right) \right]$  ▷ UAV altitude reward
9:   if  $c_c \geq \zeta$  and  $a = \text{Down}$  then
10:     $r \leftarrow r + r_{\text{conf}}$ 
11:   end if
12: else
13:    $r \leftarrow r_{\text{action}}$  ▷ Action cost
14:    $r \leftarrow r - \left[ r_{\text{dct}} \cdot \left( 1 - \frac{z_u - z_{\text{min}}}{z_{\text{max}} - z_{\text{min}}} \right) \right]$  ▷ UAV altitude cost
15:    $r \leftarrow r - \left[ r_{\text{dct}} \cdot \left( 1 - 0.5^{4 \cdot d_v / d_w} \right) \right]$  ▷ Horizontal distance cost
16:    $r \leftarrow r + r_{\text{fov}} \cdot \varepsilon$  ▷ Footprint overlap cost
17: end if
18: return  $r$ 

```

Table 1. Applied reward values to the reward function R , defined in Algorithm 1.

Variable	Value	Description
r_{out}	-25	Cost of UAV breaching safety limits
r_{dct}	+25	Reward for detecting potential crack
r_{conf}	+50	Reward for confirmed crack
r_{action}	-2.5	Cost per action taken
r_{fov}	-5	Footprint overlapping cost

Observations

An observation $o \in O$ is defined as follows:

$$o = (o_{p_u}, o_{\text{dct}}, o_{p_c}, o_{\zeta}), \quad (5)$$

where o_{p_u} is the estimated position of the UAV by the autopilot; o_{dct} is the flag triggered by potential segmentations of desiccation cracks received by the CNN model; o_{p_c} is the local position of the crack blob; and o_{ζ} is the detection confidence. As segmented desiccation cracks are defined by irregular shapes, o_{p_c} is estimated by calculating the centroid of the blob's bounding rectangle box.

The detection confidence o_{ζ} measures the frequency of positive detections between the last two observation calls:

$$o_{\zeta} = \frac{\sum_{i=1}^n o_{\text{dtc}_i}}{n}, \quad (6)$$

where n is the number of segmented frames between observation calls, and o_{dtc_i} is the flag indicating a positive detection per processed frame i .

Observation Model

This implementation uses augmented belief trees (ABT) [33], an online POMDP solver that contains a model that generates T and Z using a modelled observation o given an action a and the next state s' . The variables contained in the generative model are the position of the UAV s'_{p_u} , the centroid of the desiccation crack segment s'_{p_c} and the detection confidence o_{ζ} .

Potential detections of desiccation cracks and their position belief estimations are determined by the position of the camera and its projected FOV. This implementation assumes that a desiccation crack might be detected if any projected two-dimensional (2D) position coordinates $s'_{p_c}(x, y)$ are contained within the camera's FOV. The 2D projected extent l of a vision-based sensor can be calculated using Equations (7) and (8):

$$l_{\text{top, bottom}} = s'_{p_u}(z) \cdot \tan\left(\alpha \pm \tan^{-1}\left(\frac{h}{2f}\right)\right), \quad (7)$$

$$l_{\text{left, right}} = s'_{p_u}(z) \cdot \tan\left(\pm \tan^{-1}\left(\frac{w}{2f}\right)\right), \quad (8)$$

where $s'_{p_u}(z)$ is the altitude of the UAV, α and β are the camera's pointing angles from the vertical z and horizontal x axis of the World coordinate frame, w is the lens width, h is the lens height, and f is the focal length.

The footprint corners c from the camera's local coordinate frame I are translated to the world's coordinate frame W using the following transformation:

$$\begin{bmatrix} c'(x) \\ c'(y) \end{bmatrix} = \begin{bmatrix} s'_{p_u}(x) + c(x) \\ s'_{p_u}(y) + c(y) \end{bmatrix}. \quad (9)$$

The detection confidence o_{ζ} is modelled using Equation (10):

$$o_{\zeta} = \frac{(1 - \zeta_{\min})(d_{uc} - z_{\min} + \zeta_{\min})}{z_{\max} - z_{\min}}, \quad (10)$$

where ζ_{\min} is the minimally accepted confidence threshold; z_{\max} and z_{\min} are the maximum and minimum UAV flying altitudes respectively; and d_{uc} is the Manhattan distance between the UAV and the desiccation crack.

4. EXPERIMENTS

This work evaluated the UAV framework design presented in Section 2 using a sub 2 kg quadcopter in an emulated planetary exploration environment. The experiments aim to compare the performance of the POMDP-based planner

(discussed in Section 3) with a baseline motion planner to record global coordinates and maps of desiccation cracks. The following subsections describe the environment setup, the tested flight modes for data acquisition and analysis, and the implemented POMDP-based motion solver.

Environment Setup

Field experiments were conducted at the Samford Ecological Research Facility (SERF), Samford Village, QLD, Australia. The property contains a mostly flat area utilised to emulate a space exploration environment. Two desiccation cracks textures were placed across a 50 m × 50 m surveyed area. The performance of the detector was further tested by placing obstacles including a car, a tree, and a chair, as illustrated in Figure 6.

Flight Modes

The UAV navigation and mapping tasks are evaluated using two flight modes at flight altitudes of 10 m and 20 m. The first mode—denominated *mission* mode from hereon—is an automatic flight mode provided by the PX4 firmware, which controls the motion of the UAV by following a series of position setpoints. These setpoints are global position coordinates that, when combined, shape a survey pattern. The pattern follows a lawnmower pattern which lets the UAV survey the flying area extent at a constant altitude. The space between transects is defined by the desired overlap between OAK-D camera frames. Specific details of the survey pattern can be found in Table 3 of the Appendix. An example lawnmower-pattern survey for a constant altitude of 10 m is illustrated in Figure 7. When the UAV operates in mission mode, the semantics detector runs onboard to record any desiccation crack blobs while the UAV is surveying the emulated planetary environment. This mode serves as the baseline motion planner of this research because of its out-of-the-box implementation provided for many UAVs.

The probabilistic-based motion planner proposed in this paper runs onboard the UAV through a novel flight mode design. This flight mode—denominated *hybrid* mode from hereon—extends the functionality of *mission* mode by incorporating autonomous inspection of potential zones with desiccation cracks. In hybrid mode, the POMDP planner starts inspecting a zone after a first crack detection is triggered by the semantics detector. As shown in Figure 8, the inspection is limited to the extent of the camera's FOV, where the planner calculates a motion policy to confirm or discard the presence of desiccation cracks. Once the planner reaches a terminal state (i.e., the UAV discards or confirms a desiccation crack), the UAV resumes its survey in mission mode. The process repeats for future detections until the UAV completes the survey mission.

POMDP solver

The companion computer calculates an approximate solution of the POMDP-based navigation problem by using the TAPIR toolkit [34]. This toolkit contains an implementation of the ABT solver programmed in C++. A ROS node was developed to integrate the solver into the proposed framework and transport action messages to the motion server.

For every solver call in hybrid flight mode, TAPIR calculates an initial motion policy for four seconds. An observation is then retrieved to update the motion policy and select the action command that provides the highest expected accumulated reward. Once the node sends the action message to the Motion Server (and subsequently to the autopilot), the solver

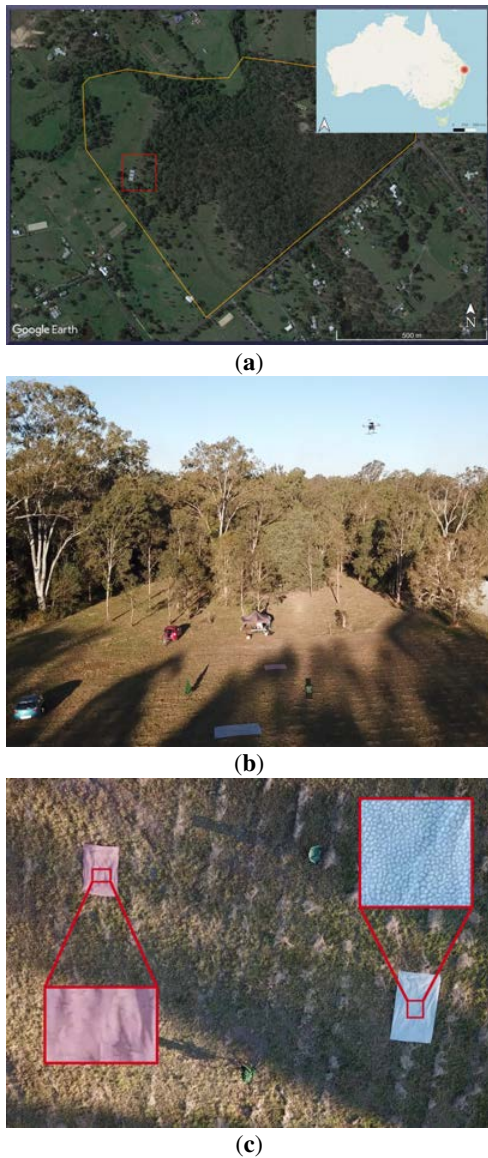


Figure 6. Test setup at the Samford Ecological Research Facility (SERF), Australia. (a) Boundary extents of SERF (orange) and flight area (red). (b) Test area composed of desiccation crack textures, and obstacles such as trees, chairs, and cars. (c) Printed textures of desiccation cracks.

applies an idle time of 3.4 seconds to wait for the UAV to reach the desired position. The process repeats by collecting a new observation, updating the policy and selecting the next action command until the planner reaches a terminal state. Further details about the solver implementation can be found in [24].

5. RESULTS AND DISCUSSION

The proposed UAV framework for autonomous detection and mapping of desiccation cracks is tested using the following indicators: 1) heatmaps of spatial distribution of recorded GNSS coordinates of crack textures; 2) accuracy metrics



Figure 7. Drafted flight plan in QGroundControl following a lawnmower pattern using the OAK-D camera properties from Table 3.

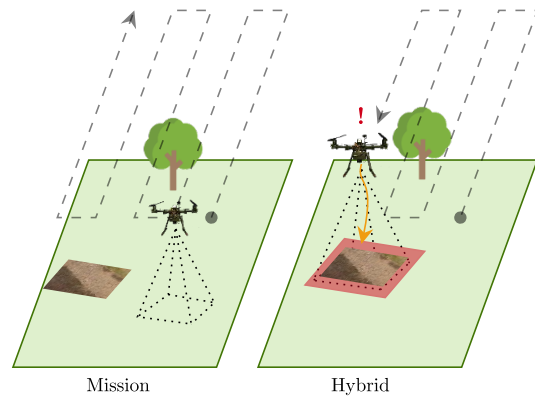


Figure 8. Executed flight modes for detection and mapping of desiccation cracks. Mission mode is the baseline motion planner and lets the UAV survey the emulated area following a lawnmower pattern. Hybrid mode extends the functionality of mission mode by running the POMDP-based planner to inspect the area for potential desiccation cracks.

of the UAV framework in mission and hybrid flight modes; 3) speed analysis of the proposed UAV system in hybrid mode to locate desiccation cracks at 10 and 20 m; and 4) comparison of compiled Quadtree maps of segmented crack textures between mission and hybrid flight modes. A real flight video demonstration of this framework implementation can be found at <https://youtu.be/2Voal0T-RWw>.

Visualisation of data noise and spatial distribution of collected GNSS coordinates between mission and hybrid flight modes is assessed using heatmaps. As depicted in Figures 9 and 10, the heatmaps suggest a strong noise reduction and improved accuracy to locate desiccation crack textures while

the UAV operates in hybrid mode.

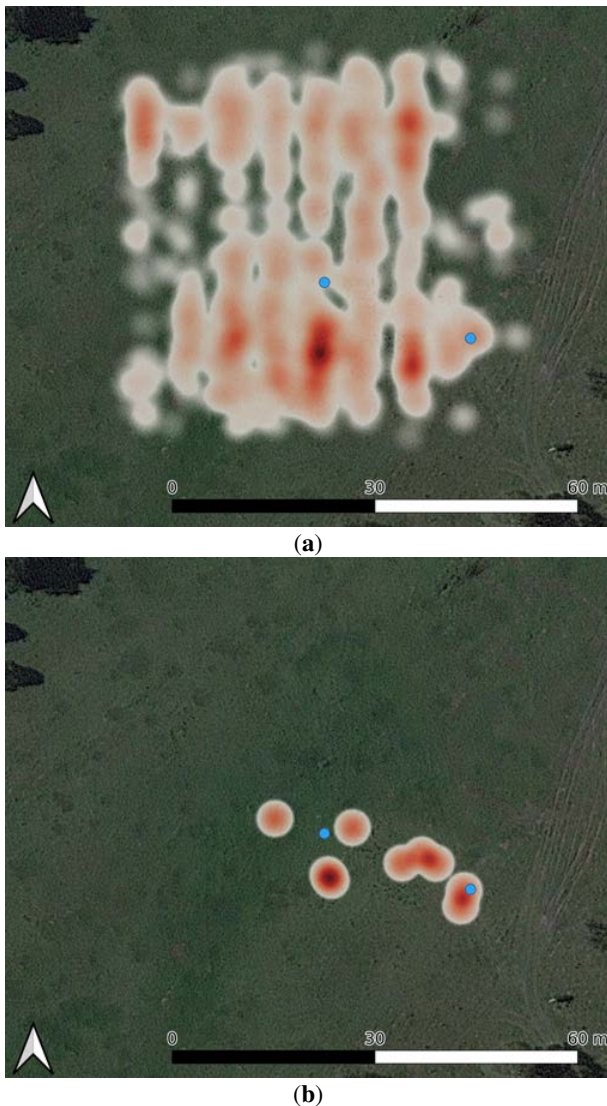


Figure 9. Comparison of recorded GNSS coordinates of desiccation cracks for UAV surveys at 10 m above ground level (AGL). The blue dots represent the actual location of the crack textures. (a) Heatmap of recorded detections in mission mode. (b) Heatmap of recorded detections in hybrid mode.

The proposed POMDP-based planner—adapted from Sandino et al. [19]—for semantic object segmentation and outdoor UAV navigation, reduces levels of detection uncertainty in flight tests using hybrid mode. By using hybrid mode, the UAV gains cognitive power to autonomously inspect areas that may contain desiccation cracks. As CNN-based detectors are not exempt from providing false positive readings of detected objects, this implementation provides higher flexibility to the UAV to interact with the environment and confirm or discard the presence of desiccation cracks. A case study of the UAV inspecting an area with a present crack texture is shown in Figure 11.

In the event of a positive detection of an existing desiccation

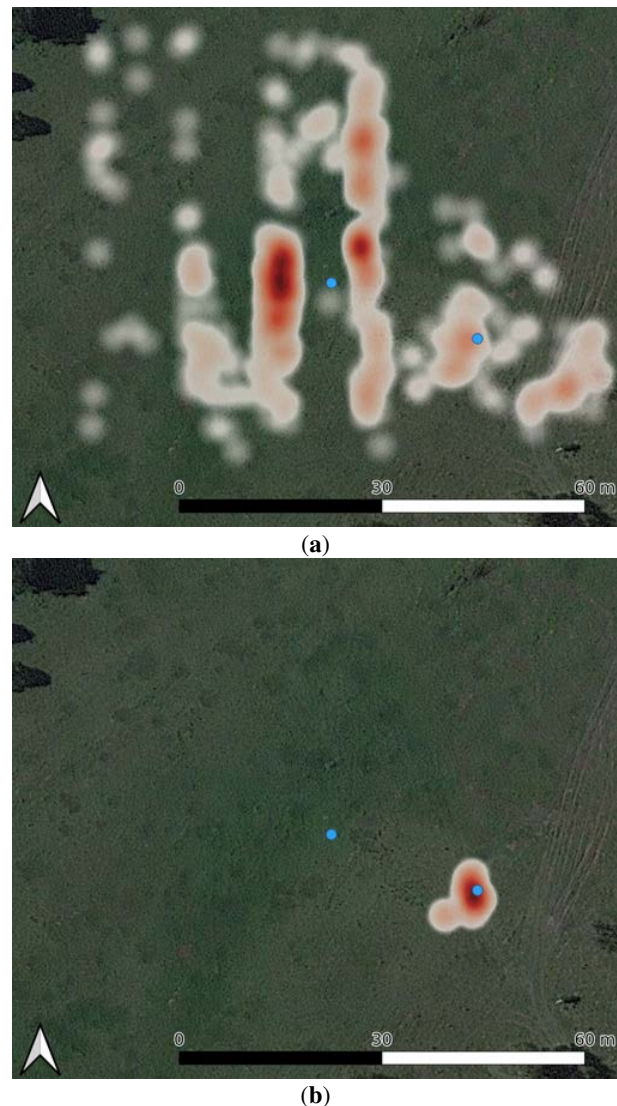
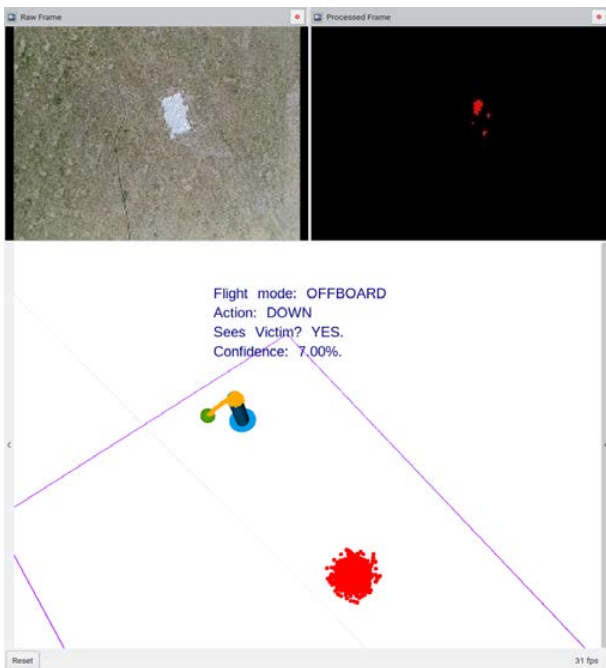
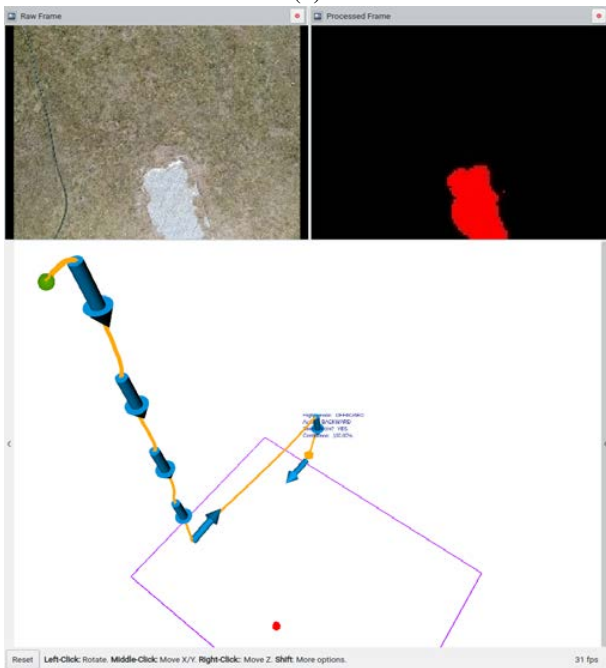


Figure 10. Comparison of recorded GNSS coordinates of desiccation cracks for UAV surveys at 20 m AGL. The blue dots represent the actual location of the crack textures. (a) Heatmap of recorded detections in mission mode. (b) Heatmap of recorded detections in hybrid mode.

crack within the camera's FOV, the POMDP-based planner computes an initial motion policy and arranges the probabilistic distribution belief of the texture location based on the initial collected observation (Figure 11a). As the detection confidence at the beginning of the inspection is usually low (i.e., 7.0%), the UAV will interact with the environment based on the motion policy by the POMDP planner. After a series of sequential actions, the UAV has a better visualisation of the desiccation crack texture by approaching it until achieving a detection confidence of 100% (Figure 11b). Adjustments in the motion policy could occur from unexpected strong wind currents, GNSS signal errors, false detections by the CNN model, and illumination changes.

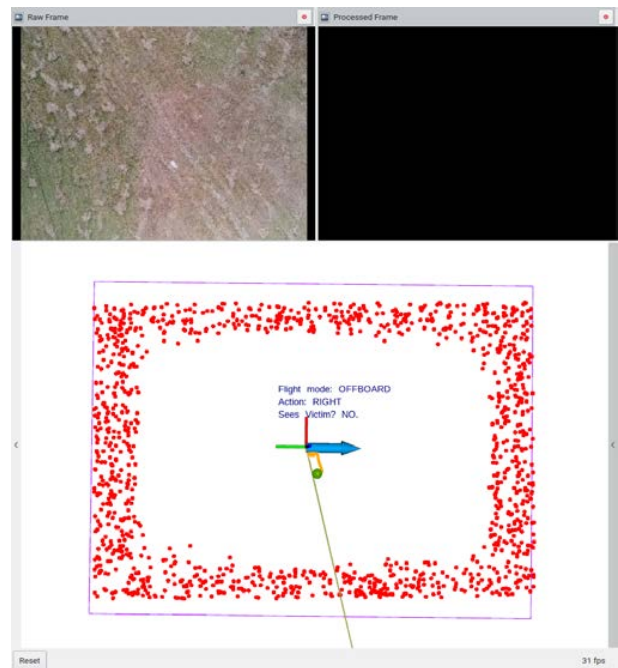


(a)

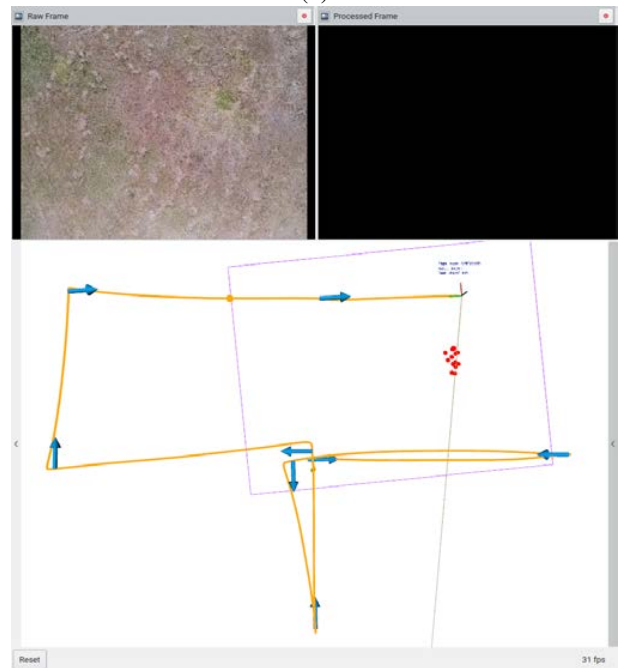


(b)

Figure 11. Traversed path by the UAV in hybrid mode to inspect an area with a potential desiccation crack. The images at the top represent the input (right) and processed frames (left). Blue arrows indicate the orientation of sequential actions taken by the aircraft. (a) Initial probability distribution of the crack localisation (red dots), with a detection confidence of 7.0%. (b) Traversed path after the UAV interacts with the environment, achieving a detection confidence of 100%



(a)



(b)

Figure 12. Traversed path by the UAV in hybrid mode to inspect an area from a false positive detection. The images at the top represent the input (right) and processed frames (left). Blue arrows indicate the orientation of sequential actions taken by the aircraft. (a) Initial probability distribution of the crack localisation (red dots), covering the camera's field of view (FOV). (b) Traversed path after the UAV interacts with the environment without any subsequent crack detections.

A case study of the UAV inspecting an area in the absence of a desiccation crack, a potential false positive, is depicted in Figure 12. When the inspection task is triggered by a false positive detection of a desiccation crack, the proposed motion planner performs a horizontal assessment of the area covered by the camera's FOV (Figure 12a). If no further detections of crack textures are triggered by the CNN model, the planner will filter out any position particles that are contained inside the camera's FOV. Once all the position belief particles are filtered, the motion planner reaches a terminal state, and the inspected area is assumed as empty of desiccation cracks (Figure 12b). If a subsequent detection is received by the CNN model while filtering particles, the motion planner will repopulate the particle belief as displayed in Figure 11 a.

Detailed accuracy statistics of the proposed UAV system between hybrid and mission flight modes are shown in Table 2. Performance metrics are composed of: True Positive (TP) detections for Texture 1 (T1) and Texture 2 (T2); number of instances per flight with Confirmed False Positives (\bar{x} CFP); and number per flight with Discarded False Positives (\bar{x} DFP).

A total of 25 flights were completed at altitudes of 10 m and 20 m. Overall, both mission and hybrid modes reported an equivalent number of flights with confirmed detections, with a slight increase of successful flights in hybrid mode to confirm Texture 2 at 10 m. Nevertheless, every flight survey in mission mode recorded an average of 177 and 72 instances of confirmed false positive detections at 10 m and 20 m, respectively. In contrast, UAV surveys in hybrid mode reported on average 0.2 and null instances of confirmed false positives per flight. An average of 1.0 instances of false positive detections were subsequently discarded by the UAV in hybrid mode at 10 m and 0.38 instances at 20 m.

The average number of instances with discarded false positive detections in hybrid mode resulted in an impact in the overall duration of the surveys. As shown in Figure 13, survey times of flights at 20 m were shorter, but tiny crack patterns are more likely to be missed compared to surveys at 10 m. Recall improvements in the semantic segmentation model are encouraged to ensure small crack blobs can also be detected for future survey operations at 20 m.

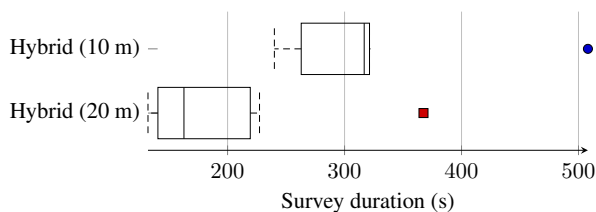


Figure 13. Comparison of elapsed time of the UAV in hybrid mode to complete the survey at 10 m and 20 m.

The UAV framework and proposed hybrid flight mode also provide improved compilation of Quadtree maps [35]—or *quadmaps*—of segmented crack textures. As quadmaps require input point cloud data for their generation and update process, Figure 14 shows the projection of the semantic segmentation model onto the ground plane. The density of the point cloud indicates the resolution at which a map is being created. Segmentations at a lower height generate finer resolution observations and are inserted in a finer resolution into the quadmap.

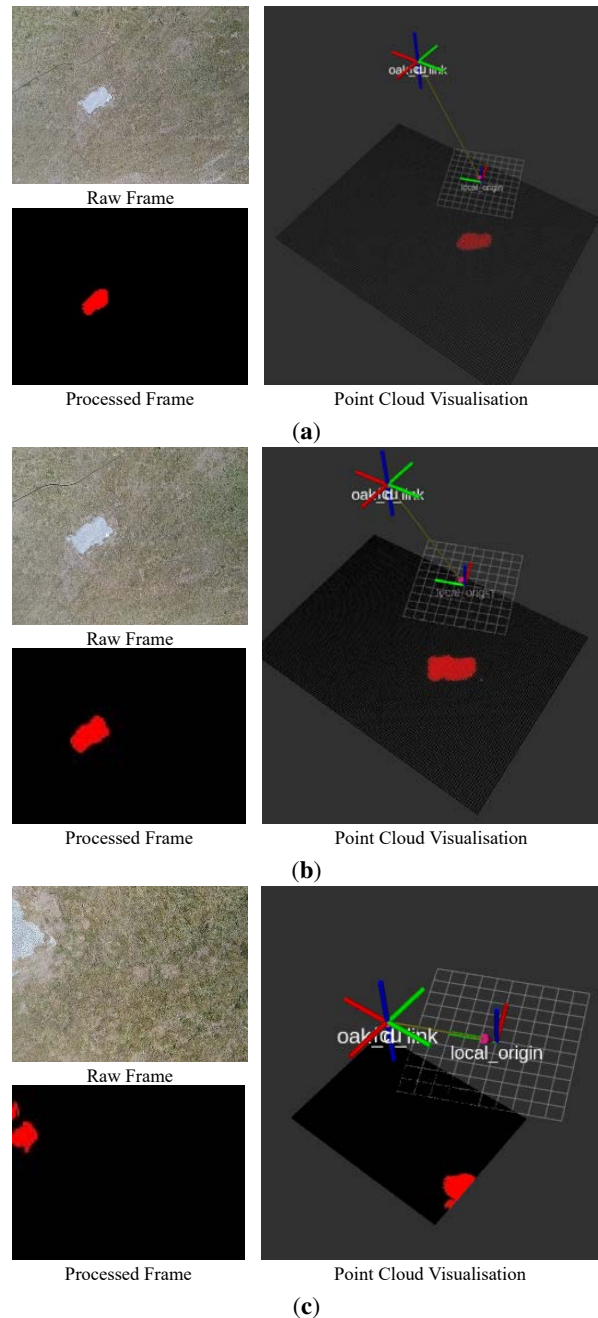


Figure 14. Level of granularity of projected processed frames into a point cloud for mapping. (a) UAV segmenting a desiccation crack contour at 25 m. (b) UAV segmenting a desiccation crack contour at 15 m. (c) UAV segmenting a desiccation crack contour at 5 m.

Higher levels of map granularity can be achieved with flights in hybrid mode. Figure 15 exemplifies two quadmaps generated by capturing outputs from the CNN model with a frequency of 8 Hz. By creating an initial quadmap from an altitude of 20 m for both flight modes, the final resolution of the quadmap in mission mode (bottom map from Figure 15) is much coarser than the one in hybrid mode, as observations for the mission were generated at a constant altitude of 20 m.

Table 2. Accuracy metrics of the proposed framework in mission and hybrid flight modes. Metrics are composed of: number of flights with True Positive (TP) detections for Texture 1 (T1) and Texture 2 (T2); number of instances per flight with Confirmed False Positives (\bar{x} CFP); and number of instances per flight with Discarded False Positives (\bar{x} DFP).

Mode	Flights	TP (T1)	TP (T2)	\bar{x} CFP	\bar{x} DFP
Mission (10 m)	7	3	3	177.0	—
Hybrid (10 m)	5	3	4	0.2	1.0
Mission (20 m)	6	6	0	72.0	—
Hybrid (20 m)	7	7	0	0.0	0.38

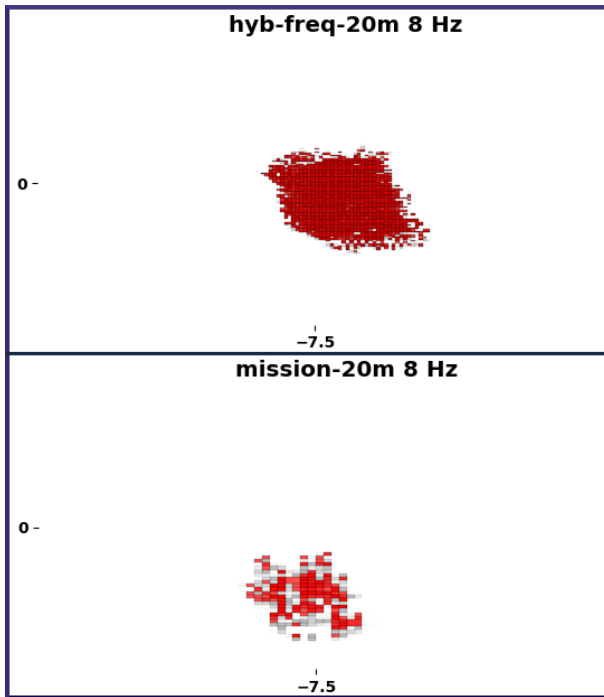


Figure 15. Level of granularity in a Quadtree map of a segmented crack texture in hybrid (top) and mission (bottom) flight modes. Red map cells indicate the presence of desiccation cracks. The intensity of red pixels is an indicator of the confidence of the map reproduction.

Furthermore, the map also shows less continuity in the area of the texture, which hints that noise corrupted the correct observation of the uniform texture area.

In continuity terms, the hybrid quadmap is sufficiently continuous, except for the boundary areas, suggesting that flights in hybrid mode could approach the texture and map it more accurately in the presence of noise. Using the UAV in hybrid mode to inspect the area for potential desiccation cracks resulted in a finer quadmap resolution. The smaller map cells show that observations were generated at a closer distance. However, the fact that the map cells are present in such a fine resolution demonstrates that they are not considered equal and cannot be pruned into summarising elements. This indicates that noise is present and the map is not equally confident about all children, maintaining the children at a fine enough resolution to distinguish differences.

Limitations in the semantics segmentation model to detect positive instances of desiccation cracks provide an upper bound for the maximum altitude for the surveys. Consequently, they have caused a negative impact on the overall speed of the system to complete the mission. The camera gimbal configuration (which delimits the extent of the camera's footprint) and image resolution of streamed frames are other factors which affect both the speed and model metrics (i.e., accuracy and recall). Improvements in the CNN-based detector metrics, processing of high-resolution images, and gimbal configuration using oblique camera angles to increase the footprint extent could reduce inspection times recorded by the UAV in hybrid mode. These improvements, however, could come with an increased demand for computational resources of the companion computer. Future research implementations of this framework should consider any computing performance impacts by onboard inference from better CNN architectures, and processing of high-resolution camera frames in resource-constrained hardware systems such as the UP² [36].

The UAV framework presented in this paper represents a novel approach towards fully automated UAV navigation for planetary exploration and mapping of desiccation cracks in the search of past life forms. Extraterrestrial environments for UAV exploration are complex and full of uncertainties, including, but not limited to, low illumination conditions, data noise caused by radiation, strong winds and dust. Challenging atmospheric conditions and absence of GNSS signal coverage, occlusion from obstacles, and imperfections from CNN model detectors can negatively affect the success of a UAV flight mission. This work proposes a probabilistic-based motion planner that addresses many of these factors by using POMDPs, which allow the modelling of uncertainty using probabilistic distributions and state beliefs. By testing a sub 2 kg UAV system for autonomous navigation and mapping of desiccation cracks in a simplified setup, future implementations should evaluate the performance of the motion planner and the UAV under more realistic testing conditions. Some areas that can be enhanced include UAV simultaneous localisation and mapping (SLAM) algorithms for autonomous navigation in GNSS-denied environments, a module for obstacle avoidance and landing detection [37], and a UAV frame design and electronic components aligned with standards for planetary exploration. In addition, autonomy and extended capabilities to survey big areas could be augmented through the use of a swarm of UAVs for cooperative detection and inspection of desiccation cracks and similar present and former aquatic ecosystems for the search of biosignatures.

6. CONCLUSIONS

This paper presented a UAV framework and motion planner for planetary exploration through the autonomous navigation and mapping of desiccation cracks in partially observable environments. The framework proposed a novel flight mode (i.e., hybrid mode) which reduced levels of object segmentation uncertainty by executing an online POMDP solver that calculates a motion policy. This policy, which can be updated mid-flight, allowed the UAV to interact with the environment and obtain improved visualisations of potential detected objects. The UAV system was evaluated with a case study of searching for biosignatures through segmentation and mapping of desiccation cracks. Real flight tests and performance metrics indicate that the proposed framework provides an improved collection of crack texture coordinates compared to a baseline motion planner. Furthermore, flight tests in hybrid mode allow the compilation of high-resolution quadmaps for improved map reproduction for post-processing analysis.

The framework discussed in this work extends the contributions of Sandino et al.[6, 24] by: (1) extending their tested UAV framework in applications beyond the scope of remote sensing and SAR operations; (2) additional validations of their framework from simulation to real flight tests; and (3) demonstrating a scaling approach to the modular framework design by adapting an OAK-D camera and a custom ResNet18 segmentation model for detection and mapping of desiccation cracks.

Avenues of future research include performance evaluation of the UAV system by processing high-resolution camera frames and more complex CNN architectures. Additional tests with multiple crack blobs and an increase in environment complexity (e.g., type and number of obstacles, slope terrain, navigation in GNSS-denied environments), as well as flight tests with UAV frame designs that are aligned with real designs for planetary exploration are encouraged. A study that compares the performance of other online POMDP solvers in resource-constrained hardware could help understand the limits of the ABT solver for UAV motion planning under uncertainty. Other planners that are not based on POMDPs could also enrich the literature on UAV path planners for planetary exploration under environment uncertainty and partial observability.

APPENDIX

The flight plan parameters for conducted tests in *mission* and *hybrid* flight modes are shown in Table 3 respectively.

The set of hyper-parameters used in the TAPIR toolkit for flights in hybrid mode are shown in Table 4.

ACKNOWLEDGMENTS

The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. Special thanks to the Samford Ecological Research Facility (SERF) team (Marcus Yates and Lorrelle Allen) for their continuous assistance and equipment provided during the flight tests. The authors would also like to gratefully thank the QUT Research Engineering Facility (REF) team (Dr Dmitry Bratanov, Gavin Broadbent, Dean Gilligan), and Marcus Yates for their technical support that made possible conducting the flight tests.

Table 3. Flight plan parameters from the OAK-D sensor properties.

Property	Value
UAV altitude	10 m and 20 m
UAV velocity	2 m/s
Lens width	5.94 mm
Lens height	4.48 mm
Field of View (v, h)	68.8°, 81°
Camera focal length	4.5 mm
Image resolution	256 px by 192 px
Overlap	50%
Bottom right waypoint	-27.3894400°, 152.8729285°
Top right waypoint	-27.3889921°, 152.8729304°
Top left waypoint	-27.3889909°, 152.8724435°
Bottom left waypoint	-27.3894399°, 152.8724408°

Table 4. Set of hyper-parameters used in the TAPIR toolkit for flights in hybrid mode.

Variable	Description	Value
z_{\max}	Maximum UAV altitude	21.5 m
z_{\min}	Minimum UAV altitude	1.25 m
φ_u	UAV Heading	0°
δ_z	UAV climb step	2 m
λ	Frame overlap	50%
α	Camera pitch angle	0°
ζ_{\min}	Minimum detection confidence	10%
ζ	Confidence threshold	85%
γ	Discount factor	0.95
Δt	Time step interval	4 s
t_{\max}	Maximum flying time	10 min

This research was funded by The Commonwealth Scientific and Industrial Research Organisation (CSIRO) through the CSIRO Data61 PhD and Top Up Scholarships (Agreement 50061686); The Australian Research Council (ARC) through the ARC Discovery Project 2018 “Navigating under the forest canopy and in the urban jungle” (grant number ARC DP180102250); and The Queensland University of Technology (QUT) through the Higher Degree Research (HDR) Tuition Fee Sponsorship.

REFERENCES

- [1] L. E. Hays, H. V. Graham, D. J. Des Marais, E. M. Hausrath, B. Horgan, T. M. McCollom, M. N. Parenteau, S. L. Potter-McIntyre, A. J. Williams, and K. L. Lynch, “Biosignature preservation and detection in mars analog environments,” pp. 363–400, Apr. 2017.
- [2] L. Gonzalez, G. Montes, E. Puig, S. Johnson, K. Mengersen, and K. Gaston, “Unmanned aerial vehicles (UAVs) and artificial intelligence revolutionizing wildlife monitoring and conservation,” *Sensors*, vol. 16, no. 1, p. 97, Jan. 2016.
- [3] J. Sandino, F. Gonzalez, K. Mengersen, and K. J.

- Gaston, "UAVs and machine learning revolutionising invasive grass and vegetation surveys in remote arid lands," *Sensors*, vol. 18, no. 2, p. 605, Feb. 2018.
- [4] F. Vanegas, J. Roberts, and F. Gonzalez, "UAV tracking of mobile target in occluded, cluttered and GPS-denied environments," in *Aerospace Conference*. IEEE, Mar. 2018, pp. 1–7.
- [5] J. Sandino, G. Pegg, F. Gonzalez, and G. Smith, "Aerial mapping of forests affected by pathogens using UAVs, hyperspectral sensors, and artificial intelligence," *Sensors*, vol. 18, no. 4, p. 944, Mar. 2018.
- [6] J. Sandino, F. Vanegas, F. Gonzalez, and F. Maire, "Autonomous UAV navigation for active perception of targets in uncertain and cluttered environments," in *Aerospace Conference*. Big Sky, MT, USA: IEEE, Mar. 2020, pp. 1–12.
- [7] B. Balaram, T. Canham, C. Duncan, H. F. Grip, W. Johnson, J. Maki, A. Quon, R. Stern, and D. Zhu, "Mars helicopter technology demonstrator," in *Atmospheric Flight Mechanics Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, Jan. 2018, pp. 1–18.
- [8] S. Potter, "Nasa's ingenuity mars helicopter to begin new demonstration phase," 2021. [Online]. Available: <https://www.nasa.gov/press-release/nasa-s-ingenuity-mars-helicopter-to-begin-new-demonstration-phase>
- [9] W. Johnson, S. Withrow-Maser, L. Young, C. Malpica, W. J. F. Koning, W. Kuang, M. Fehler, A. Tuano, A. Chan, A. Datta, C. Chi, R. Lumba, D. Escobar, J. Balaram, T. Tzanetos, and H. Fjaer Grip, "Mars science helicopter conceptual design," NASA, CA, US, Tech. Rep., 2020.
- [10] Johns Hopkins APL, "What is dragonfly?" 2021. [Online]. Available: <https://dragonfly.jhuapl.edu/What-Is-Dragonfly/>
- [11] R. D. Lorenz, E. P. Turtle, J. W. Barnes, M. G. Trainer, D. S. Adams, K. E. Hibbard, C. Z. Sheldon, K. Zacny, P. N. Peplowski, D. J. Lawrence, M. A. Ravine, T. G. Mcgee, K. S. Sothen, S. M. Mackenzie, J. W. Langelaan, S. Schmitz, L. S. Wolfarth, and P. D. Bedini, "Dragonfly: A rotorcraft lander concept for scientific exploration at titan," *Johns Hopkins APL Technical Digest*, vol. 34, no. 3, pp. 374–387, 2018.
- [12] S. Macdonald and A. Stevens, "How to explore planets with drones," *Astronomy & Geophysics*, vol. 59, no. 3, pp. 3.18–3.22, Jun. 2018.
- [13] M. Hassanalian, D. Rice, and A. Abdelkefi, "Evolution of space drones for planetary exploration: A review," *Progress in Aerospace Sciences*, vol. 97, pp. 61–105, Feb. 2018.
- [14] U. Ilhan, L. Gardashova, and K. Kilic, "UAV using dec-POMDP model for increasing the level of security in the company," *Procedia Computer Science*, vol. 102, pp. 458–464, 2016.
- [15] M. Chen, E. Frazzoli, D. Hsu, and W. S. Lee, "POMDP-lite for robust robot planning under uncertainty," *International Conference on Robotics and Automation*, pp. 5427–5433, Feb. 2016.
- [16] C. Ponzoni Carvalho Chanel, A. Albore, J. T'Hooft, C. Lesire, and F. Teichteil-Königsbuch, "Ample: an anytime planning and execution framework for dynamic and uncertain problems in robotics," *Autonomous Robots*, vol. 43, no. 1, pp. 37–62, Jan. 2019.
- [17] F. Vanegas and F. Gonzalez, "Uncertainty based online planning for UAV target finding in cluttered and GPS-denied environments," in *Aerospace Conference*, vol. 2016-June. IEEE, Mar. 2016, pp. 1–9.
- [18] F. Vanegas, D. Campbell, M. Eich, and F. Gonzalez, "UAV based target finding and tracking in GPS-denied and cluttered environments," in *International Conference on Intelligent Robots and Systems*. Daejeon, South Korea: IEEE/RSJ, Oct. 2016, pp. 2307–2313.
- [19] J. Sandino, F. Vanegas, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez, "UAV framework for autonomous onboard navigation and people/object detection in cluttered indoor environments," *Remote Sensing*, vol. 12, no. 20, p. 3386, Oct. 2020.
- [20] S. Ragi and E. K. P. Chong, "UAV path planning in a dynamic environment via partially observable markov decision process," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2397–2412, Oct. 2013.
- [21] ———, "UAV guidance algorithms via partially observable markov decision processes," in *Handbook of Unmanned Aerial Vehicles*, K. Valavanis and G. Vachtsevanos, Eds. Dordrecht: Springer Netherlands, 2015, ch. 73, pp. 1775–1810.
- [22] D. S. Bayard, D. T. Conway, R. Brockers, J. H. Delaune, L. H. Matthies, H. F. Grip, G. B. Merewether, T. L. Brown, and A. M. San Martin, "Vision-based navigation for the nasa mars helicopter," in *Scitech Forum*. Reston, Virginia: American Institute of Aeronautics and Astronautics, Jan. 2019.
- [23] T. Sasaki, K. Otsu, R. Thakker, S. Haesaert, and A. A. Agha-Mohammadi, "Where to map? iterative rover-copter path planning for mars exploration," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2123–2130, Apr. 2020.
- [24] J. Sandino, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez, "Drone-based autonomous motion planning system for outdoor environments under object detection uncertainty," *Remote Sensing*, vol. 13, no. 21, p. 4481, Nov. 2021.
- [25] J. Sandino, P. Caccetta, C. Sanderson, F. Maire, and F. Gonzalez, "Reducing object detection uncertainty from rgb and thermal data for UAV outdoor surveillance," in *Aerospace Conference*. Big Sky, MT, USA: IEEE, Mar. 2022, p. (accepted).
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem. IEEE Computer Society, Dec. 2015, pp. 770–778.
- [27] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2017.
- [28] J. Galvez-Serna, N. Mandel, J. Sandino, F. Vanegas, N. Ly, D. Flannery, and F. Gonzalez, "Real-time segmentation of desiccation cracks onboard UAVs for planetary exploration," in *Aerospace Conference*. Big Sky, MT, USA: IEEE, Mar. 2022, p. (accepted).
- [29] Luxonis, "DepthAI: Embedded machine learning and computer vision API," Littleton, CO, US, 2020.
- [30] A. Koubaa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, "Micro air vehicle link (mavlink) in a nutshell: A survey," *IEEE Access*, vol. 7, pp. 87 658–87 680, Jun. 2019.

- [31] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.
- [32] A. Chovancová, T. Fico, L. Chovanec, and P. Hubinsk, “Mathematical modelling and parameter identification of quadrotor (a survey),” *Procedia Engineering*, vol. 96, pp. 172–181, 2014.
- [33] H. Kurniawati and V. Yadav, “An online POMDP solver for uncertainty planning in dynamic environment,” in *Robotics Research. Springer Tracts in Advanced Robotics*, M. Inaba and P. Corke, Eds. Springer, Cham, 2016, vol. 114, pp. 611–629.
- [34] D. Klimenko, J. Song, and H. Kurniawati, “Tapir: A software toolkit for approximating and adapting POMDP solutions online,” in *Australasian Conference on Robotics and Automation*. Melbourne, Australia: ARAA, Dec. 2014, pp. 1–9.
- [35] N. Mandel, J. Sandino, J. Galvez-Serna, F. Vanegas, and F. Gonzalez, “Resolution-adaptive quadrees for semantic segmentation mapping in UAV applications,” in *Aerospace Conference*. Big Sky, MT, USA: IEEE, Mar. 2022, p. (accepted).
- [36] N. Mandel, M. Milford, and F. Gonzalez, “A method for evaluating and selecting suitable hardware for deployment of embedded system on UAVs,” *Sensors*, vol. 20, no. 16, p. 4420, Aug. 2020.
- [37] R. Brockers, J. Delaune, P. Proença, P. Schoppmann, M. Domnik, G. Kubiak, and T. Tzanetos, “Autonomous safe landing site detection for a future mars science helicopter,” in *IEEE Aerospace Conference*, 2021, pp. 1–8.

BIOGRAPHY



Juan Sandino is a research engineer at the Queensland University of Technology (QUT) Centre for Robotics and CSIRO Data61, Brisbane, Australia. He holds a B.Eng. (Mechatronics) and is undertaking a PhD in robotics and autonomous systems at QUT, Brisbane, Australia. His primary interests comprise autonomous decision-making in small Unmanned Aerial Vehicles (UAVs), machine learning and computer vision applied on UAVs for remote sensing, with a focus on hyperspectral and high-resolution image processing. Juan has worked for research projects in biosecurity, environment monitoring and time-critical applications such as land search and rescue to find lost people in collapsed buildings and bushlands.



Julian Galvez-Serna received his B.S. degree in Mechatronics Engineering in 2016 and a M.Sc. in Engineering from the University of Antioquia in 2019. He is currently a Ph.D. candidate in robotics and autonomous systems at Queensland University of Technology. His current research activities include UAV mission planning for planetary exploration in uncertain environments.



Nicolas Mandel received the BSc in Sports and Technology from the Otto-von-Guericke University Magdeburg and the MSc in Biomedical Engineering Sciences from the University of Applied Sciences Technikum Vienna after writing his thesis about mechanical modelling of cardiovascular treatments at the University of Cape Town. He is currently pursuing his PhD with the QUT Centre for Robotics at the Queensland University of Technology, Brisbane, focussing on employing semantics for UAV navigation.



Fernando Vanegas is a Research Fellow in Robotics at Queensland University of Technology. He holds a M.Sc. in Electrical Engineering from Halmstad University and a PhD in Aerial Robotics from QUT. He has developed frameworks for UAV mission planning in GPS-denied environments for navigation, target finding and tracking purposes using machine learning techniques based on reinforcement learning and optimal sequential decision making. His current research interests include motion planning for UAV exploration in GPS-denied and cluttered environments, POMDP, SLAM and Visual Odometry.



Felipe Gonzalez is a Professor at the School of Electrical Engineering and Robotics (EER), Engineering Faculty with a passion for innovation in the fields of aerial robotics and automation. Gonzalez interest is in creating aerial robots, drones or UAVs that possess a high level of cognition using efficient on-board computer algorithms using advanced optimisation and game theory approaches that assist us to understand and improve our physical and natural world. Professor Gonzalez leads the Airborne Sensing Lab at QUT. He is the co-author of several books in UAV based remote sensing and UAV based design based on evolutionary optimisation and game strategies and as of 2021 has published nearly 155 refereed papers.

Chapter 8

Conclusions

THIS research developed a fully autonomous decision-making system using sequential decision processes (SDPs) onboard a sub-2 kg quadrotor UAV with increased cognition to autonomously explore and find objects, without human intervention in partially observable indoor and outdoor environments. This research created a novel scalable UAV framework for onboard inference of convolutional neural network (CNN) models and developed a partially observable Markov decision process (POMDP)-based motion planner for small UAVs to operate in real-world environments under object detection uncertainty. The research contributions were validated throughout the publication of five academic peer-reviewed papers, presented in Chapters 3 to 7 as body chapters of this dissertation. The six key contributions are:

1. Paper 1: the formulation of a UAV decision-making problem using a POMDP for real-time navigation in global navigation satellite system (GNSS)-denied cluttered indoor environments under various types of object detection uncertainty, tested in software in the loop (SIL) and with onboard inference of a CNN object detector using hardware in the loop (HIL).
2. Paper 2: the development of a decision-making framework for autonomous decision-making onboard sub-2 kg UAVs for navigation in GNSS-denied environments under object detection uncertainty, validated in both simulated and real flight tests.
3. Paper 3: an extension of the problem formulation (Paper 1) for autonomous UAV navigation in outdoor environments under object detection uncertainty, using a novel reward function definition that improved UAV's traversed path. The established UAV system adapts its motion policy with and without pre-defined survey plans due to the creation of a novel flight mode (*i.e.*, hybrid mode).
4. Paper 3: the design of a UAV system architecture that bridges the gap between simulated and real flight tests to obtain high-fidelity statistics of UAV behaviour in simulation, and speed the development and testing process of problems formulated using model-based POMDP solvers and implemented UAV frameworks.
5. Paper 4: the creation of a modular and scalable UAV framework for autonomous exploration and object finding in outdoor environments, tested successfully in SAR contexts, with a range of red, green, blue (RGB) and thermal camera payloads, and CNN models, validated with real flight tests.
6. Paper 5: the development of a decision-making framework for autonomous detection and mapping of desiccation cracks using small UAVs, demonstrating the scalability capabilities of

the UAV framework (Paper 4) in a new survey domain (planetary exploration), camera payload, and CNN-based semantic segmentation model.

This research established two key components as the most impactful research contributions: 1) the design of the reward function and tuned hyperparameter values; and 2) the problem formulation using a POMDP, as they critically influenced the behaviour of the UAV to accomplish its mission during the experimental phase. The design of the reward function extended the traditional definition of the sum of conditioned reward variables and fixed values into an algorithm that prioritised the evaluation of system states (and returned reward values) into three key stages: 1) assessment of next states that compromised the integrity of the UAV, environment, or objects of interest; 2) assessment of next states that led the UAV to achieve the mission (*e.g.*, potential victim/dessication crack found, or victim/dessication crack confirmed); and 3) evaluation of other “non-terminal” states by returning constant cost values or established equations, such as *UAV to object* vertical and horizontal distance cost, and explored footprint cost. Reward function definitions and tuned hyperparameters per case study presented in papers 1 to 5, Chapters 3 to 7, converged after an initial iterative process with the UAV system in HIL simulations by following a grid search technique. This research demonstrated that enabling SIL and HIL support resulted in a robust architecture design, as no major adjustments to hyperparameters were required while transitioning between HIL simulations to real flight tests.

8.1 Research Findings

The questions presented in Section 1.2.1 determined the scope of this research to design a UAV system that can navigate autonomously and detect uncertain objects in uncertain environments. This section highlights the research findings arising from each research question.

Research Question 1.1

What level of artificial cognitive learning and uncertainty modelling is required to identify, localise or quantify objects positioned in cluttered and challenging scenarios using small UAVs and vision-based object detectors?

This research identified and confirmed that using POMDPs enable autonomous decision-making under uncertainty and partial observability in multi-objective UAV navigation problems. A model-based online POMDP solver was employed to compute a path planning strategy, and subsequently update it onboard the aircraft after collecting new observations in challenging indoor and outdoor environments. These model-based POMDP solvers, such as adaptive belief tree (ABT) enable flexible modelling of uncertainty via probabilistic distributions over a set of belief states.

This effective approach to model uncertainty using a POMDP framework was established (Chapters 3 and 4) by: 1) predicting local position changes of the UAV per action taken through a system identification process (SIP) in MATLAB, which converges to an approximate transfer function of the UAV motion dynamics (including the selection of most suitable function order), and applied discrete control theory to obtain the motion response equation of the UAV (Section 3.4.6); 2) estimating the detection, or presence of a potential object conditioned by the camera pose at the UAV frame and its projected footprint of the environment (Section 3.4.9); and, 3) modelling the detection confidence from potential detections using a linear function that returned high confidence values the closer the UAV is from the object and vice versa (Section 4.4.4). These models were implemented in the generative model of the ABT solver, which populates probabilistic data to the transition T and observation

Z functions from a modelled observation $o \in O$ given an action $a \in A$ and the next state $s' \in S$.

The use of POMDPs for autonomous UAV navigation under uncertainty enabled the modelling of initial belief states using probability distributions, testing three types of distribution setups in accordance to the situational awareness of the survey mission, prior to any flight operations. Normal distributions were applied in flight setups where the object (*i.e.*, victims/dessication crack) believed to be placed at one or multiple locations, and uniform distributions covering the flight area if the location of the potential object was unknown. A third extension to initial particle belief distribution was used in outdoor flight operations in hybrid mode, with uniform distributions covering the camera's field of view (FOV), and enable the UAV to inspect areas with the potential presence of objects. The UAV was therefore capable of executing sequential action commands for an average of one second and four seconds in indoor and outdoor environments, respectively, by implementing the formulated navigation problem using the ABT solver. Higher accuracy of recorded object locations was identified in experimental tests in outdoor environments by incorporating the developed POMDP-based motion planner (and offboard and hybrid modes), with the expense of slightly reduced speeds to survey areas compared to baseline motion planners (*e.g.*, traditional missions in lawnmower patterns from a list of position waypoints).

Research Question 1.2

Which factors define the complexity of an SDP so as to reduce object detection uncertainty from collected environment observations using vision-based sensors in time-critical applications?

This research into the modelling of UAV navigation problems with POMDPs demonstrated the greatest impact on the behaviour of the aircraft to reduce object detection uncertainty relies on the definition of the reward function $R(s, a)$. For instance, (Chapter 3) a simplified definition of R (Section 3.4.7) as the sum of individual reward variables with assigned values only after matching a system state s after taking an action a . Increasing the complexity of R (Section 4.4.4) by adding two-dimensional (2D) map representations of paths previously traversed by the UAV improved the UAV trajectories when navigating under an initial object location belief defined using normal distributions in single or multiple locations (Chapter 4). Since some flight tests reported suboptimal trajectories if the initial belief state of the location of the object was defined using uniform probability distribution across the flying area (Figure 4.13), R was extended using a novel pseudocode containing a set of conditionals for each reward variable and new concepts such as the UAV altitude, horizontal distance, and footprint overlap costs (Section 5.3.5). Recorded trajectories (Chapter 5 onwards) revealed a significant improvement in UAV behaviour for exploration and inspection of areas with potential presence of objects in outdoor environments, with and without a predefined flight plan.

Other factors that follow the definition of R , in terms of agent behaviour (but to a lesser degree), are hyperparameter tuning, and the definition of the action space A . Hyperparameter tuning in this research was restricted to set the optimal values of reward variables from R , and hyperparameters of ABT implemented in toolkit for approximating and adapting POMDP solutions in real time (TAPIR), which were found after conducting an iterative set of HIL simulations using a grid search technique. An extension in the definition of low-level action commands for experiments conducted in outdoor environments (Section 5.3.2) had a positive impact on UAV traversed paths considering that ABT only enables the definition of discrete actions. This extension enabled execution of adjustable horizontal action commands without augmenting the set of actions in the POMDP formulation by declaring the desired overlap, which depends on the altitude of the UAV and the camera's FOV.

The software capabilities of flight modes supported by the UAV autopilot simplified the com-

plexity requirements of the motion planner. A novel flight mode (*i.e.*, *hybrid mode*) combined the capabilities of *mission* mode and *offboard* mode to inspect areas with potential objects using the POMDP-based motion planner. This extension reused the software implementation of the motion planner in offboard mode by redefining of the search extents of the survey to be conditioned by the camera's FOV, and a software update, to iteratively restart the Decision-making Module after positive detections were triggered by the CNN model, until the UAV reached the last position waypoint. Developing this hybrid mode saved research and development resources as it enabled multiple objects to be detected without further modifications to the formulated problem and POMDP-based motion planner.

Research Question 1.3

What are the modelling considerations in a formulated SDP that enable the scalability of a UAV framework for autonomous navigation for a range of diverse vision-based payloads, and remote sensing application needs beyond object detection?

This research identified (Chapters 6 and 7) three key modelling parameters that allow scaling the UAV framework are the: 1) mathematical modelling of a camera's FOV and footprint extents using the sensor lens properties; 2) motion model of a sub 2-kg quadrotor UAV using a SIP; and, 3) modular design of the system architecture which allocates high-level detection data from CNN-based object or segmentation models into an Observation Server.

UAV framework implementations (Chapters 6 and 7) demonstrated the benefits of modelling a generic FOV and footprint estimations based on the specifications of the sensor lens. A switch between vision-based payloads, such as the HBV 1615, Arducam b019710, GoPro Hero9 and OpenCV AI kit (OAK)-D RGB cameras, and the FLIR Tau 2 thermal camera, only required updating the sensor lens specifications, most of which are available from the sensor datasheet, to the POMDP formulated problem, in order to integrate the camera with the UAV system.

The motion modelling of the Holybro X500 quadrotor UAV using a SIP enabled high level of abstraction of action commands regardless of their desired magnitude in traversed horizontal distance (*e.g.*, 1, 2, 5, 10 m). However, the framework implementation for indoor and outdoor environments was limited to a single UAV frame. Other multirotor UAV frames are expected to work provided a new SIP is conducted beforehand with the help of a high-quality Motion Tracking System such as VICON. The use of a GNSS-RTK device is encouraged to collect step responses of the UAV frame by changes in position and velocity commands in outdoor environments with the absence of motion tracking systems.

The system architecture design (Figures 6.2 and 7.2) implemented an Observation Server which translates explicit detection outputs by onboard inference of CNN models into generic observations compatible with the defined observations in the problem formulation using POMDPs. This scalability capability was validated in the framework implementation that detected and mapped desiccation cracks (Chapter 7), after successfully converting segmented blobs into bounding boxes with centroids that represent objects (Section 7.2.2).

Research Question 2.1

How can computationally intensive tasks such as online SDP algorithms, and onboard inference of CNN models for object detection and segmentation, be integrated to run simultaneously under resource-constrained hardware in sub-2 kg UAVs?

This research developed modular frameworks that formulate, implement and execute UAV systems with enhanced cognition for autonomous navigation in time-critical applications, and in indoor and outdoor environments. The design contributions of UAV system architecture (Chapters 3, 4, 6 and 7) demonstrated a successful selection of hardware and software resources to optimise the framework implementation for sub 2-kg UAVs. Developing realistic simulation environments (Chapters 4 and 5) bridged the gap between simulated experiments and real flight tests as the behaviour of the UAV remained consistent after transitioning from simulated to physical experiments without returning hyperparameters, and reducing therefore research and development efforts on determining the optimal combination of hardware and software tools.

This simulation environment was developed by: 1) emulating the specifications and capabilities of the UAV autopilot through SIL; 2) enabling HIL to test developed SDPs (*i.e.*, model-based POMDP solvers using ABT) running onboard the physical companion computer attached to the sub-2 kg UAV frame; 3) performing onboard inference of CNN models (through HIL and the companion computer) for object detection and segmentation after streaming camera frames from an emulated vision-based sensor; and, 4) integrating real-world datasets and visual terrain textures collected from airborne UAV flights, such as high-resolution RGB orthomosaics and light detection and ranging (LiDAR) data.

The simulation environment developed in the Gazebo simulator and robot operating system (ROS) lead to the design of modular framework components, as well as emulating the UAV autopilot capabilities through the PX4 SIL plugging. An Intel UP², the companion computer used to compute the ABT solver (through the TAPIR toolkit) in the simulator and real flight tests, contained a Myriad X vision processing unit (VPU) connected to a PCIe slot, utilised for onboard inference of CNN models.

Research Question 2.2

What are the design criteria to scale a UAV framework for autonomous navigation under environment and object detection uncertainty to other remote sensing applications that require autonomous decision-making capabilities onboard small UAVs?

This research established a UAV system architecture design (Chapters 5 to 7) to execute key modular components such as the Computer Vision Module, Mapping Module, Decision-making Module, and Motion Module. The UAV framework implemented in this research followed the above-mentioned architecture design, and enabled a sub-2 kg UAV to: 1) search and locate a child-shaped mannequin in an emulated office building; 2) explore and inspect areas to detect potential victims using adult-shaped mannequins in a bushland; and, 3) detect and map desiccation cracks from dry lake beds.

These scalability capabilities were validated after integrating multiple vision-based payloads in the Holybro X500 quadrotor frame, a set of custom CNN models for onboard inference per case study and image source (*i.e.*, RGB or thermal), and high-level hyperparameter adjustments for each flight operation. The modelling of the camera lens properties in the POMDP problem formulation enabled the use of payloads, ranging from RGB cameras (*i.e.*, HBV 1615, Arducam b019710, GoPro Hero 9), thermal cameras with real time detection of adults (*i.e.*, FLIR Tau 2, shown in Chapter 6 and video preview¹) and an OAK-D. The design of the Computer Vision Module enabled high levels of abstraction and use various CNN model architectures such as MobileNet SSD to detect victims using RGB frames, TinyYoloV2 to detect victims using Thermal imagery, and a ResNet18 for semantic segmentation of desiccation cracks using RGB data. This UAV framework is robust and can be expanded to other uses, such as the development of a UAV system for real-time segmentation of desiccation cracks [100], and

¹ <https://youtu.be/yIPNBwNYtAo?t=148>

semantic quad- and oc-trees for UAV map representation and navigation [101].

8.2 Recommendations

The following seven technical recommendations should be considered to reproduce the UAV framework and problem formulation for autonomous navigation onboard small UAVs using POMDPs, computer vision and vision-based sensors:

- **Middleware:** despite the use of ROS satisfied the needs of implemented UAV frameworks, an upgrade to ROS2 as the communication middleware would provide an increasing number of libraries, optimisations, and technical support beyond May 2025.
- **Payload (cameras):** vibrations in the frame of many small multirotor UAVs are common and could negatively affect the quality of streamed camera frames while the UAV is flying. Every source of vibration needs to be evaluated with strategies to mitigate it, such as the use of anti-vibration and gimbal mounts.
- **Thermal cameras:** the scalability demonstration of the autonomous UAV system for object finding using thermal imagery (Chapter 6) established in preliminary real flight tests that the optimal timeframe to operate with thermal imagery is night conditions when the background temperature (*i.e.*, soil or grass, trees and buildings) is considerably different to the body temperature of people. The strong likelihood for background temperatures to reach or surpass human body temperatures (36°C approx.) during daylight conditions could result in visualising merged human silhouettes with the background.
- **Reward function:** the complexity of the reward function formulated using a POMDP is key in influencing the behaviour of the UAV using model-based POMDP solvers (*i.e.*, ABT solver). A significant improvement in the traversed path by the UAV was achieved from indoor to outdoor flight tests (Chapters 4 and 5) after defining the penalties for footprint overlap from past explored regions, and suboptimal UAV altitude while exploring the environment.
- **Experiment setup:** high-fidelity results between SIL/HIL simulations and real flight tests suggest future experiments involving POMDPs could be conducted in simulation first, and then followed with real flight tests. Enabling HIL by running the Computer Vision, Decision-making and Motion Modules on the companion computer through the use of ROS and the PX4 SIL plugin saved economic and time resources while converging in the required models for the transition and observation functions, the tuning of hyperparameters of the TAPIR toolkit, and evaluation of fail-safes against unexpected behaviour of the motion planner while operating the UAV in *offboard* and *hybrid* modes.
- **Flight tests:** a comprehensive evaluation of software and hardware fail-safe modes is necessary to regain control from unexpected behaviour of the POMDP-based motion planner or UAV autopilot. Software fail-safes include overriding action commands sent by the motion planner if such commands lead the UAV to collide with obstacles or overpass a pre-defined geofence. Defining physical switches from the UAV remote controller could assist in regaining control of the aircraft regardless of the current flight mode controlled by the motion server, especially if operating the UAV in *hybrid* mode. An isolated test with the UAV pointing to the objects at a range of heights and orientations will identify the CNN model limits and the maximum altitude for UAV surveys.

- **Onboard inference from CNN models:** evaluating any performance and speed loss by onboard inference in resource-constrained hardware is recommended as other CNN architectures could provide better object detection performance than using off-the-shelf detectors such as MobileNet SSD. Streaming high-resolution camera frames and message transporting using ROS could increase the demand of computational resources in embedded systems such as the UP² and Intel Myriad VPU.

8.3 Future Research Avenues

The contributions from this research provided the foundation to develop small UAVs with fully automated onboard navigation under environment and object detection uncertainty. This research demonstrates the versatility of POMDP-based motion planners and established frameworks for small UAVs in indoor and outdoor environments, research that can be expanded by:

- **Updating 3D occupancy maps mid-flight:** the UAV framework for autonomous navigation in indoor and outdoor environments reads a 3D occupancy map that represents the location and dimensions of any present objects (*i.e.*, obstacles) in the surveyed area. Since the occupancy map was compiled prior to any flight tests in simulation and with real hardware, future research could integrate sensors capable of displaying depth maps through 3D point clouds. These point clouds can be generated using LiDAR or depth cameras (such as Intel Realsense D435), which can be processed in Octomap to refresh the 3D occupancy map mid-flight.
- **Implementing complex camera configurations:** the problem formulation created here covered downward-looking camera configurations. Extending the estimation of camera footprint, FOV, and object position to fit camera gimbals onboard small UAVs using oblique angles could provide additional real-time reconnaissance in emergency situations.
- **Adapting the UAV framework to commercial UAV solutions:** while the UAV framework presented in this thesis is a competitive approach for academic and research purposes, higher adoption probabilities of this implementation could occur for real-world UAV applications provided the framework follows a standalone design compatible with commercial UAV manufacturers such as DJI, Parrot and Yuneec. This adaptation effort should address the flight endurance shortcomings of the current UAV frame for extended flight operations for up to 30 minutes.
- **Extending the problem formulation for multiple and dynamic objects:** the problem formulation to detect and locate a single and static object was based on a per camera frame and could be extended: for multiple objects appearing simultaneously in the same camera frame; and tracking of dynamic objects [64] in environments with and without GNSS signal coverage for remote sensing applications such as border protection, surveillance, and pest control.
- **Increasing the action and state space:** future extensions to the POMDP formulated problem established from this research include adding new actions such as dynamic camera gimbal angle, velocity motion commands, and high-level actions included by other UAV autopilots (*e.g.*, follow me, take-off, landing). Additional states that could expand this research, in the context of SAR and victim detection uncertainty, include whether the victim is alive or injured, their age and gender. The evaluation of online POMDP solvers compatible with continuous (rather than discrete) actions could improve traversed paths by the UAV if low-level motion commands such as position of velocity setpoints are implemented.

- **Extending problem formulation for a swarm of UAVs:** a swarm of UAVs uniformly distributed in a survey area will aid applications that demand surveying extensive areas of land as fast as possible as occurs in SAR missions, specifically *Reflex Tasking*, a standard first response search that defines a 300 m radius of search extent from the last known position, place last seen and initial planning position [46].
- **Improving UAV interactivity with detected objects:** specific actuators attached to small UAVs could improve their use in real-world environments, including the deployment of first aid kits to found victims and attached probes for ground sampling of rock sediments.
- **Evaluating system performance with other CNN architectures:** incorporating CNN architectures other than the off-the-shelf MobileNet SSD [102], such as Cascade RCNN, RefineDet and CornerNet [94], [103] is recommended for future research. These state-of-the-art architectures could make surveillance tasks more effective as the UAV should be able to cover larger areas by detecting smaller victims/objects at equivalent altitudes and with a higher detection rate. In addition, evaluating the performance impacts in resource-constrained hardware for onboard inference using these CNN models from high-resolution streamed frames could lead to a better understanding of the limits of the decision-making framework in sub-2 kg UAVs.
- **Evaluating the use of other online POMDP solvers:** this research could be expanded with a performance comparison between ABT and other model-based online POMDP solvers such as POMCPOW, LABECOP, and Dec-POMDP [104], or other software toolkits such as JuliaPOMDP [105] which encapsulates a range of modular MDP and POMDP solvers that can be tested within the same toolkit.
- **Investigating existing standards to regulate the use of small UAVs for SAR:** Australia regulations for land SAR operations [46] demand a methodology to successfully integrate the use of small autonomous UAVs such as that presented successfully in this thesis into existing protocols, equipment, and procedures.

References

- [1] B. J. Stark and Y. Q. Chen, “Remote sensing methodology for unmanned aerial systems,” in *Unmanned Aircraft Systems*, E. M. Atkins, A. Ollero, and A. Tsourdos, Eds., New York, USA: Wiley, 2016, ch. 2, pp. 17–27 (cit. on pp. [1](#), [12](#), [13](#)).
- [2] G. Pajares, “Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs),” *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 4, pp. 281–330, Apr. 2015. DOI: [10.14358/PERS.81.4.281](#) (cit. on pp. [1](#), [11](#)).
- [3] L. Gonzalez, G. Montes, E. Puig, S. Johnson, K. Mengersen, and K. Gaston, “Unmanned aerial vehicles (UAVs) and artificial intelligence revolutionizing wildlife monitoring and conservation,” *Sensors*, vol. 16, no. 1, p. 97, Jan. 2016. DOI: [10.3390/s16010097](#) (cit. on pp. [1](#), [9](#), [11](#), [23](#)).
- [4] J. Sandino, F. Gonzalez, K. Mengersen, and K. J. Gaston, “UAVs and machine learning revolutionising invasive grass and vegetation surveys in remote arid lands,” *Sensors*, vol. 18, no. 2, p. 605, Feb. 2018. DOI: [10.3390/s18020605](#) (cit. on pp. [1](#), [9](#), [11](#)).
- [5] J. Sandino, G. Pegg, F. Gonzalez, and G. Smith, “Aerial mapping of forests affected by pathogens using UAVs, hyperspectral sensors, and artificial intelligence,” *Sensors*, vol. 18, no. 4, p. 944, Mar. 2018. DOI: [10.3390/s18040944](#) (cit. on pp. [1](#), [9](#), [11](#)).
- [6] J. Jiménez López and M. Mulero-Pázmány, “Drones for conservation in protected areas: Present and future,” *Drones*, vol. 3, no. 1, p. 10, Jan. 2019. DOI: [10.3390/drones3010010](#) (cit. on pp. [1](#), [13](#)).
- [7] M. Erdelj and E. Natalizio, “UAV-assisted disaster management: Applications and open issues,” in *International Conference on Computing, Networking and Communications*, Kauai, HI, USA: IEEE, Feb. 2016, pp. 1–5. DOI: [10.1109/ICCNC.2016.7440563](#) (cit. on pp. [1](#), [6](#), [13](#)).
- [8] M. G. Pensieri, M. Garau, and P. M. Barone, “Drones as an integral part of remote sensing technologies to help missing people,” *Drones*, vol. 4, no. 2, p. 15, Apr. 2020. DOI: [10.3390/drones4020015](#) (cit. on p. [1](#)).
- [9] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegł, *On Integrating Unmanned Aircraft Systems into the National Airspace System*. Dordrecht: Springer Netherlands, 2012, pp. 1–305. DOI: [10.1007/978-94-007-2479-2](#) (cit. on pp. [1](#), [12](#)).
- [10] C. T. Recchiuto and A. Sgorbissa, “Post-disaster assessment with unmanned aerial vehicles: A survey on practical implementations and research approaches,” *Journal of Field Robotics*, vol. 35, no. 4, pp. 459–490, Jun. 2018. DOI: [10.1002/rob.21756](#) (cit. on pp. [2](#), [14](#), [15](#)).
- [11] S. Mayer, L. Lischke, and P. W. Woźniak, “Drones for search and rescue,” in *1st International Workshop on Human-Drone Interaction*, Glasgow, United Kingdom: Ecole Nationale de l’Aviation Civile [ENAC], May 2019, pp. 1–7 (cit. on p. [2](#)).

- [12] K. P. Valavanis and G. J. Vachtsevanos, "Future of unmanned aviation," in *Handbook of Unmanned Aerial Vehicles*, K. P. Valavanis and G. J. Vachtsevanos, Eds., Dordrecht: Springer Netherlands, 2015, ch. 126, pp. 2993–3009. DOI: [10.1007/978-90-481-9707-1_95](https://doi.org/10.1007/978-90-481-9707-1_95) (cit. on pp. 2, 23).
- [13] W. T. Weldon and J. Hupy, "Investigating methods for integrating unmanned aerial systems in search and rescue operations," *Drones*, vol. 4, no. 3, p. 38, Jul. 2020. DOI: [10.3390/drones4030038](https://doi.org/10.3390/drones4030038) (cit. on p. 2).
- [14] F. Vanegas and F. Gonzalez, "Enabling UAV navigation with sensor and environmental uncertainty in cluttered and GPS-denied environments," *Sensors*, vol. 16, no. 5, p. 666, May 2016. DOI: [10.3390/s16050666](https://doi.org/10.3390/s16050666) (cit. on pp. 2, 15, 20).
- [15] H. Zhang, S. Veres, and A. Kolling, "Simultaneous search and monitoring by unmanned aerial vehicles," in *Annual Conference on Decision and Control*, Melbourne, Australia: IEEE, Dec. 2017, pp. 903–910. DOI: [10.1109/CDC.2017.8263774](https://doi.org/10.1109/CDC.2017.8263774) (cit. on p. 2).
- [16] C. Sampedro, A. Rodriguez-Ramos, H. Bavle, A. Carrio, P. de la Puente, and P. Campoy, "A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques," *Journal of Intelligent & Robotic Systems*, pp. 1–27, Jul. 2018. DOI: [10.1007/s10846-018-0898-1](https://doi.org/10.1007/s10846-018-0898-1) (cit. on pp. 2, 3, 15).
- [17] R. Murphy, "Human–robot interaction in rescue robotics," *Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 2, pp. 138–153, May 2004. DOI: [10.1109/TSMCC.2004.826267](https://doi.org/10.1109/TSMCC.2004.826267) (cit. on p. 3).
- [18] T. Tomic, K. Schmid, P. Lutz, *et al.*, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 46–56, Sep. 2012. DOI: [10.1109/MRA.2012.2206473](https://doi.org/10.1109/MRA.2012.2206473) (cit. on pp. 3, 24).
- [19] V. San Juan, M. Santos, and J. M. Andújar, "Intelligent UAV map generation and discrete path planning for search and rescue operations," *Complexity*, vol. 2018, pp. 1–17, Apr. 2018. DOI: [10.1155/2018/6879419](https://doi.org/10.1155/2018/6879419) (cit. on pp. 3, 15).
- [20] A. Wada, T. Yamashita, M. Maruyama, T. Arai, H. Adachi, and H. Tsuji, "A surveillance system using small unmanned aerial vehicle (UAV) related technologies," *NEC Technical Journal*, vol. 8, no. 1, pp. 68–73, 2013 (cit. on p. 3).
- [21] P. Molina, M. P. Eulalia, I. Colomina, *et al.*, "Drones to the rescue!" *Inside GNSS*, vol. Jul./Aug. Pp. 38–47, 2012 (cit. on p. 3).
- [22] D. Giordan, Y. Hayakawa, F. Nex, F. Remondino, and P. Tarolli, "Review article: The use of remotely piloted aircraft systems (rpas) for natural hazards monitoring and management," *Natural Hazards and Earth System Sciences*, vol. 18, no. 4, pp. 1079–1096, Apr. 2018. DOI: [10.5194/nhess-18-1079-2018](https://doi.org/10.5194/nhess-18-1079-2018) (cit. on pp. 3, 24).
- [23] A. Al-Kaff, D. Martín, F. García, A. de la Escalera, and J. María Armingol, "Survey of computer vision algorithms and applications for unmanned aerial vehicles," *Expert Systems with Applications*, vol. 92, pp. 447–463, Feb. 2018. DOI: [10.1016/j.eswa.2017.09.033](https://doi.org/10.1016/j.eswa.2017.09.033) (cit. on pp. 5, 12).
- [24] S. Bricknell, "Missing persons: Who is at risk?" Australian Institute of Criminology, Canberra, Australia, Tech. Rep., 2017, pp. 1–38 (cit. on p. 6).
- [25] V. Jorge, R. Granada, R. Maidana, *et al.*, "A survey on unmanned surface vehicles for disaster robotics: Main challenges and directions," *Sensors*, vol. 19, no. 3, p. 702, Feb. 2019. DOI: [10.3390/s19030702](https://doi.org/10.3390/s19030702) (cit. on p. 6).

- [26] J. Scherer, B. Rinner, S. Yahyanejad, *et al.*, “An autonomous multi-UAV system for search and rescue,” in *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, New York, New York, USA: ACM Press, 2015, pp. 33–38. DOI: [10.1145/2750675.2750683](https://doi.org/10.1145/2750675.2750683) (cit. on p. 6).
- [27] M. Bejiga, A. Zeggada, A. Nouffidj, and F. Melgani, “A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery,” *Remote Sensing*, vol. 9, no. 2, p. 100, Jan. 2017. DOI: [10.3390/rs9020100](https://doi.org/10.3390/rs9020100) (cit. on pp. 6, 24).
- [28] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge, “Multipurpose UAV for search and rescue operations in mountain avalanche events,” *Geomatics, Natural Hazards and Risk*, vol. 8, no. 1, pp. 18–33, Jan. 2017. DOI: [10.1080/19475705.2016.1238852](https://doi.org/10.1080/19475705.2016.1238852) (cit. on p. 6).
- [29] R. Alterovitz, S. Koenig, and M. Likhachev, “Robot planning in the real world: Research challenges and opportunities,” *AI Magazine*, vol. 37, no. 2, p. 76, Jul. 2016. DOI: [10.1609/aimag.v37i2.2651](https://doi.org/10.1609/aimag.v37i2.2651) (cit. on p. 8).
- [30] R. K. Rangel and A. C. Terra, “Development of a surveillance tool using UAV’s,” in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2018, pp. 1–11. DOI: [10.1109/AERO.2018.8396603](https://doi.org/10.1109/AERO.2018.8396603) (cit. on p. 9).
- [31] R. Jurdak, A. Elfes, B. Kusy, *et al.*, “Autonomous surveillance for biosecurity,” *Trends in Biotechnology*, vol. 33, no. 4, pp. 201–207, Apr. 2015. DOI: [10.1016/j.tibtech.2015.01.003](https://doi.org/10.1016/j.tibtech.2015.01.003) (cit. on p. 9).
- [32] C. Anderson, S. Low-Choy, P. Whittle, *et al.*, “Australian plant biosecurity surveillance systems,” *Crop Protection*, vol. 100, pp. 8–20, Oct. 2017. DOI: [10.1016/j.cropro.2017.05.023](https://doi.org/10.1016/j.cropro.2017.05.023) (cit. on p. 9).
- [33] M. Shahbazi, J. Théau, and P. Ménard, “Recent applications of unmanned aerial imagery in natural resource management,” *GIScience & Remote Sensing*, vol. 51, no. 4, pp. 339–365, Jul. 2014. DOI: [10.1080/15481603.2014.926650](https://doi.org/10.1080/15481603.2014.926650) (cit. on p. 9).
- [34] F. Vanegas, D. Bratanov, K. Powell, J. Weiss, and F. Gonzalez, “A novel methodology for improving plant pest surveillance in vineyards and crops using UAV-based hyperspectral and spatial data,” *Sensors*, vol. 18, no. 1, p. 260, Jan. 2018. DOI: [10.3390/s18010260](https://doi.org/10.3390/s18010260) (cit. on pp. 9, 11, 23).
- [35] A. Rees, L. Avens, K. Ballorain, *et al.*, “The potential of unmanned aerial systems for sea turtle research and conservation: A review and future directions,” *Endangered Species Research*, vol. 35, pp. 81–100, Feb. 2018. DOI: [10.3354/esr00877](https://doi.org/10.3354/esr00877) (cit. on pp. 9, 11).
- [36] H. Chao, Y. Cao, and Y. Chen, “Autopilots for small unmanned aerial vehicles: A survey,” *International Journal of Control, Automation and Systems*, vol. 8, no. 1, pp. 36–44, Feb. 2010. DOI: [10.1007/s12555-010-0105-z](https://doi.org/10.1007/s12555-010-0105-z) (cit. on p. 9).
- [37] M. J. Kochenderfer, J. E. Holland, and J. P. Chryssanthacopoulos, “Next-generation airborne collision avoidance system,” *Lincoln Laboratory Journal*, vol. 19, no. 1, pp. 17–33, 2012 (cit. on p. 9).
- [38] W. Kong, D. Zhou, D. Zhang, and J. Zhang, “Vision-based autonomous landing system for unmanned aerial vehicle: A survey,” in *International Conference on Multisensor Fusion and Information Integration for Intelligent Systems*, IEEE, Sep. 2014, pp. 1–8. DOI: [10.1109/MFI.2014.6997750](https://doi.org/10.1109/MFI.2014.6997750) (cit. on p. 9).
- [39] A. Gautam, P. Sujit, and S. Saripalli, “Autonomous quadrotor landing using vision and pursuit guidance,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 501–10 506, Jul. 2017. DOI: [10.1016/j.ifacol.2017.08.1982](https://doi.org/10.1016/j.ifacol.2017.08.1982) (cit. on p. 9).

- [40] I. Maza, F. Caballero, J. Capitán, J. R. Martínez-de-Dios, and A. Ollero, “Experimental results in multi-UAV coordination for disaster management and civil security applications,” *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 563–585, Jan. 2011. DOI: [10.1007/s10846-010-9497-5](https://doi.org/10.1007/s10846-010-9497-5) (cit. on p. 9).
- [41] J. Cruz, D. Sanchez, and G. Pajares, *System for guiding an unmanned vehicle towards a platform using visual analysis*, 2012. [Online]. Available: <https://patentimages.storage.googleapis.com/b1/82/26/545d9cfe53f827/ES2387144B2.pdf> (cit. on p. 9).
- [42] K. A. Ghamry, Y. Dong, M. A. Kamel, and Y. Zhang, “Real-time autonomous take-off, tracking and landing of UAV on a moving ugv platform,” in *Mediterranean Conference on Control and Automation*, IEEE, Jun. 2016, pp. 1236–1241. DOI: [10.1109/MED.2016.7535886](https://doi.org/10.1109/MED.2016.7535886) (cit. on pp. 9, 24).
- [43] S. Shapira, T. Levi, Y. Bar-Dayán, and L. Aharonson-Daniel, “The impact of behavior on the risk of injury and death during an earthquake: A simulation-based study,” *Natural Hazards*, vol. 91, no. 3, pp. 1059–1074, Apr. 2018. DOI: [10.1007/s11069-018-3167-5](https://doi.org/10.1007/s11069-018-3167-5) (cit. on p. 11).
- [44] S. Doocy, A. Daniels, C. Packer, A. Dick, and T. D. Kirsch, “The human impact of earthquakes: A historical review of events 1980-2009 and systematic literature review.,” *PLoS currents*, vol. 5, no. APR 2013, Apr. 2013. DOI: [10.1371/currents.dis.67bd14fe457f1db0b5433a8ee20fb833](https://doi.org/10.1371/currents.dis.67bd14fe457f1db0b5433a8ee20fb833) (cit. on p. 11).
- [45] R. Z. B. Bravo, A. Leiras, and F. L. Cyrino Oliveira, “The use of UAVs in humanitarian relief: An application of POMDP-based methodology for finding victims,” *Production and Operations Management*, vol. 28, no. 2, pp. 421–440, Feb. 2019. DOI: [10.1111/poms.12930](https://doi.org/10.1111/poms.12930) (cit. on pp. 11, 20).
- [46] Australian National Search and Rescue Council, “Volume 2 - search and rescue operations,” in *National Search and Rescue Manual*, February, 2020th ed., Canberra, Australia: Australian Maritime Safety Authority (AMSA), 2020, ch. 3, pp. 73–353 (cit. on pp. 11, 146).
- [47] U. Chopra, *Drone disaster relief - how drones are used for disaster response - blogs - diydrones*, 2021. [Online]. Available: <https://diydrones.com/profiles/blogs/drone-disaster-relief> (visited on 11/29/2021) (cit. on p. 12).
- [48] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, “A review of deep learning methods and applications for unmanned aerial vehicles,” *Journal of Sensors*, vol. 2017, pp. 1–13, 2017. DOI: [10.1155/2017/3296874](https://doi.org/10.1155/2017/3296874) (cit. on pp. 12, 13, 15, 23).
- [49] M. Bouhali, F. Shamani, Z. E. Dahmane, A. Belaidi, and J. Nurmi, “FPGA applications in unmanned aerial vehicles - a review,” in *Lecture Notes in Computer Science*, vol. 10216 LNCS, 2017, pp. 217–228. DOI: [10.1007/978-3-319-56258-2_19](https://doi.org/10.1007/978-3-319-56258-2_19) (cit. on p. 12).
- [50] R. Bähneemann, M. Pantic, M. Popović, *et al.*, “The eth-mav team in the mbz international robotics challenge,” *Journal of Field Robotics*, vol. 36, no. 1, pp. 78–103, Jan. 2019. DOI: [10.1002/rob.21824](https://doi.org/10.1002/rob.21824) (cit. on pp. 13, 18).
- [51] N. Mandel, M. Milford, and F. Gonzalez, “A method for evaluating and selecting suitable hardware for deployment of embedded system on UAVs,” *Sensors*, vol. 20, no. 16, p. 4420, Aug. 2020. DOI: [10.3390/s20164420](https://doi.org/10.3390/s20164420) (cit. on pp. 13, 24).
- [52] E. Faniadis and A. Amanatiadis, “Deep learning inference at the edge for mobile and aerial robotics,” in *International Symposium on Safety, Security, and Rescue Robotics*, Abu Dhabi, UAE: IEEE, Nov. 2020, pp. 334–340. DOI: [10.1109/SSRR50563.2020.9292575](https://doi.org/10.1109/SSRR50563.2020.9292575) (cit. on p. 13).

- [53] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005, pp. 485–542 (cit. on pp. 13, 15, 16, 19).
- [54] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018, pp. 1–548 (cit. on pp. 13, 17).
- [55] F. Kendoul, “Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems,” *Journal of Field Robotics*, vol. 29, no. 2, pp. 315–378, Mar. 2012. DOI: [10.1002/rob.20414](https://doi.org/10.1002/rob.20414). arXiv: [10.1.1.91.5767](https://arxiv.org/abs/10.1.1.91.5767) (cit. on pp. 13, 15).
- [56] Y. Zhao, Z. Zheng, and Y. Liu, “Survey on computational-intelligence-based UAV path planning,” *Knowledge-Based Systems*, vol. 158, pp. 54–64, Oct. 2018. DOI: [10.1016/j.knosys.2018.05.033](https://doi.org/10.1016/j.knosys.2018.05.033) (cit. on pp. 14, 15).
- [57] S. Lohr, *Is there a smarter path to artificial intelligence? some experts hope so*, New York, USA, Jun. 2018. [Online]. Available: <https://www.nytimes.com/2018/06/20/technology/deep-learning-artificial-intelligence.html> (cit. on p. 15).
- [58] G. Marcus, “Deep learning: A critical appraisal,” pp. 1–27, Jan. 2018. arXiv: [1801.00631](https://arxiv.org/abs/1801.00631) (cit. on p. 15).
- [59] D. Silver, A. Huang, C. J. Maddison, *et al.*, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016. DOI: [10.1038/nature16961](https://doi.org/10.1038/nature16961). arXiv: [1206.2944](https://arxiv.org/abs/1206.2944) (cit. on pp. 15, 21).
- [60] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, “Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search,” in *International Conference on Robotics and Automation*, vol. June, IEEE, May 2016, pp. 528–535. DOI: [10.1109/ICRA.2016.7487175](https://doi.org/10.1109/ICRA.2016.7487175). arXiv: [1509.06791](https://arxiv.org/abs/1509.06791) (cit. on p. 15).
- [61] J. G. C. Zuluaga, J. P. Leidig, C. Trefftz, and G. Wolffe, “Deep reinforcement learning for autonomous search and rescue,” in *National Aerospace and Electronics Conference*, vol. 2018-July, IEEE, Jul. 2018, pp. 521–524. DOI: [10.1109/NAECON.2018.8556642](https://doi.org/10.1109/NAECON.2018.8556642) (cit. on p. 15).
- [62] O. Walker, F. Vanegas, F. Gonzalez, and S. Koenig, “A deep reinforcement learning framework for UAV navigation in indoor environments,” in *Aerospace Conference*, vol. 2019-March, Big Sky, MT, US: IEEE, Mar. 2019, pp. 1–14. DOI: [10.1109/AERO.2019.8742226](https://doi.org/10.1109/AERO.2019.8742226) (cit. on pp. 15, 24).
- [63] S. Kulkarni, V. Chaphekar, M. M. Uddin Chowdhury, F. Erden, and I. Guvenc, “UAV aided search and rescue operation using reinforcement learning,” in *SoutheastCon*, vol. 2, Raleigh, NC, USA: IEEE, Mar. 2020, pp. 1–8. DOI: [10.1109/SoutheastCon44009.2020.9368285](https://doi.org/10.1109/SoutheastCon44009.2020.9368285). arXiv: [2002.08415](https://arxiv.org/abs/2002.08415) (cit. on p. 15).
- [64] F. Vanegas, J. Roberts, and F. Gonzalez, “UAV tracking of mobile target in occluded, cluttered and GPS-denied environments,” in *Aerospace Conference*, IEEE, Mar. 2018, pp. 1–7. DOI: [10.1109/AERO.2018.8396449](https://doi.org/10.1109/AERO.2018.8396449) (cit. on pp. 15, 24, 145).
- [65] F. Vanegas-Alvarez, “Uncertainty based online planning for UAV missions in GPS-denied and cluttered environments,” PhD Thesis, Queensland University of Technology, 2017, pp. 1–170. DOI: [10.5204/thesis.eprints.103846](https://doi.org/10.5204/thesis.eprints.103846) (cit. on pp. 15, 23).
- [66] F. Vanegas and F. Gonzalez, “Uncertainty based online planning for UAV target finding in cluttered and GPS-denied environments,” in *Aerospace Conference*, vol. 2016-June, Big Sky, MT, USA: IEEE, Mar. 2016, pp. 1–9. DOI: [10.1109/AERO.2016.7500566](https://doi.org/10.1109/AERO.2016.7500566) (cit. on p. 15).
- [67] U. Ilhan, L. Gardashova, and K. Kilic, “UAV using dec-POMDP model for increasing the level of security in the company,” *Procedia Computer Science*, vol. 102, pp. 458–464, 2016. DOI: [10.1016/j.procs.2016.09.427](https://doi.org/10.1016/j.procs.2016.09.427) (cit. on pp. 15, 20).

- [68] Y. Rizk, M. Awad, and E. W. Tunstel, "Decision making in multiagent systems: A survey," *Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 514–529, Sep. 2018. DOI: [10.1109/TCDS.2018.2840971](https://doi.org/10.1109/TCDS.2018.2840971) (cit. on p. 15).
- [69] E. M. Atkins, A. Ollero, and A. Tsourdos, Eds., *Unmanned Aircraft Systems*. New York, USA: Wiley, 2016, pp. 1–710 (cit. on p. 18).
- [70] A. Dutech and B. Scherrer, "Partially observable markov decision processes," in *Markov Decision Processes in Artificial Intelligence*, O. Sigaud and O. Buffet, Eds., Hoboken, NJ USA: John Wiley & Sons, Inc., Mar. 2013, ch. 7, pp. 185–228. DOI: [10.1002/9781118557426.ch7](https://doi.org/10.1002/9781118557426.ch7) (cit. on p. 19).
- [71] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of markov decision processes," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441–450, Aug. 1987. DOI: [10.1287/moor.12.3.441](https://doi.org/10.1287/moor.12.3.441) (cit. on p. 20).
- [72] S. A. Miller, Z. A. Harris, and E. K. Chong, "A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 1, p. 724 597, Dec. 2009. DOI: [10.1155/2009/724597](https://doi.org/10.1155/2009/724597) (cit. on p. 20).
- [73] S. Ragi and E. K. P. Chong, "UAV path planning in a dynamic environment via partially observable markov decision process," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2397–2412, Oct. 2013. DOI: [10.1109/TAES.2013.6621824](https://doi.org/10.1109/TAES.2013.6621824) (cit. on p. 20).
- [74] Y. Zhao, X. Wang, W. Kong, L. Shen, and S. Jia, "Decision-making of UAV for tracking moving target via information geometry," in *Chinese Control Conference*, Chengdu, China: IEEE, Jul. 2016, pp. 5611–5617. DOI: [10.1109/ChiCC.2016.7554231](https://doi.org/10.1109/ChiCC.2016.7554231) (cit. on p. 20).
- [75] C. M. Eaton, E. K. Chong, and A. A. Maciejewski, "Robust UAV path planning using POMDP with limited fov sensor," in *Conference on Control Technology and Applications (CCTA)*, Hawaii, USA: IEEE, Aug. 2017, pp. 1530–1535. DOI: [10.1109/CCTA.2017.8062674](https://doi.org/10.1109/CCTA.2017.8062674) (cit. on p. 20).
- [76] M. J. Kochenderfer, *Decision making under uncertainty: theory and application*. Cambridge, MA: MIT Press, 2015, pp. 1–323 (cit. on p. 20).
- [77] F. Vanegas, D. Campbell, M. Eich, and F. Gonzalez, "UAV based target finding and tracking in GPS-denied and cluttered environments," in *International Conference on Intelligent Robots and Systems*, Daejeon, South Korea: IEEE/RSJ, Oct. 2016, pp. 2307–2313. DOI: [10.1109/IROS.2016.7759360](https://doi.org/10.1109/IROS.2016.7759360) (cit. on p. 20).
- [78] S. Ragi and E. K. P. Chong, "UAV guidance algorithms via partially observable markov decision processes," in *Handbook of Unmanned Aerial Vehicles*, K. Valavanis and G. Vachtsevanos, Eds., Dordrecht: Springer Netherlands, 2015, ch. 73, pp. 1775–1810. DOI: [10.1007/978-90-481-9707-1_59](https://doi.org/10.1007/978-90-481-9707-1_59) (cit. on p. 20).
- [79] C. Ponzoni Carvalho Chanel, A. Albore, J. T'Hooft, C. Lesire, and F. Teichteil-Königsbuch, "Ample: An anytime planning and execution framework for dynamic and uncertain problems in robotics," *Autonomous Robots*, vol. 43, no. 1, pp. 37–62, Jan. 2019. DOI: [10.1007/s10514-018-9703-z](https://doi.org/10.1007/s10514-018-9703-z) (cit. on p. 20).
- [80] M. Chen, E. Frazzoli, D. Hsu, and W. S. Lee, "POMDP-lite for robust robot planning under uncertainty," in *International Conference on Robotics and Automation*, Stockholm, Sweden: IEEE, Feb. 2016, pp. 5427–5433. DOI: [10.1109/ICRA.2016.7487754](https://doi.org/10.1109/ICRA.2016.7487754) (cit. on p. 20).
- [81] S. C. W. Ong, S. W. Png, D. Hsu, and W. S. Lee, "Pomdps for robotic tasks with mixed observability," in *Robotics: Science and Systems V*, Seattle, US: Robotics: Science and Systems Foundation, Jun. 2009. DOI: [10.15607/RSS.2009.V.026](https://doi.org/10.15607/RSS.2009.V.026) (cit. on p. 20).

- [82] Z. N. Sunberg and M. J. Kochenderfer, "Online algorithms for pomdps with continuous state, action, and observation spaces," in *International Conference on Automated Planning and Scheduling*, Delft, Netherlands: AAAI Press, Jun. 2018, pp. 259–263 (cit. on p. 20).
- [83] C. M. Eaton, "Autonomous UAV control and testing methods utilizing partially observable markov decision processes," PhD Thesis, Colorado State University, 2018, p. 115 (cit. on p. 20).
- [84] L. W. Krakow, C. M. Eaton, and E. K. P. Chong, "Simultaneous non-myopic optimization of UAV guidance and camera gimbal control for target tracking," in *Conference on Control Technology and Applications (CCTA)*, Copenhagen, Denmark: IEEE, Aug. 2018, pp. 349–354. DOI: [10.1109/CCTA.2018.8511346](https://doi.org/10.1109/CCTA.2018.8511346) (cit. on p. 20).
- [85] Q. Yang, J. Zhang, and G. Shi, "Path planning for unmanned aerial vehicle passive detection under the framework of partially observable markov decision process," in *Chinese Control And Decision Conference*, IEEE, Jun. 2018, pp. 3896–3903. DOI: [10.1109/CCDC.2018.8407800](https://doi.org/10.1109/CCDC.2018.8407800) (cit. on p. 20).
- [86] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds., Vancouver, BC, Canada: Curran Associates, Inc., 2010, pp. 2164–2172 (cit. on pp. 21, 23).
- [87] H. Kurniawati and V. Yadav, "An online POMDP solver for uncertainty planning in dynamic environment," in *Robotics Research. Springer Tracts in Advanced Robotics*, M. Inaba and P. Corke, Eds., vol. 114, Springer, Cham, 2016, pp. 611–629. DOI: [10.1007/978-3-319-28872-7_35](https://doi.org/10.1007/978-3-319-28872-7_35) (cit. on pp. 21, 23).
- [88] J. Sandino, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez, "Drone-based autonomous motion planning system for outdoor environments under object detection uncertainty," *Remote Sensing*, vol. 13, no. 21, p. 4481, Nov. 2021. DOI: [10.3390/rs13214481](https://doi.org/10.3390/rs13214481) (cit. on p. 21).
- [89] D. Klimenko, J. Song, and H. Kurniawati, "Tapir: A software toolkit for approximating and adapting POMDP solutions online," in *Australasian Conference on Robotics and Automation*, Melbourne, Australia: ARAA, Dec. 2014, pp. 1–9 (cit. on p. 22).
- [90] C. Chanel, F. Teichteil-Königsbuch, and C. Lesire, "Multi-target detection and recognition by UAVs using online pomdps," in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, Bellevue, Washington: AAAI Press, 2013, pp. 1381–1387 (cit. on pp. 23, 24).
- [91] L. Lopez-Fuentes, J. van de Weijer, M. González-Hidalgo, H. Skinnemoen, and A. D. Bagdanov, "Review on computer vision techniques in emergency situations," *Multimedia Tools and Applications*, vol. 77, no. 13, pp. 17 069–17 107, Jul. 2018. DOI: [10.1007/s11042-017-5276-7](https://doi.org/10.1007/s11042-017-5276-7). [Online]. Available: <http://link.springer.com/10.1007/s11042-017-5276-7> (cit. on p. 23).
- [92] H. Choi, M. Geeves, B. Alsalam, and F. Gonzalez, "Open source computer-vision based guidance system for UAVs on-board decision making," in *Aerospace Conference*, Big Sky, MT, US: IEEE, Mar. 2016, pp. 1–5. DOI: [10.1109/AERO.2016.7500600](https://doi.org/10.1109/AERO.2016.7500600) (cit. on p. 24).
- [93] B. H. Y. Alsalam, K. Morton, D. Campbell, and F. Gonzalez, "Autonomous UAV with vision based on-board decision making for remote sensing and precision agriculture," in *Aerospace Conference*, IEEE, Mar. 2017, pp. 1–12. DOI: [10.1109/AERO.2017.7943593](https://doi.org/10.1109/AERO.2017.7943593) (cit. on p. 24).
- [94] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Computer Communications*, vol. 156, pp. 1–10, Apr. 2020. DOI: [10.1016/j.comcom.2020.03.012](https://doi.org/10.1016/j.comcom.2020.03.012) (cit. on pp. 24, 146).

- [95] Hossain and Lee, "Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with gpu-based embedded devices," *Sensors*, vol. 19, no. 15, p. 3371, Jul. 2019. DOI: [10.3390/s19153371](https://doi.org/10.3390/s19153371) (cit. on p. 24).
- [96] E. Lygouras, N. Santavas, A. Taitzoglou, K. Tarchanidis, A. Mitropoulos, and A. Gasteratos, "Unsupervised human detection with an embedded vision system on a fully autonomous UAV for search and rescue operations," *Sensors*, vol. 19, no. 16, p. 3542, Aug. 2019. DOI: [10.3390/s19163542](https://doi.org/10.3390/s19163542) (cit. on p. 24).
- [97] B. Lindqvist, C. Kanellakis, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, *Compra: A compact reactive autonomy framework for subterranean mav based search-and-rescue operations*, Aug. 2021. arXiv: [2108.13105](https://arxiv.org/abs/2108.13105). [Online]. Available: <http://arxiv.org/abs/2108.13105> (cit. on p. 24).
- [98] A. Tullu, B. Endale, A. Wondosen, and H.-Y. Hwang, "Machine learning approach to real-time 3d path planning for autonomous navigation of unmanned aerial vehicle," *Applied Sciences*, vol. 11, no. 10, p. 4706, May 2021. DOI: [10.3390/app11104706](https://doi.org/10.3390/app11104706) (cit. on p. 24).
- [99] A. Devos, E. Ebeid, and P. Manoonpong, "Development of autonomous drones for adaptive obstacle avoidance in real world environments," in *Euromicro Conference on Digital System Design*, IEEE, Aug. 2018, pp. 707–710. DOI: [10.1109/DSD.2018.00009](https://doi.org/10.1109/DSD.2018.00009) (cit. on p. 24).
- [100] J. Galvez-Serna, N. Mandel, J. Sandino, *et al.*, "Real-time segmentation of desiccation cracks onboard UAVs for planetary exploration," in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2022, (accepted) (cit. on p. 143).
- [101] N. Mandel, J. Sandino, J. Galvez-Serna, F. Vanegas, and F. Gonzalez, "Resolution-adaptive quadrees for semantic segmentation mapping in UAV applications," in *Aerospace Conference*, Big Sky, MT, USA: IEEE, Mar. 2022, (accepted) (cit. on p. 144).
- [102] Y. Chuanqi, *Caffe implementation of google mobilenet ssd detection network, with pretrained weights on voc0712 and map=0.727*. 2020. [Online]. Available: <https://github.com/chuanqi305/MobileNet-SSD> (visited on 08/30/2020) (cit. on p. 146).
- [103] P. Mittal, R. Singh, and A. Sharma, "Deep learning-based object detection in low-altitude UAV datasets: A survey," *Image and Vision Computing*, vol. 104, p. 104 046, Dec. 2020. DOI: [10.1016/j.imavis.2020.104046](https://doi.org/10.1016/j.imavis.2020.104046) (cit. on p. 146).
- [104] M. Hoerger and H. Kurniawati, "An on-line POMDP solver for continuous observation spaces," *CoRR*, vol. abs/2011.0, Nov. 2020. arXiv: [2011.02076](https://arxiv.org/abs/2011.02076) (cit. on p. 146).
- [105] M. Egorov, Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta, and M. J. Kochenderfer, "POMDPs.jl: A framework for sequential decision making under uncertainty," *Journal of Machine Learning Research*, vol. 18, no. 26, pp. 1–5, 2017 (cit. on p. 146).