



## Article

# Framework for Autonomous UAV Navigation and Target Detection in Global-Navigation-Satellite-System-Denied and Visually Degraded Environments

Sebastien Boiteau <sup>1,2,\*</sup>, Fernando Vanegas <sup>1,2</sup> and Felipe Gonzalez <sup>1,2</sup>

<sup>1</sup> 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.); felipe.gonzalez@qut.edu.au (F.G.)

<sup>2</sup> QUT Centre for Robotics (QCR), Queensland University of Technology (QUT), Level 11, S Block, 2 George Street, Brisbane City, QLD 4000, Australia

\* Correspondence: sebastienthierry.boiteau@hdr.qut.edu.au

**Abstract:** Autonomous Unmanned Aerial Vehicles (UAVs) have possible applications in wildlife monitoring, disaster monitoring, and emergency Search and Rescue (SAR). Autonomous capabilities such as waypoint flight modes and obstacle avoidance, as well as their ability to survey large areas, make UAVs the prime choice for these critical applications. However, autonomous UAVs usually rely on the Global Navigation Satellite System (GNSS) for navigation and normal visibility conditions to obtain observations and data on their surrounding environment. These two parameters are often lacking due to the challenging conditions in which these critical applications can take place, limiting the range of utilisation of autonomous UAVs. This paper presents a framework enabling a UAV to autonomously navigate and detect targets in GNSS-denied and visually degraded environments. The navigation and target detection problem is formulated as an autonomous Sequential Decision Problem (SDP) with uncertainty caused by the lack of the GNSS and low visibility. The SDP is modelled as a Partially Observable Markov Decision Process (POMDP) and tested using the Adaptive Belief Tree (ABT) algorithm. The framework is tested in simulations and real life using a navigation task based on a classic SAR operation in a cluttered indoor environment with different visibility conditions. The framework is composed of a small UAV with a weight of 5 kg, a thermal camera used for target detection, and an onboard computer running all the computationally intensive tasks. The results of this study show the robustness of the proposed framework to autonomously explore and detect targets using thermal imagery under different visibility conditions. Devising UAVs that are capable of navigating in challenging environments with degraded visibility can encourage authorities and public institutions to consider the use of autonomous remote platforms to locate stranded people in disaster scenarios.

**Keywords:** partially observable Markov decision process; unmanned aerial vehicles; search and rescue; low visibility; embedded systems; remote sensing; motion planner



**Citation:** Boiteau, S.; Vanegas, F.; Gonzalez, F. Framework for Autonomous UAV Navigation and Target Detection in Global-Navigation-Satellite-System-Denied and Visually Degraded Environments. *Remote Sens.* **2024**, *16*, 471. <https://doi.org/10.3390/rs16030471>

Academic Editors: Francisco Javier Mesas Carrascosa, José Emilio Meroño-Larriva and María Jesús Aguilera-Ureña

Received: 27 November 2023

Revised: 19 January 2024

Accepted: 24 January 2024

Published: 25 January 2024



**Copyright:** © 2024 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

Over 50 years, the number of natural disasters driven by climate change and more extreme weather has increased by a factor of five [1]. Disaster events caused by natural hazards affected approximately 100 million people and resulted in 15,082 deaths in 2020 alone [2]. Therefore, it is important to improve Search and Rescue technologies that could assist the rescuers in locating victims, especially in challenging environments where human interventions are often dangerous and unfeasible.

Autonomous UAVs can be utilised to remotely operate in challenging environments. As an example, autonomous UAVs are used in critical applications such as surveillance [3], SAR [4], and mining [5]. These critical applications increase in complexity in environments

characterised by the absence of the GNSS and the presence of visual obstructions in the form of low luminosity, smoke, and dust. These particular conditions are a great challenge to UAV autonomous navigation, decision making, and target detection. Possible applications for autonomous UAVs are SAR operations such as the UAV Challenge [6] and subterranean operations like the DARPA SubT Challenge [7]. Australia also started using drones to detect sharks to learn more about their behaviours and to notify lifeguards of their presence [8].

Autonomous perception and localisation in low-visibility conditions without the GNSS require the use of multiple sensors, allowing the agent to perceive its surrounding environment. A framework for UAV navigation under degraded visibility was developed in [9–12]. The objective was to allow a small UAV to autonomously explore and navigate in subterranean environments in the presence of visual obstructions. The final framework combined the use of 3D (three-dimensional) LIDAR (Light Detection and Ranging), a thermal camera, an RGB (red, green, blue) camera and an IMU to perform odometry and Simultaneous Localisation and Mapping (SLAM). The LIDAR was used for the LIDAR odometry and mapping (LOAM) method [13], and both cameras were used for odometry only. The agent was able to explore cluttered, low-light environments in the presence of smoke and dust, without a GNSS signal. The main limitation of these papers was the restricted use of their framework in cluttered and indoor environments, where LIDAR is extremely efficient. Another gap in the knowledge from these papers is the lack of uncertainty modelling. Subterranean environments can cause uncertainty in observations, states, and actions due to their challenging conditions. An analysis of the use of multiple low-cost on-board sensors for ground robots or drones navigating in visually degraded environments was proposed by Sizintsev et al. [14]. An IMU, stereo cameras with LED lights, active infrared (IR) cameras, and 2D (two-dimensional) LIDAR were successfully tested on a ground robot, but not on a UAV.

The use of thermal imagery for object detection is a possible solution for SAR missions operating in low-visibility conditions such as smoke, fog, or low luminosity (night). Jiang et al. [15] proposed a UAV thermal infrared object detection framework using You Only Look Once (YOLO) models. The FLIR Thermal Starter Dataset [16] was used to train the model, which contains four classes: person, car, bicycle, and dog. Their research consisted of car and person multi-object detection experiments using YOLOv3, YOLOv4, and YOLOv5 models. The conclusion drawn from these experiments was that the YOLOv5 model can be used on board a UAV due to its small size, relative precision, and speed of detection. Similarly, Dong et al. [17] and Lagman et al. [18] used YOLO models for human detection using thermal imagery.

Decision making under uncertainty is the process of an agent receiving an incomplete or noisy observation at a precise time and then choosing an action based on this observation [19]. In SAR missions taking place under challenging conditions, the robotic agent is subjected to uncertainty due to the possible lack of the GNSS and by the presence of visual obstructions. Uncertainty and partial observability are modelled in two mathematical frameworks, the Markov Decision Process (MDP), and POMDPs. POMDPs were proven to be viable in UAV frameworks for autonomous navigation under partial observability and uncertainty [20–23]. Sandino et al. [24] proposed a framework for UAV autonomous navigation under uncertainty and partial observability from imperfect sensor readings in cluttered indoor scenarios. The navigation problem was modelled as a POMDP and was solved in real time with the ABT solver. Only colour imagery was used to detect the target. In later work, Sandino et al. [25] proposed a framework for UAV autonomous navigation in outdoor environments. The navigation problem was modelled with uncertainty and using a POMDP. Colour and thermal imagery were used to make the system more robust against the environmental conditions. Multiple flight modes were tested: a classic motion planner working with a list of position and velocity waypoints creating a survey zone, the POMDP motion planner, and a fusion of the two flight modes. The developed framework was able to find an adult mannequin by reducing the object detection uncertainty. Both research papers used the ABT online POMDP solver [26]. A gap in the knowledge highlighted by

both these works is the lack of testing in different visibility conditions, as the frameworks were not tested under poor visibility conditions.

This paper presents a framework for autonomous UAV navigation, exploration and target finding in low-visibility environments without the GNSS. The navigation problem is mathematically formulated using a POMDP, which models action and state uncertainty with probabilistic distributions. The navigation task is inspired by SAR in challenging environments. A UAV with a weight of 5 kg is deployed in a cluttered indoor environment to explore and locate a victim. Thermal imagery is used to detect the heat signature of a person under low-visibility conditions. This work evaluates the performance of the proposed framework via Software in the Loop (SITL) simulations, Hardware in the Loop (HIL), and real-life testing (RLT).

## 2. Background

This section covers the theory and principles of POMDPs and the Adaptive Belief Tree, which is the POMDP online solver used in this research.

### 2.1. POMDP

For this research problem, the navigation problem was modelled as a Sequential Decision Problem. As the testing environment is characterised by a low luminosity and the absence of the GNSS, modelling and accounting for uncertainty was a fundamental feature required in this work. The POMDP, a mathematical framework that models decision making under uncertainty in motion and action in a non-fully observable environment [27,28], was selected.

A POMDP is modelled using these parameters:  $(S, A, O, T, Z, R, \gamma)$  [29].  $S$  is the state space, a finite set of states representing the possible conditions of the agent and of the environment.  $A$  is a finite set of actions that the agent can execute to go from one state to another.  $O$  is a finite set of observations that the UAV can perceive.  $T$  is the transition function, modelling the transition of the agent from one state to the next after performing an action  $a$ .  $Z$  is the distribution function, modelling the probability of observing  $o$  from state  $s$  after executing an action  $a$ .  $R$  is a finite set of rewards, and  $\gamma$  is the discount factor with  $\gamma \in [0, 1]$ . In a POMDP, the robot state is not represented by a single estimation, but by a probability distribution called a belief state  $b$ , with  $B$  being the set of all possible belief states.

The objective of a POMDP is to determine a sequence of actions given a current belief state  $b$  that maximises the discounted accumulated reward. This sequence of actions is called a policy and is represented by the symbol  $\pi$ . The discounted accumulated reward is the sum of all the discounted rewards from each action executed during the mission. The aim of the POMDP is to find an optimal policy  $\pi^* : B \rightarrow A$  that maps belief states to actions and that maximises the total expected discounted return. Equation (1) represents the mathematical expression of the optimal policy.

$$\pi^* : \underset{\pi}{\operatorname{argmax}} \left( E \left[ \sum_{r=0}^{\infty} \gamma^r R(S^{tr}, \pi(b^{tr})) | b^{t0}, \pi \right] \right) \quad (1)$$

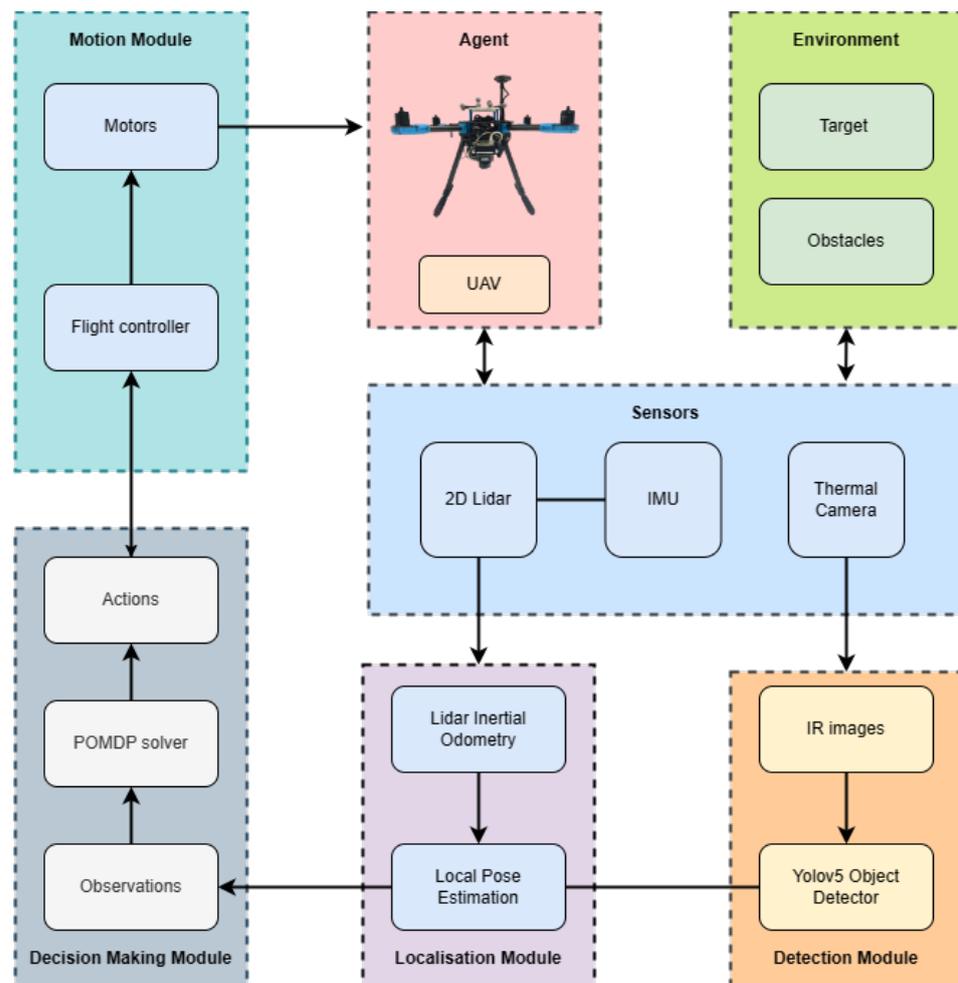
### 2.2. ABT

In this research, the ABT online POMDP solver [26] was selected. Online POMDP solvers are characterised by their ability to update the model during the execution, while only the known part of the environment and its dynamics are modelled. Most of the current online POMDP solvers have one main limitation: they recompute policies at every time step over again, thus wasting computational resources. This restriction heavily impacts platforms constrained in size and power, such as small UAVs. The ABT solver was selected for its ability to reuse and improve the previous policy when there are changes in the POMDP model. Moreover, the states and actions can be modelled in a continuous representation using a generative model.

The ABT algorithm contains two processes: the preprocess, which generates the first policy estimation offline, and the runtime process, which updates the previously computed policies. To start with, the agent executes the first action computed by the offline policy. Then, observations are collected using onboard sensors, and the ABT updates the online policy. The next action is then executed based on the updated policy.

### 3. System Architecture

The system architecture highlighted in Figure 1 was designed to allow a UAV to perform autonomous navigation in GNSS-denied and visually degraded environments. Decision making under uncertainty is executed following the POMDP representation shown in Section 2.1.



**Figure 1.** System architecture for autonomous UAV navigation in GNSS-denied and visually degraded environments. It is composed of a detection module processing raw IR frames from a thermal camera, a localisation module using LIDAR inertial odometry, a decision-making module sending an action to the flight controller, and a motion module controlling the dynamics of the UAV.

Multiple modules are used to distribute the different functions implemented in this framework. The detection module processes IR images from the thermal camera using a YOLOv5 object detector trained to detect the thermal signature of a human being. The localisation module uses 2D LIDAR and an IMU to perform odometry, allowing the agent to compute a local pose estimation. In this paper, the localisation module is yet to be implemented; however, the pose estimation uncertainty, representing the uncertainty in localisation, was modelled from real-life experiments. This will be covered in more depth in Section 4.6. The decision-making module is composed of the observations of the

environment made by the detection and localisation modules, which are then used by the POMDP solver. It then computes the optimal sequence of actions to accomplish the flight mission. These actions are fed to the motion module, which will manage the speed of the actuators to control the dynamics of the UAV.

#### 4. Framework Implementation

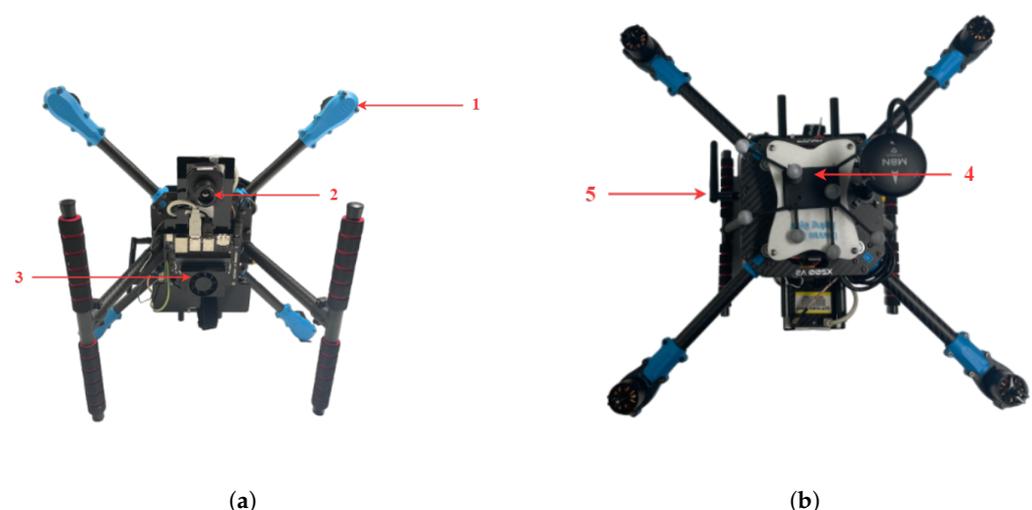
This section presents the hardware and software used to implement the framework as presented in Section 3. To begin with, the UAV frame and payload, as well as operating systems and communication interface of the system, are highlighted. Then, the decision-making module, consisting of the POMDP formulation, is presented. Finally, both the computer vision and decision-making modules are explained.

##### 4.1. UAV Hardware

The UAV frame used in this paper is a Holybro (Shenzhen City, China) X500 V2 multi-rotor kit. The motors are Holybro 2216 KV920 motors, with 1045 propellers (10"/254 mm length, 4.5"/114.3 mm pitch). The selected autopilot is the Pixhawk 6c autopilot from Holybro, which is the same flight controller unit (FCU) interface used in simulation and in real-life testing, and communication is performed with a Holybro 915 MHz telemetry radio. The dimensions of the UAV are 500 × 215 × 1440 mm, with a payload capacity of 1 kg. The UAV is powered using a four-cell 5000 mAh LiPo battery. This provides a flight autonomy of approximately 8 min with a payload and 12 min without a payload.

The onboard computer processing the decision-making and detection module is an NVIDIA (Santa Clara, CA, USA) Jetson Orin Nano developer kit. It is composed of an NVIDIA Ampere architecture with 1024 NVIDIA® CUDA® cores and 32 Tensor cores, a 6-core Arm Cortex-A78AE v8.2 64-bit CPU, 8GB of LPDDR5 DRAM, four USB Type A 3.2 Gen2 ports, one USB Type C, one display port and a microSD card slot for main storage.

The sensors used in this framework are a FLIR (Wilsonville, OR, USA) TAU2 thermal camera with a 13 mm lens and a Field of View (FOV) of 45° (horizontal) × 37° (vertical), and an embedded IMU from Bosch® (Gerlingen, Germany) and InvenSense® (San Jose, CA, USA) in the Pixhawk 6c. Ultimately, the framework will include RPI S1 two-dimensional LIDAR from SLAMTEC (Shanghai, China) to perform LIDAR/inertial odometry to estimate the x and y coordinates of the UAV. The current pose estimation is performed with the integrated IMU and the OptiTrack system [30]. In this framework, the thermal camera is solely used for target detection. Figure 2 shows the fully integrated UAV platform.



**Figure 2.** Fully implemented UAV frame used in this research. (a) Below view of the UAV highlighting the frame (1); thermal camera FLIR TAU2 (2); and companion computer Jetson Orin Nano (3). (b) Above view of the UAV highlighting the Optitrack tracker (4) and Holybro 915 MHz Telemetry Radio (5).

#### 4.2. Operating Systems and Middleware

Ubuntu 20.04, which is a 64-bit Linux-based operating system (OS), was installed on the companion computer. The robotic operating system (ROS) Noetic version [31] was used to communicate between the sensors, decision-making module, and detection module. To control the UAV, the open-source flight control software PX4 firmware version 1.13.2 [32] was used. The PX4 architecture can be separated into two layers: the flight stack, which is an estimation and flight control system, and the middleware, a general robotic layer. The former contains altitude and position estimators, as well as the pipelines of flight controllers. The latter can support a large variety of autonomous robots and contains the communication interfaces and the simulation layer. MAVROS [33], a ROS wrapper of the Micro Air Vehicle Link (MAVLink) protocol [34], was used for the communication between the decision-making module, which is processed on the onboard computer, and the motion module from the PX4. This communication is performed via a high-speed UART interface. Finally, communication to the ground control station (GCS) is achieved via the 915 MHz telemetry radio. The software used in the GCS is QGroundControl (version 4.0), which allows the GCS to communicate with the PX4 autopilot.

#### 4.3. Mission Objectives

The navigation task selected is a SAR mission in which the agent is required to explore and find targets in an indoor environment composed of several obstacles, a restricted flying area, no GNSS, and low visibility. Search and Rescue was selected for its challenges and the large range of possible applications. A target, with a position unknown to the UAV, was placed in the flying area. The principal objective of the UAV is to explore the environment in search of the target. To create low-visibility conditions, the framework was tested in two different settings. In simulations, the mission was executed in the presence of smoke, while real-life testing was effectuated in a low-luminosity environment. The target used in this research is a mannequin able to emit a heat signature similar to a human being. The mission comprises two phases: take-off to the required altitude and the autonomous navigation phase, which ends when the target is found. A successful mission is defined by several parameters, including staying in the restricted flying area, avoiding obstacles, and detecting the target.

#### 4.4. Decision-Making Module

In this work, the decision-making module receives input from the environment called observations and translates these observations into action commands resulting in a new agent state. The decision-making module is modelled following the POMDP representation, and more precisely implemented using the TAPIR software (commit e2c0da7) [35]. TAPIR is a C++ implementation of the POMDP online solver ABT. Several parameters such as the discount factor and the magnitude of displacement need to be selected and tuned in order for the optimal policy to be estimated. Parameter tuning will be covered in Section 5.2.2. This estimated optimal policy results in a successful mission, characterised by the agent's ability to explore its environment and find the target while avoiding obstacles and staying in the region of interest (ROI). The following current assumptions are made in this paper:

- The flight controller unit (autopilot) is assumed to include the take-off, landing, and return-home functions. The operator can regain control at all times if needed.
- LIDAR and inertial odometry for pose and motion estimation are assumed to be incorporated in the UAV.
- Observations are from real-time streaming of processed thermal imagery.
- The autonomous mission (POMDP) starts when the UAV reaches an initial waypoint.
- The mission finishes if the target is detected or if the maximum flight time is reached.
- The target and obstacles are assumed static.

The problem formulation, including the state space, set of actions, transition function, rewards and reward function, observation space, and observation model, as well as the belief states, will be covered below.

#### 4.4.1. State Space

The state space in this research is composed of  $S_a$ , the state space of the agent, and  $S_t$ , the state space of the target, and is represented by  $S = (S_a, S_t)$ . The state space of the agent  $S_a$  first includes its position  $p_a = (x_a, y_a, z_a)$  and its rotation  $o_a = \psi_a$  in the world Cartesian frame. To simplify the dynamic representation of the UAV, rotation is only in terms of the yaw angle  $\psi_a$ . The state space of the agent also includes two discrete states representing if the agent has crashed and if it is outside the ROI, respectively,  $f_c, f_r$ . Similarly, the target state comprises its position  $p_t = (x_t, y_t, z_t)$  and a discrete state  $f_t$  representing if the target has been found or not. The latter is marked true if and only if the frequency of target detection is higher than a selected threshold. In this representation,  $f_c, f_r, f_t$  are terminal states.

#### 4.4.2. Actions

Several actions were chosen for the UAV to interact with its environment as shown in Table 1. The actions were restricted to pure translation (no rotations) to simplify the UAV's dynamic.

**Table 1.** Set of actions selected in this research problem. Each action will apply a displacement  $\Delta$  from time step  $t$  to  $t + 1$ , impacting the corresponding x, y or z coordinate of the agent.

Action	$x_a(t+1)$	$y_a(t+1)$	$z_a(t+1)$
Forward	$x_a(t) + \Delta_x$	$y_a(t)$	$z_a(t)$
Backward	$x_a(t) - \Delta_x$	$y_a(t)$	$z_a(t)$
Left	$x_a(t)$	$y_a(t) + \Delta_y$	$z_a(t)$
Right	$x_a(t)$	$y_a(t) - \Delta_y$	$z_a(t)$
Up	$x_a(t)$	$y_a(t)$	$z_a(t) + \Delta_z$
Down	$x_a(t)$	$y_a(t)$	$z_a(t) - \Delta_z$
Hover	$x_a(t)$	$y_a(t)$	$z_a(t)$

#### 4.4.3. Transition Function

A dynamic model of a UAV can be described using a rotation matrix  $R_r$  of a quadrotor [36].  $R_r$ , as shown by Equation (2), was modified based on the actions as described in the previous section. Thus, the angles  $\theta_a$  and  $\phi_a$  are assumed constant, and  $\psi_a = 0^\circ$ . The POMDP is a mathematical framework used to model decision making under uncertainty; thus, action uncertainty needs to be represented. An angle deviation  $\varphi_a$  is introduced into the rotation matrix to represent action uncertainty in the Euler yaw angle. A normal distribution is used to model this uncertainty, with mean  $\mu = \psi_a = 0^\circ$  and standard deviation  $\theta = 2.0^\circ$ . This standard deviation value was selected based on the compass magnetometer heading accuracy used by the drone detailed in Section 4.1.

$$R_r = \begin{bmatrix} \cos(\varphi_a) & -\sin(\varphi_a) & 0 \\ \sin(\varphi_a) & \cos(\varphi_a) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

To model changes in position at each time step, Equation (3) was used, and can be expanded as shown in Equation (4).  $\Delta p_{r_t}$  represents the change in the agent's location from time step  $t$  to  $t + 1$ .

$$p_{a_{t+1}} = p_{a_t} + R_{r_t} \Delta p_{r_t} \quad (3)$$

$$\begin{bmatrix} x_{a_{t+1}} \\ y_{a_{t+1}} \\ z_{a_{t+1}} \end{bmatrix} = \begin{bmatrix} x_{a_t} \\ y_{a_t} \\ z_{a_t} \end{bmatrix} + \begin{bmatrix} \cos(\varphi_{a_t}) & -\sin(\varphi_{a_t}) & 0 \\ \sin(\varphi_{a_t}) & \cos(\varphi_{a_t}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_{a_t} \\ \Delta y_{a_t} \\ \Delta z_{a_t} \end{bmatrix} \quad (4)$$

#### 4.4.4. Reward Function

The reward function  $R(a, s)$  is used to calculate the reward  $r$  resulting from an action  $a$  taken by the agent at state  $s$ . The reward function is a capital element of the POMDP formulation, as it will dictate the behaviour of the agent by encouraging or punishing certain actions. The reward variables are the following:

$$R = (r_{crash}, r_{out}, r_f, r_{exp}, r_{new}, r_{alt}) \quad (5)$$

Table 2 highlights the selected values for each reward. Negative rewards are used to avoid certain actions that will put the agent in an unwanted state. Both  $r_{crash}$  and  $r_{out}$  are negative rewards. The former is used when the UAV crashes into an obstacle, and the latter when it is out of the ROI.  $r_f$  is a positive reward when the agent finds the target. To encourage the UAV to explore the environment,  $r_{exp}$  is a negative accumulative reward, punishing the agent if an action results in an already explored area, and  $r_{new}$  is a positive reward, rewarding the UAV if an action results in an unexplored area.  $r_{alt}$  is a reward encouraging the UAV to increase its altitude to facilitate the target detection module. The reward function is represented by Algorithm 1.

**Table 2.** Set of reward values used in the reward function R, defined in Algorithm 1.

Reward	Value	Description
$r_{crash}$	−110	Crashing negative reward
$r_{out}$	−110	Leaving ROI negative reward
$r_f$	200	Target found positive reward
$r_{exp}$	−5	Explored negative reward
$r_{new}$	20	Exploration reward
$r_{alt}$	−75	Altitude reward

**Algorithm 1** Reward function algorithm.

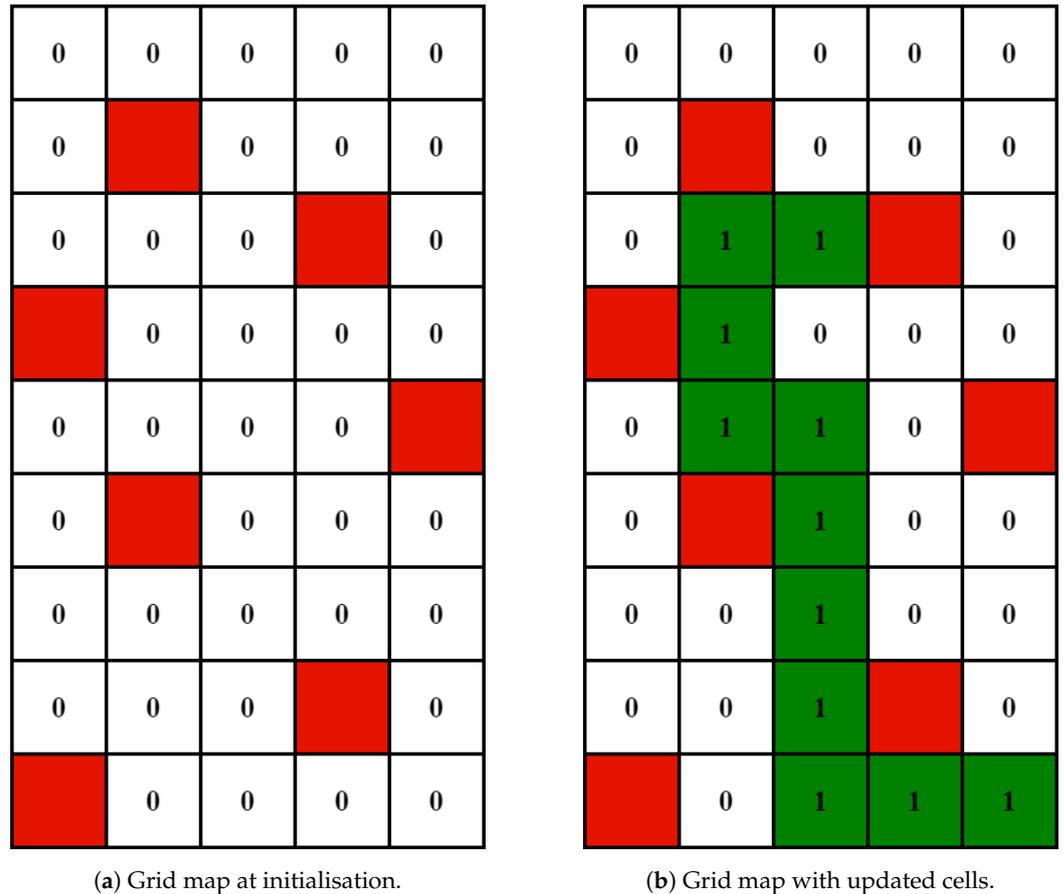
```

1:  $r \leftarrow 0$ 
2: if  $crash$  then
3:    $r \leftarrow r - r_{crash}$ 
4: end if
5: if  $out$  then
6:    $r \leftarrow r - r_{out}$ 
7: end if
8: if  $found$  then
9:    $r \leftarrow r + r_f$ 
10: end if
11: if  $explored$  then
12:    $r \leftarrow r - r_{explored} * count$ 
13: else
14:    $r \leftarrow r + r_{new}$ 
15:    $r \leftarrow r - r_{alt} * (1 - \frac{z_a - z_{min}}{z_{max} - z_{min}})$ 
16: end if
17: return  $r$ 

```

An important aspect of this research problem is to encourage the agent to explore the environment as much as possible, as the target position is unknown. To assist the agent in exploring its surroundings and avoid already explored areas, the environment is represented by a two-dimensional grid map composed of  $1 \times 1$  m cells. Cells are modelled as a structure type that encapsulates data and parameters. Each cell is composed of top right and bottom left corner coordinates, a Boolean flag marking the cell as explored or unexplored, and an integer counting the number of times the UAV has explored the cell. For a cell to be marked as “explored”, the Boolean flag is set to true if and only if the observed

position of the UAV is inside the cell. Figure 3 illustrates how the grid map is initialised and updated. The illustration represents the environment in which the system was tested in simulation and in real life. Unexplored cells are white, explored cells are highlighted in green, and obstacles are highlighted in red. Cells that can be explored also contain in their centre the number of times they have been explored. A functionality allowing the agent to keep exploring until the target was detected or until the maximum flight was reached was also introduced by keeping track of the number of cells explored and selecting an arbitrary percentage of exploration. If this percentage is attained, all the cells are marked as unexplored, and their count is reset.



**Figure 3.** Grid map representation of the testing environment at the start of the mission (a) and during flight with updated cells (b). The unexplored cells are in white, the explored cells in green, and each cell contains the number of times they have been explored. The obstacles are represented in red.

#### 4.4.5. Observation Space

The set of observations  $O$  is defined as:

$$O = o_{p_a}, o_{p_t}, o_{t_f}, o_{o_d} \quad (6)$$

where  $o_{p_a}$  is the local pose estimation of the agent,  $o_{p_t}$  is the local pose estimation of the target, and  $o_{t_f}$  is a discrete observation defining if the target has been detected or not.  $o_{p_t}$  is received only when the target is detected.  $o_{o_d}$  is a flag describing if the agent is close to an obstacle. This observation is obtained using an occupancy map object and the agent's location within this map.

#### 4.4.6. Observation Model

The observation model is a distribution function modelling the probability of observing an observation  $o$  from state  $s$  after performing action  $a$ . As the ABT solver is a generative model; it is required to model an observation  $o$  that is the result of an action  $a$  and a new state  $s'$ . The observation model includes the UAV-estimated position in the world coordinate frame  $o_{p_a}$ , as well as the location of the target  $o_{p_t}$  when it is detected by the camera. The detection of a victim using the camera's FOV was based on Sandino et al. [24,25]. The footprint area of the thermal camera is calculated using its sensor width, height, and focal length. The target location point is predicted to be within the camera's FOV if the calculated sum between the pairs of points comprising the footprint area and the target belief position is equal to  $2\pi$ . In this research, it is assumed that there is no detection uncertainty, meaning that detection is always assumed to be correct. Furthermore, this observation model is designed for the detection of stationary objects and not for moving objects.

#### 4.4.7. Belief States

Two sets of belief states are used in this work: the position of the UAV, represented by its  $(x, y, z)$  coordinates, and the position of the victim, represented by their coordinates as well. The position of the UAV is represented as a Gaussian distribution. Section 4.6 will cover in more depth how the mean  $\mu$  and standard deviation  $\sigma$  were defined. A Gaussian distribution is used to represent the  $x$  and  $y$  coordinates and  $z$  is assumed to be known with a negligible uncertainty. As previously stated in Section 4.3, the target position is unknown to the UAV. Therefore, the belief state of the target is uniformly distributed in the restricted flying area.

#### 4.4.8. Obstacle Avoidance

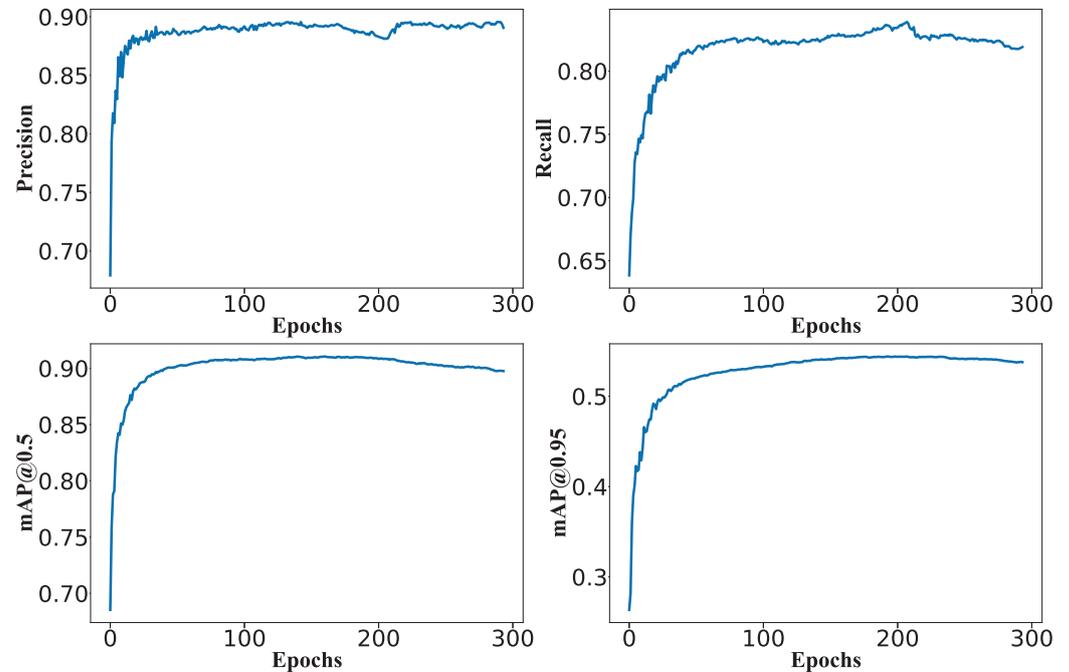
Obstacle avoidance is implemented in this framework using occupancy maps. Occupancy maps are a 3D representation of the environment using occupancy grids consisting of cells. Each cell contains binary values representing whether the cell is occupied, free, or unknown. In this work, the Octomap library [37] was used. This library is used to make occupancy maps using 3D point cloud data, making it a popular choice for robotic applications requiring the computation of maps using sensors such as LIDAR and depth cameras. For this research, occupancy maps were generated manually prior to the mission and then used by the decision-making module.

### 4.5. Detection Module

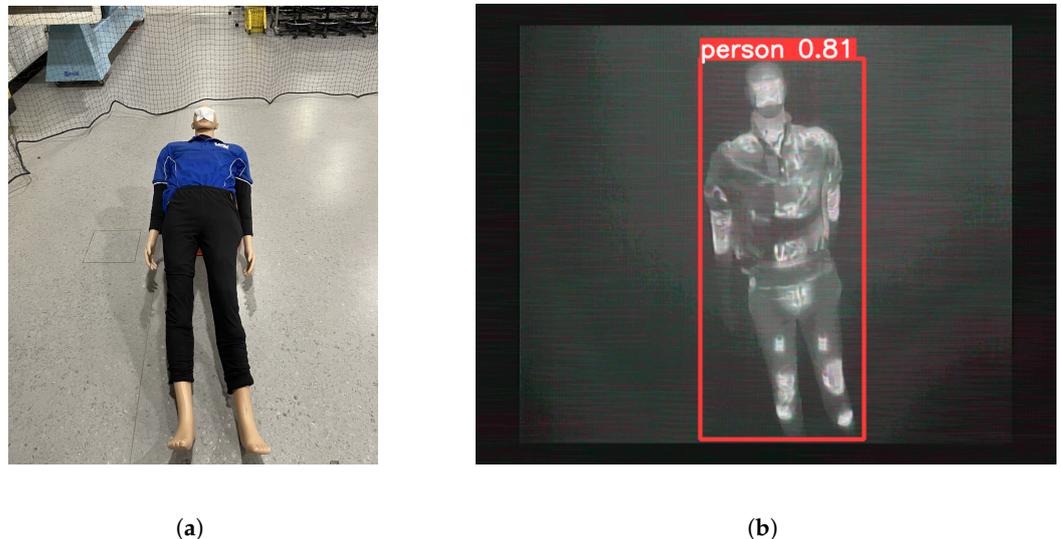
The computer vision module is composed of a deep learning network structure that processes object detection and classification from the FLIR TAU 2 thermal camera's raw frames. The deep learning model used was YOLOv5 [38], which is part of the You Only Look Once (YOLO) family of computer vision models. YOLOv5 was selected following Jiang et al.'s [15] research, which compared YOLOv3, YOLOv4, and YOLOv5 models for a UAV thermal infrared object detection framework. YOLOv5 was also selected for its simple integration with the ROS. The dataset used was developed by the Roboflow Universe Project to detect persons in infrared frames [39]. The YOLOv5 model was then trained using this dataset containing over 15,000 images. These images consist of a mix of thermal images taken from a car using the FLIR Tau2 thermal camera (Teledyne FLIR Thermal Dataset) [16] and thermal images taken by surveillance cameras. Figure 4 highlights the evaluation indicator of the YOLOv5 model during the training used in this paper. A total of 500 epochs were originally selected for training this model; however, the training was stopped at 293 epochs as the results were not improving. The YOLOv5 model achieved metric values of 89.1%, 81.9%, 89.8%, and 0.54% for precision, recall, mean average precision with an intersection over union (IoU) of 50 % (mAP\_0.5), and mean average precision with an IoU interval of 50 % to 95% (mAP\_0.95). Overall, the only metric with a low performance was mAP\_0.95, while the precision, recall, and mAP\_0.5 metrics are satisfactory. The model was tested in simulations and in real life to verify if improvements were needed before the

full integration of the system. Even with a low mAP\_0.95, the model was able to detect the heated mannequin consistently with few false positives and false negatives in simulation and real life.

Figure 5 shows the heated mannequin used in real-life experiments and the YOLOv5 detection when the UAV was flying above the target. More information about the mannequin will be given in Section 5.1.



**Figure 4.** Visual analysis of YOLOv5 evaluation indicators during training.



**Figure 5.** Target detection of the thermal mannequin using the FLIR TAU 2 thermal camera and YOLOv5. (a) Thermal mannequin side view. (b) Target detected with an 81% confidence during flight after processing the raw frame from the FLIR TAU 2 thermal camera.

#### 4.6. Localisation Module

LIDAR/inertial odometry is yet to be integrated within the current system; however, it was individually tested in low-luminosity conditions. The sensor used was the RPI S1 two-dimensional sensor, and the Hector SLAM package [40] was used to estimate the motion of the UAV. Two sets of data were processed: the ground truth and the pose estimation

from LIDAR/inertial odometry. As the pose uncertainty in the POMDP representation is modelled as a Gaussian distribution, the standard deviation of the error between these two sets of data was calculated. This value was then used in the Gaussian distribution, with the observed agent's position as the mean, and the calculated value as the standard deviation. Pose estimation was achieved using the PX4 SITL and HIL capabilities. In simulations, PX4 relies on Gazebo plugins and simulated sensors such as the GNSS and IMUs. In real life, it relies on the Optitrack system (please refer to Section 4.1 for more information) and internal sensors.

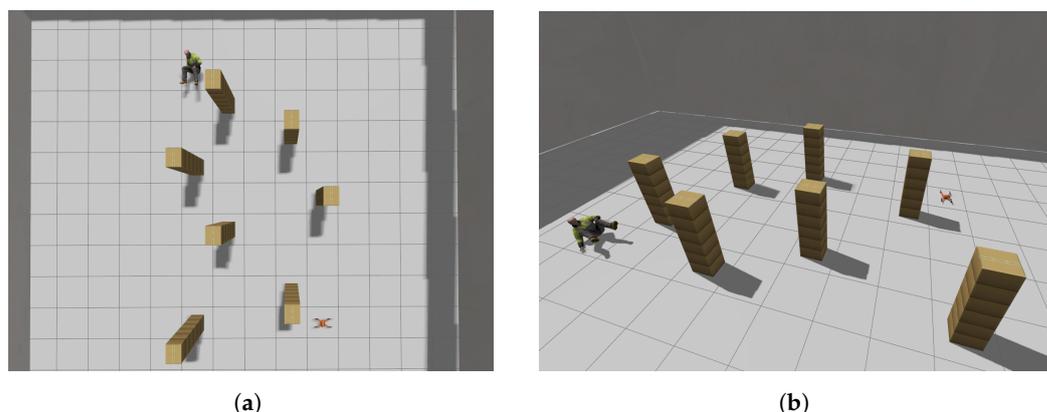
## 5. Experiments

The framework presented in Section 4 was tested in a SAR mission in an indoor environment with several obstacles in normal and low-visibility conditions. SITL simulations and real flight tests were carried out. This section covers the environment setup, as well as the parameters used in the POMDP problem formulation.

### 5.1. Environment Setup

The environment in which the UAV was operated is an indoor flying area of  $9\text{ m} \times 5\text{ m} \times 3.7\text{ m}$  (length, width, height). Seven column-shaped obstacles were placed in the search area. The target used was a mannequin able to simulate a human heat signature, and it was positioned opposite the UAV's starting point. The target was lying down in real-life testing and sitting down in SITL simulations. Both the target and obstacles were static; only the lighting was modified to create low-visibility conditions. This setup was selected to recreate realistic SAR operations, with an unknown target position, cluttered environment, lack of the GNSS, and low visibility, as much as possible. Simulations tested the framework in the presence of smoke, while real-life experiments tested it under low-luminosity conditions.

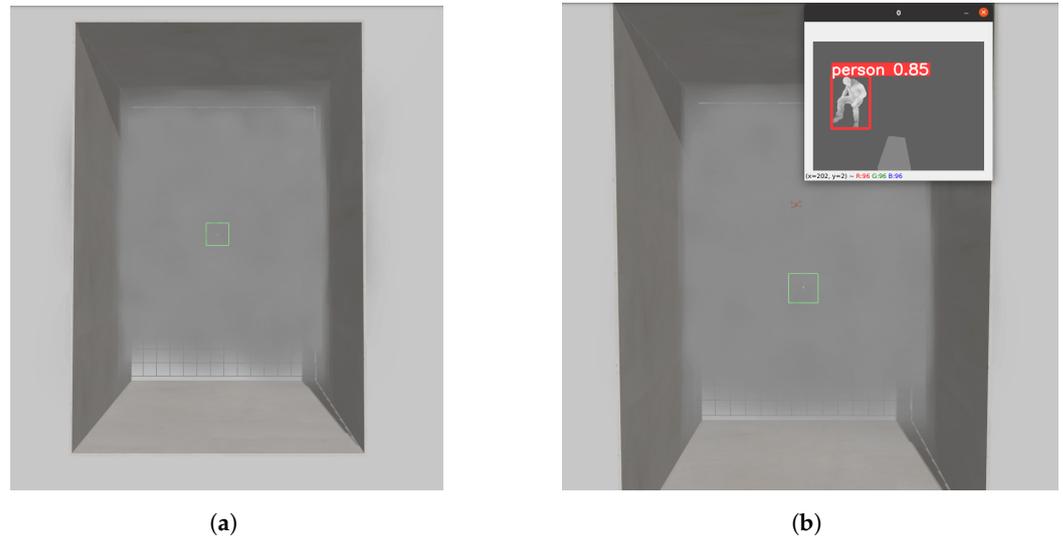
SITL experiments were carried out using the open-source flight controller software PX4 [32], Gazebo, a derivation and upgrade from Gazebo Classic, and a ROS. Gazebo is an open-source simulator for robotics, in which sensor models and high-fidelity physics are made available for users. Gazebo was chosen over Gazebo Classic for its ability to model visual obstructions in the form of particles, creating simulated smoke or fog. Simulations were carried out on a desktop station featuring an 11th Gen Intel(R) (Santa Clara, CA, USA) Core(TM) i7-11700K at 3.6GHz, 32GB DDR4 RAM, 1TB SSD, and a 24 GB NVIDIA GeForce RTX 3090. The simulation environment in Gazebo (version 5.4.0.) is represented in Figure 6 with normal visibility conditions, and Figure 7 with smoke.



**Figure 6.** Environment setup in SITL with normal visibility conditions. (a) Top view of the environment simulated in Gazebo. (b) Side view of the SITL environment.

Real-life experiments were conducted at the Queensland University of Technology (QUT) Garden Point campus, in the O block flying area (O1304), 2 George St, Brisbane City, QLD, 4000. For safety reasons, an adult mannequin with heat packs and heated clothes was used to recreate the human heat signature as shown in Figure 5. The framework was

tested under two different lighting conditions: normal lighting and obscurity, as shown in Figure 8 and Figure 9 respectively. The position of the target and obstacles were static, and no external disturbances (wind) were applied during the mission. In the current framework implementation, LIDAR/inertial odometry is not integrated. The system is also currently required to use a fixed occupancy map to know the obstacles' position. For both real-life and simulation experiments, the UAV path, target and pose estimation belief states, and occupancy map were visualised on RViz [41].



**Figure 7.** SITL environment setup with low-visibility conditions. (a) Top view of Gazebo with smoke at the start of the mission. (b) Top view of Gazebo with smoke during flight with the detection output.



**Figure 8.** Environment setup for real-life experiments at QUT O1304 with normal visibility conditions. (a) Angled view of the flying area. (b) Flying area outside the net.



**Figure 9.** Environment setup for real-life experiments at QUT O1304 with low-visibility conditions.

## 5.2. POMDP Problem Formulation

This subsection covers the uncertainty and initial belief of the POMDP formulation, as well as the parameters used in the TAPIR workspace.

### 5.2.1. Uncertainty and Initial Belief

An important aspect of this research is to create difficult conditions for a SAR operation. Therefore, the position of the target is unknown and represented by a uniform probability distribution covering the entirety of the search area. The mannequin is always located at the same position, the top left corner of the map, during simulations and real-life testing. For the uncertainty in the pose estimation of the agent, the standard deviation used was 0.1 m. This value was selected after performing LIDAR/inertial odometry experiments, as explained in Section 4.6. The maximum standard deviation found from these experiments was 0.08 m.

### 5.2.2. TAPIR Parameters

The following table highlights the parameter values selected in the POMDP model. The discount factor, responsible for how far the POMDP solver will try to plan the optimal sequence of actions, has a value of 0.93. The altitude range of the UAV is between 1.3 m and 3.5 m; actions in the x and y-axis result in a displacement of 1 m, and actions in the z-axis result in a displacement of 0.3 m. The UAV has a maximum of 135 time-steps to find the target, resulting in a maximum flight duration of approximately 5 min in simulations and real flight. Table 3 summarizes the parameters used in the POMDP model.

**Table 3.** Parameters values used in the POMDP model.

Variable	Value	Description
$\gamma$	0.93	Discount factor
$z_{min}$	1.3 m	Maximum altitude
$z_{max}$	3.5 m	Minimum altitude
$\Delta_x$	1 m	x displacement
$\Delta_y$	1 m	y displacement
$\Delta_z$	0.3 m	z displacement
$p_{a0}$	(−4.5;−2.5;2.0)	Initial agent position

Table 3. Cont.

Variable	Value	Description
$tstep_{max}$	135	Number of time steps
$\Delta t$	2 s	Time step
$v$	0.5 m/s	Velocity
$\mu_{\theta}$	1 rad	Angle uncertainty

## 6. Results

Two different setups were tested in simulation (S) and in real-life testing. Each setup was tested in normal (NV) and low-visibility (LV) conditions. The first testing configuration (M1) consists of obstacles of different heights, from 1.65 m to 3.5 m (limit of altitude), while the second setup (M2) only has obstacles of 3.5 m. The metrics used to evaluate the success of each setup consist of Success (target found), Crash (if the UAV collides with an obstacle), ROI Out (if the UAV leaves the ROI), and Timeout (target not found in the time limit). In these tests, the target position was opposite the starting position of the UAV, with coordinates of (4.5;1.5). Figure 10 shows the RViz environment of both maps, and Table 4 summarises the results:

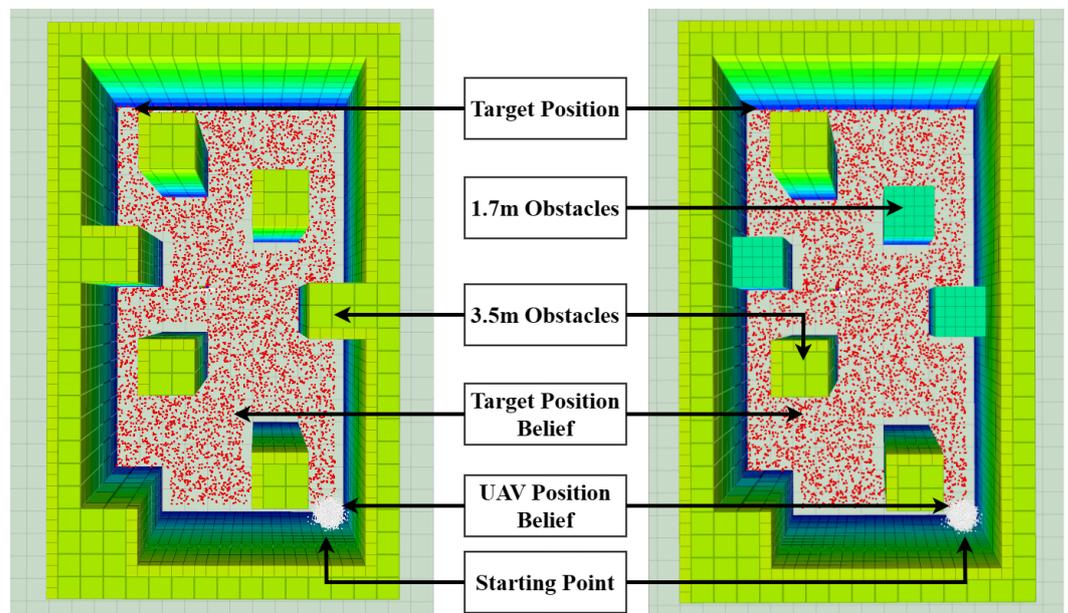


Figure 10. RViz environment for both maps at the start of the mission.

Table 4. Performance metrics for SITL simulations and real flight tests.

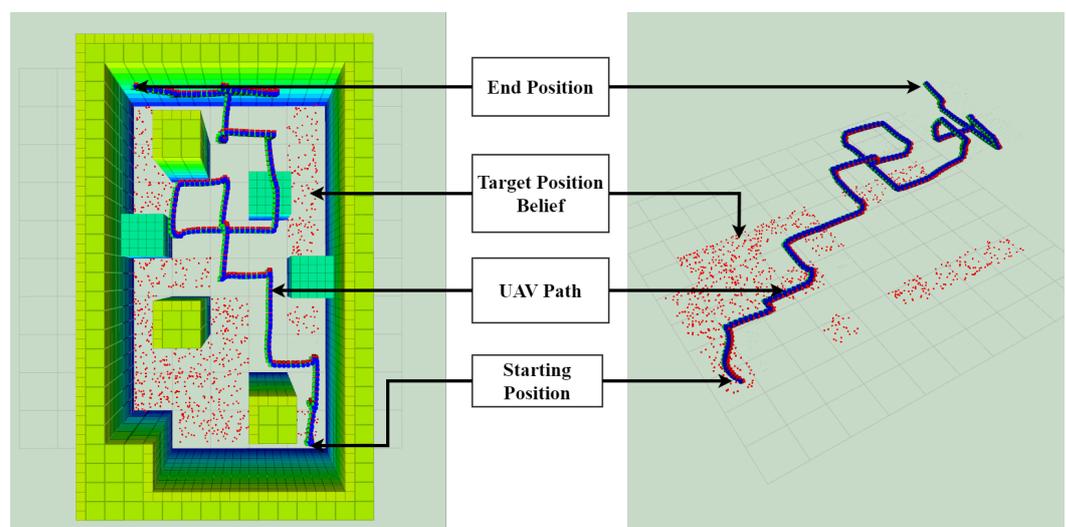
Setup	Iterations	Success Rate	Crash Rate	ROI Out	Timeout Rate
M1 NV (S)	30	100%	0%	0%	0%
M1 NV (RLT)	8	100%	0%	0%	0%
M1 LV (S)	30	100%	0%	0%	0%
M1 LV (RLT)	8	100%	0%	0%	0%
M2 NV (S)	30	100%	0%	0%	0%
M2 NV (RLT)	8	87.5%	0%	12.5%	0%
M2 LV (S)	30	100%	0%	0%	0%

Table 4. Cont.

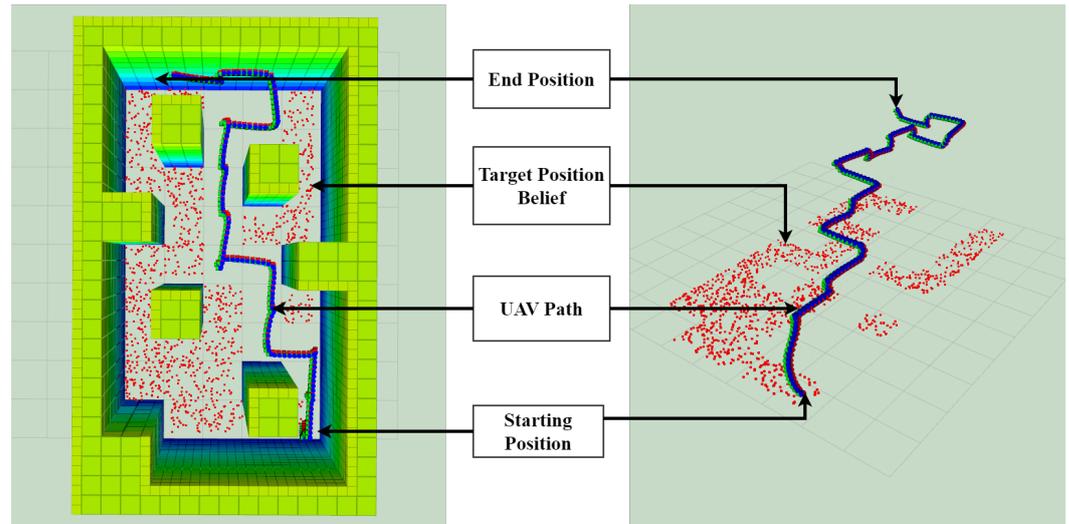
Setup	Iterations	Success Rate	Crash Rate	ROI Out	Timeout Rate
M2 LV (RLT)	8	100%	0%	0%	0%
S	120	100%	0%	0%	0%
RLT	32	96.87%	0%	3.13%	0%

From all the testing scenarios presented, the proposed framework performed perfectly in simulations, with a 100% success rate (target found with no crashes), and a 96.87% success rate in real-life testing. For the simulation, a total of 120 iterations through all the setups (30 each) were performed, and 32 iterations for real-life testing (8 each). Overall, the proposed system performed identically in simulations and in real life, with similar results. The agent found the target in both maps and in different visibility conditions, highlighting the ability of the decision-making module to explore different environments and the ability of the thermal camera and object detection module to detect the target in normal and low-visibility conditions. The differences between the heat signature of a human being and the heated mannequin used in real-life testing did not impact the operation of the agent. On the contrary, the mannequin was placed in a seated position in simulations to facilitate detection, as the laying-down model was consistently miss-detected. These miss-detections were caused by differences between the target used in simulation and the actual detection model. The YOLOv5 model was trained using actual people and was not trained to detect the simulated mannequin.

The most frequent trajectory for M1 is presented in Figure 11, and the most frequent trajectory for M2 is shown in Figure 12. In all tested environments, the first series of actions often resulted in the following steps: meeting the altitude requirement to avoid the negative reward and going to the centre of the map. The former was part of the POMDP formulation, with  $r_{alt}$  encouraging the UAV to reach the required altitude. The latter, on the other hand, was the result of the complete formulation and reward function. The POMDP always outputs a policy in which the first actions result in the agent's position being in the centre of the map to maximise the number of unexplored cells around the UAV. When the centre is reached, the agent will either keep going toward the top side of the map, as shown in Figure 11 and Figure 12, or the actions will result in the agent exploring the bottom side of the map.

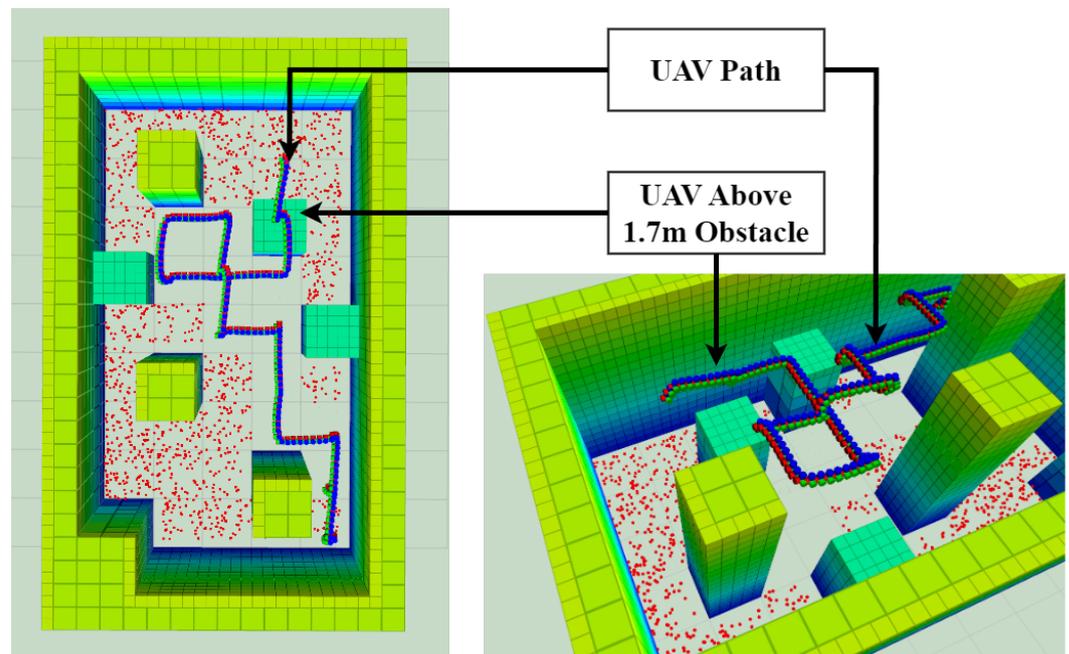


**Figure 11.** Most frequent trajectory of the agent for Map 1 (M1) in normal and low-visibility conditions with the target position belief as red particles. Top view and side view of the trajectory in RViz with higher obstacles and walls in light green, and lower obstacles in medium spring green.



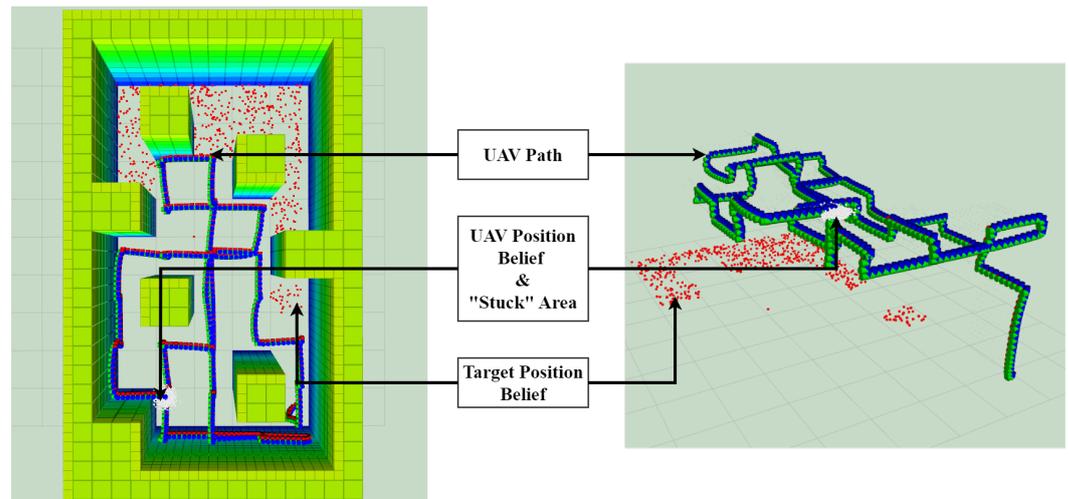
**Figure 12.** Most frequent trajectory of the agent for Map 2 (M2) in normal and low-visibility conditions with the target position belief as red particles. Top view and side view of the trajectory in RViz with higher obstacles and walls in light green, and lower obstacles in medium spring green.

M1 was also used to highlight the ability of the agent to go above an obstacle, as shown in Figure 13. The POMDP solver was able to recognise the lower obstacles as a “non-threat” and output actions resulting in the agent’s position being above these smaller obstacles, without causing any collisions and successfully avoiding the obstacle.



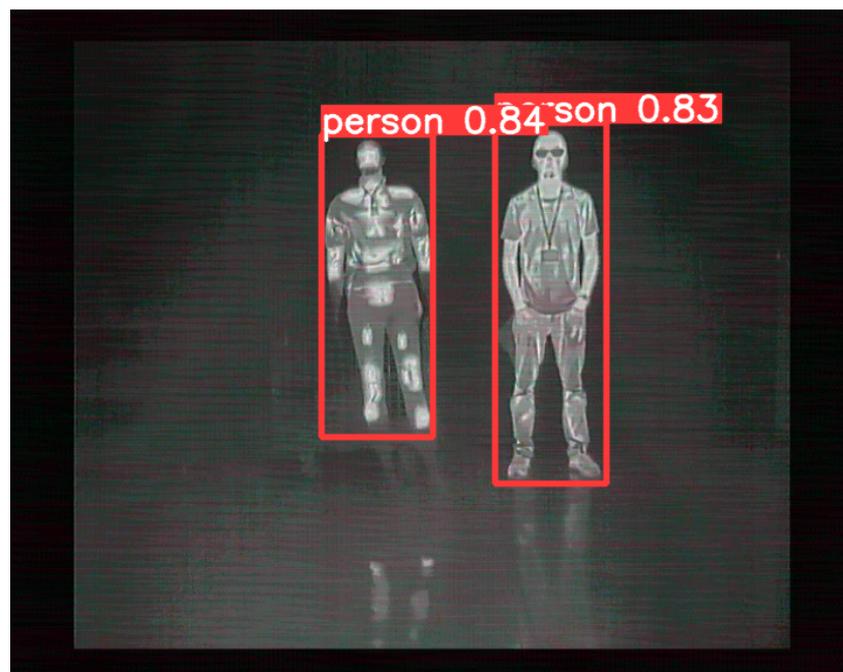
**Figure 13.** Agent can fly over the smaller obstacles in medium spring green with the target position belief as red particles. Top view and side view of the trajectory going over the obstacle in RViz with higher obstacles and walls in light green.

The “Out of ROI” state that occurred in M2 NV in real-life testing was caused by a known issue of the framework, which put the agent in an area far from other unexplored cells, causing it to repeat UP and DOWN actions, forcing the agent to go higher than the altitude limit. This is represented in Figure 14. This issue was limited by the functionality of keeping track of the number of cells explored and resetting the explored cells when a percentage of exploration is attained, as explained in Section 4.4.4.

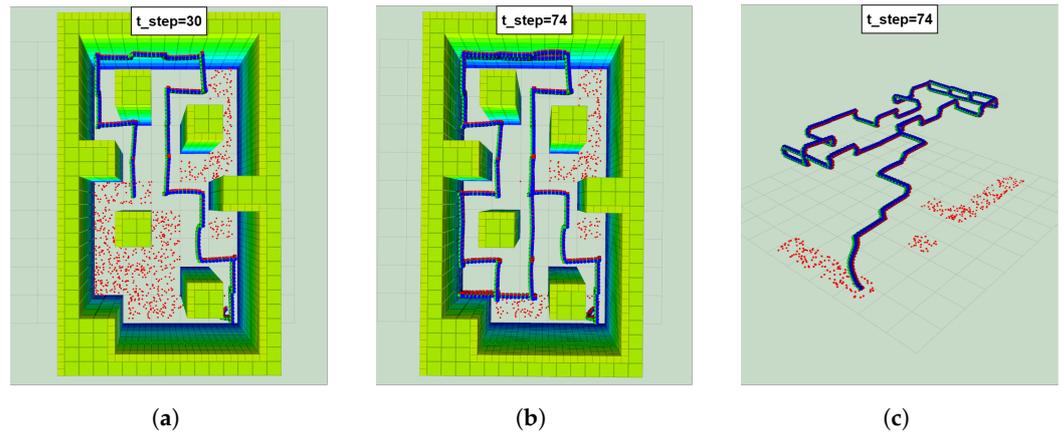


**Figure 14.** Representation of the POMDP solver getting stuck in the bottom part of the map. As the agent is too far from other unexplored cells, the POMDP is not able to compute practical actions and repeats UP and DOWN commands. The obstacles and walls are in green; the target position belief states are red particles. Top view and side view of the path on RViz.

The detection model was trained using images from an external dataset detecting human beings. A mannequin was used in the experiments for safety purposes, hence creating differences between the model and the target. Figure 15 highlights the differences between the heat signature of the mannequin and a person. These differences caused a few miss-detections, which are highlighted in Figure 16. These miss-detections forced the agent to keep exploring the environment, often activating the functionality and resetting the states of the cells. In the few times this miss-detection happened, the agent was always able to explore a larger part of the map and come back to the target area for successful detection. These miss-detections were also caused by a loss of heat in the heat-packs and heated clothes after a long testing session, creating a noticeable difference between the mannequin and an actual human being.

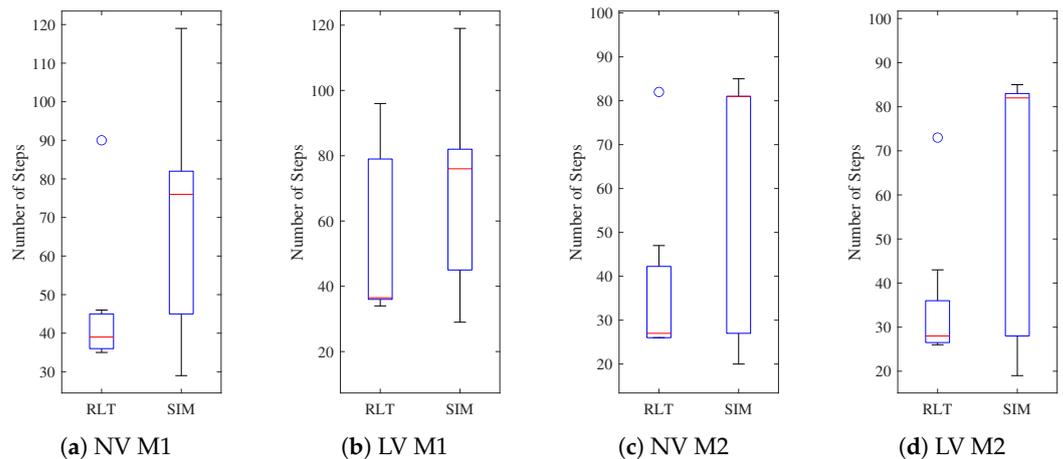


**Figure 15.** YOLOv5 detection output of thermal imagery with a heated mannequin and a human being.



**Figure 16.** Representation of a miss-detection in Map 2 (M2), forcing the agent to explore the environment and using the functionality to reset the states of the cells. Obstacles and walls are shown in light green. The target position belief is shown as red particles. (a) Path of the agent flying over the target without detecting it. (b) Final path of the agent with a top view on RViz after the detection. (c) Final path of the agent with a side view on RViz without the occupancy map.

To assess the performance of the proposed framework, an analysis of the number of time steps taken by the agent to detect the target was conducted. Box plots were used to summarise the results of this analysis as shown in Figure 17.



**Figure 17.** RLT vs SIM for Map 1 (M1) and Map 2 (M2) in normal (NV) and low-visibility (LV) conditions. Blue boxes represent the interquartile range, red lines the median, black lines the whiskers, and blue circles outliers.

The difference between the lengths of the whiskers of the simulation and real-life results was caused by how the UAV motion in Gazebo is modelled. The equations modelling the motion in Gazebo approximate the behaviour of a generic quadcopter rather than the exact frame used in real-life testing, which in this case, was the Holybro X500 V2.

The main variance between each map is the height of the obstacles. The first map, with obstacles of 1.7 m and 3.5 m, offers the agent a larger range of choices to explore the environment as it has the possibility to fly over some obstacles. On the other hand, the second map, with obstacles of 3.5 m only, restricts the freedom of the agent. This is highlighted in Figure 17, with the interquartile range of 90 steps for both M1 in NV M1 and LV M1, compared to 65 and 66 steps for NV M2 and LV M2, respectively, in simulations. The problem formulation did not include any change in behaviour between normal and low-visibility conditions. This is represented in the median for each boxplot, with a median of 76 for both NV M1 and LV M1 in simulations, and a median of 39 and 36.5 steps in real-life testing (Figure 17a,b). For NV M2 and LV M2, a median of 81 steps and 82 steps, respectively, in simulations, and 27 and 28 steps for real-life testing (Figure 17c,d) was

determined. The mean altitude in M1 for simulations was 2.97 m and 2.91 m in real-life testing. For M2, the mean altitude was 2.98 m in simulations and 2.87 m in real-life testing.

## 7. Discussion

The framework proposed in this research paper offers a viable and interesting solution for autonomous UAV navigation and decision making in GNSS-denied and visually degraded environments. This work is a continuation of [42], which itself was an extension of the contributions from Vanegas and Gonzalez [23] and Sandino et al. [24]. In [42], a thermal camera was added to the framework to improve target detection under low-visibility conditions, some modifications to the problem formulation and TAPIR parameters were made to improve the performance (see below), and real-life experiments were performed to verify the simulation results.

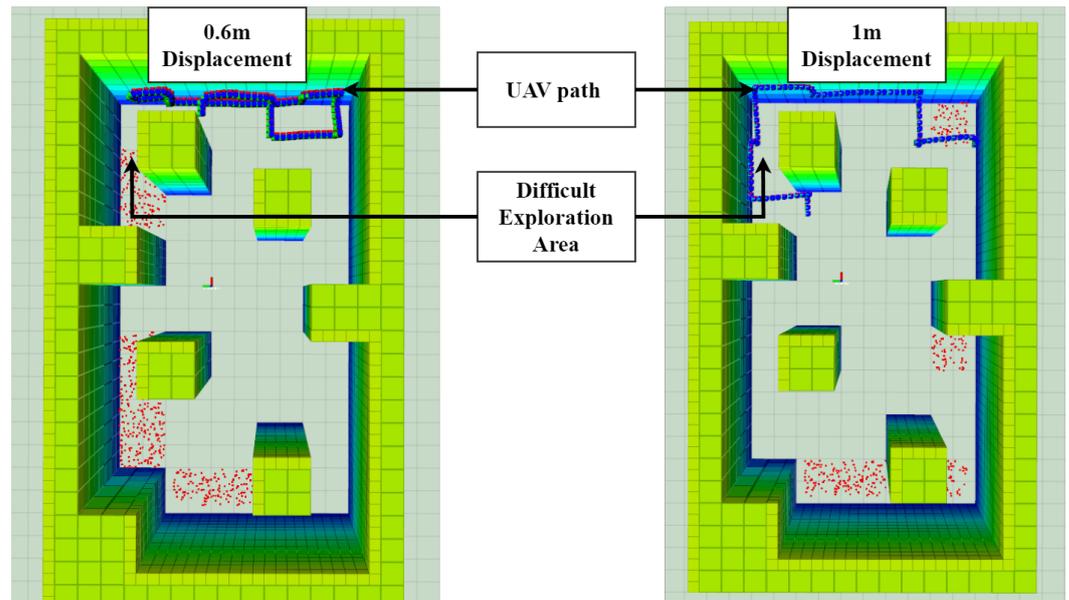
The first version of the framework was modelled with a velocity of 0.3 m/s and a time step of 2 s, resulting in a displacement of 0.6m for the forward, backward, left, and right actions. This displacement was calculated by the transition function, modelling the transition from one state to another, which in this case, was the drone's position state. This transition function is represented by the following equations:

$$\Delta x = (v * \cos(\theta + \mu_\theta) - v * \sin(\theta + \mu_\theta)) * \Delta t * 0.05 \quad (7)$$

$$\Delta y = (v * \sin(\theta + \mu_\theta) + v * \cos(\theta + \mu_\theta)) * \Delta t * 0.05 \quad (8)$$

This displacement of 0.6 m was later found to cause issues in more crowded environments, such as the map tested in the experiments. More precisely, a displacement of 0.6 m was not fully compatible with the environment representation of a 1 m × 1 m cell and 1 m × 1 m obstacles. For example, the cells located in the top left corner were difficult for the UAV to access. The obstacle created a single possible path, forcing the UAV along the wall and around the obstacle. However, a displacement of 0.6 m did not allow the agent to follow this path, as shown in Figure 18. This misrepresentation of the displacement magnitude and the environment highlights the proper functioning of the POMDP. Exploration at the cost of collisions or exiting the ROI is not a possible output of the solver. However, the lack of compatibility between these two components produced unexplored areas, and often caused the UAV to be stuck in a series of opposite actions (UP, DOWN, UP, DOWN). To fix this problem, the x and y displacement was increased to 1 m per action. As the agent starting point is at the centre of a cell, each action would move the UAV from the centre of one cell to another. The uncertainty in action, represented by an uncertainty angle of one radian, would still result in an approximate position at the centre of the cells. To attain this displacement, the velocity was selected to be 0.5 m/s, with a time step of 2 s. These parameters allowed the UAV to explore most of the map while avoiding obstacles and staying inside the ROI, as shown in Figure 18.

Overall, this study offers a flexible framework for autonomous UAV decision making in low-visibility environments with no GNSS. The framework is scalable, flexible, and portable depending on the selected sensors, mission goals, and algorithms used in the object detection and decision-making modules. The POMDP formulation was designed to work with other multi-rotor UAVs of larger or smaller sizes, thanks to the transition function modelling the dynamics of a UAV. The transition function was built based on a quadcopter dynamic model, but can easily be modified to model the dynamics of a hexacopter, for example. The reward function, state space, and other parameters of the problem formulation are not specific to the drone. Actions are modelled only for multi-rotor UAVs, as they can hover and move to waypoints. Larger environments can also be used to test this problem formulation, and will not impact the performance of the POMDP solvers or other algorithms. Thermal cameras also offer a robust solution for human detection in degraded visibility environments.



**Figure 18.** Top view of the agent’s trajectory for Map 2 (M2) with the target position belief as red particles. A 0.6 m displacement does not allow the agent to explore some areas, while a 1 m displacement allows the agent to explore crowded areas.

In terms of limitations, the current framework is dependent on the occupancy map provided before the mission and is not updated during the flight, therefore assuming a static environment. The aspect of the environment that can be assumed to be stochastic is the visibility conditions, as thermal cameras are efficient in both normal and low-visibility conditions for heat signature detection. Not updating the map during the mission forces the pose estimation system to be highly reliable, as a high uncertainty for position estimation will have higher chances of causing collisions. Furthermore, the representation of the environment as a 2D grid map with  $1 \times 1$  m cells limits the obstacles to a “square” shape or to “danger areas” with variable heights. Another limitation of this framework is the restriction to translation motion, with no rotations. This was selected to facilitate the dynamics and motion of the UAV, as well as to ease the integration between the possible actions and the environment representation. It is however possible to include heading actions in the problem formulation and to ignore the left, right, and backward commands.

A successful implementation of this framework for real-life applications in SAR operations requires the examination of several challenges, including but not limited to:

- The UAV flight autonomy, which will restrain the mission duration.
- The implementation of the localisation module using a sensor able to estimate the pose estimation of the agent with a low uncertainty.
- A localisation module that is efficient enough to perform even in low-visibility conditions, including smoke or low light.
- Hazards which might damage the UAV, including but not limited to chemical, electrical, and poor weather conditions.
- Proper handling of LiPo batteries to avoid fire hazards from possible impacts and sources of ignition.
- The size of the UAV which will be able to handle a heavy payload constituting several sensors.

## 8. Conclusions and Future Work

This paper presented a problem formulation and a framework for autonomous UAV navigation and target detection in GNSS-denied and visually degraded environments. The problem formulation was developed as a Partially Observable Markov Decision Process, and the system performed a mission modelled as a Search and Rescue operation. The mis-

sion consisted of exploring an environment composed of obstacles and visual obstructions to find a victim in an unknown position. The POMDP allows for the modelling of the target and agent's position uncertainty as belief states using probability distributions. Simulation and real-life experiments were used to assess the performance of the framework. The ABT solver used in this paper allowed the system to first compute an offline policy a few seconds before the start of the mission, as well as improve the previously computed policy during the flight to output the most optimal path. The system proposed in this research allowed a small UAV to autonomously explore and navigate an environment composed of obstacles under low-visibility conditions to successfully find and detect a human being with an unknown position using target detection based on thermal imagery.

The primary contributions of this paper are:

1. A UAV framework for autonomous navigation and target detection in GNSS-denied environments with low visibility and obstacles. The problem is formulated as a POMDP and has possible applications in Search and Rescue operations.
2. The framework was implemented, integrated, and tested in real life to verify the simulation results. The decision-making and object detection modules run on the on-board computer and are considered as resource-limited hardware. The sensors, onboard computer, and other hardware are safely mounted on a small UAV, considered as a weight-constrained platform.
3. Integration of a deep learning detection model using a thermal camera with a POMDP framework onboard a UAV.

This paper offers a novel POMDP formulation for SAR operations in GNSS-denied, visually degraded, and cluttered environments. As highlighted in the literature review in Section 1, efficient solutions to explore and navigate challenging environments using UAVs already exist. However, these solutions either do not model decision making under uncertainty or have not been tested in low-visibility environments. This research is innovative by using a POMDP to allow a drone to autonomously explore a cluttered environment in different visibility conditions without a GNSS with state and action uncertainty.

Future work includes the implementation of the localisation module using sensors such as LIDAR or depth cameras to fully rely on the hardware present onboard the UAV without the use of the GNSS. It also includes researching the use of unknown maps and/or dynamic environments and modifying the problem formulation for this new challenge. A study comparing the performance of the current framework using the ABT with that of other POMDP solvers would also be highly beneficial. Moreover, training and testing newer object detection systems like YOLOv8 could be advantageous, as the framework allows for other tools to be implemented. Integrating the results of the objection model in the observation model of the POMDP could enhance the robustness of the POMDP in real scenarios.

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

**Funding:** The authors would like to acknowledge The Australian Research Council (ARC) through the ARC Discovery Project 2020 “When every second counts: Multi-drone navigation in GPS-denied environments” (GA64830).

**Data Availability Statement:** The collected data supporting this article's research findings are available at <https://1drv.ms/u/s!AmUEDov2Mv7ztV3M9V2vavsh-AjU> (accessed on 22 November 2023).

**Acknowledgments:** The authors acknowledge the continued support from Queensland University of Technology (QUT) through the QUT Centre for Robotics (QCR) and Engineering Faculty for allowing access and flight tests at O block, QUT. The authors would also like to thank Dennis Brar for his help with the LIDAR/Inertial odometry testing in low-visibility conditions.

**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 Tree
FCU	Flight Controller Unit
FOV	Field of View
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HIL	Hardware in the Loop
IoU	Intersection over Union
IR	Infrared
LIDAR	Light Detection and Ranging
M1 LV	Map One, Low Visibility
M1 NV	Map One, Normal Visibility
M2 LV	Map Two, Low Visibility
M2 NV	Map Two, Normal Visibility
mAP	Mean Average Precision
MDP	Markov Decision Process
OS	Operating System
POMDP	Partially Observable Markov Decision Process
RGB	Red, Green, Blue
RLT	Real-Life Testing
ROI	Region of Interest
ROS	Robot Operating System
SAR	Search and Rescue
SITL	Software in the Loop
SLAM	Simultaneous Localisation and Mapping

### References

1. WMO. Weather-Related Disasters Increase over Past 50 Years, Causing More Damage but Fewer Deaths. Available online: <https://public.wmo.int/en/media/press-release/weather-related-disasters-increase-over-past-50-years-causing-more-damage-fewer> (accessed on 20 March 2023).
2. Jones, R.L.; Guha-Sapir, D.; Tubeuf, S. Human and economic impacts of natural disasters: Can we trust the global data? *Sci. Data* **2022**, *9*, 572. [CrossRef] [PubMed]
3. Kwak, J.; Park, J.H.; Sung, Y. Emerging ICT UAV applications and services: Design of surveillance UAVs. *Int. J. Commun. Syst.* **2021**, *34*, e4023. [CrossRef]
4. Zimroz, P.; Trybala, P.; Wroblewski, A.; Goralczyk, M.; Szrek, J.; Wojcik, A.; Zimroz, R. Application of UAV in search and rescue actions in underground mine a specific sound detection in noisy acoustic signal. *Energies* **2021**, *14*, 3725. [CrossRef]
5. Dawei, Z.; Lizhuang, Q.; Demin, Z.; Baohui, Z.; Lianglin, G. Unmanned aerial Vehicle (UaV) Photogrammetry Technology for Dynamic Mining Subsidence Monitoring and Parameter Inversion: A Case Study in China. *IEEE Access* **2020**, *8*, 16372–16386. [CrossRef]
6. UAV Challenge. Search and Rescue. Available online: <https://uavchallenge.org/search-and-rescue/> (accessed on 15 January 2024).
7. UAV Challenge. DARPA Subterranean (SubT) Challenge. Available online: <https://www.darpa.mil/program/darpa-subterranean-challenge> (accessed on 15 January 2024).
8. Department of Agriculture and Fisheries. SharkSmart Drone Trial. Available online: <https://www.daf.qld.gov.au/business-priorities/fisheries/shark-control-program/shark-control-equipment/drone-trial> (accessed on 15 January 2024).

9. Tsiourva, M.; Papachristos, C. Multi-modal Visual-Thermal Saliency-based Object Detection in Visually-degraded Environments. In Proceedings of the 2020 IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2020; pp. 1–9. [[CrossRef](#)]
10. Dang, T.; Mascarich, F.; Khattak, S.; Papachristos, C.; Alexis, K. Graph-based Path Planning for Autonomous Robotic Exploration in Subterranean Environments. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 3105–3112. [[CrossRef](#)]
11. Papachristos, C.; Khattak, S.; Alexis, K. Autonomous exploration of visually-degraded environments using aerial robots. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 775–780. [[CrossRef](#)]
12. Khattak, S.; Nguyen, H.; Mascarich, F.; Dang, T.; Alexis, K. Complementary Multi-Modal Sensor Fusion for Resilient Robot Pose Estimation in Subterranean Environments. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 1024–1029. [[CrossRef](#)]
13. Zhang, J.; Singh, S. LOAM : Lidar Odometry and Mapping in real-time. In Proceedings of the Robotics: Science and Systems Conference (RSS), Rome, Italy, 13–15 July 2014; pp. 109–111.
14. Sizintsev, M.; Rajvanshi, A.; Chiu, H.P.; Kaighn, K.; Samarasekera, S.; Snyder, D.P. Multi-Sensor Fusion for Motion Estimation in Visually-Degraded Environments. In Proceedings of the 2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Würzburg, Germany, 2–4 September 2019; pp. 7–14. [[CrossRef](#)]
15. Jiang, C.; Ren, H.; Ye, X.; Zhu, J.; Zeng, H.; Nan, Y.; Sun, M.; Ren, X.; Huo, H. Object detection from UAV thermal infrared images and videos using YOLO models. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102912. [[CrossRef](#)]
16. FLIR. Free Teledyne FLIR Thermal Dataset for Algorithm Training. Available online: <https://www.flir.com/oem/adas/adas-dataset-form/> (accessed on 15 February 2023).
17. Dong, J.; Ota, K.; Dong, M. Real-Time Survivor Detection in UAV Thermal Imagery Based on Deep Learning. In Proceedings of the 2020 16th International Conference on Mobility, Sensing and Networking (MSN), Tokyo, Japan, 17–19 December 2020; pp. 352–359. [[CrossRef](#)]
18. Lagman, J.K.D.; Evangelista, A.B.; Paglinawan, C.C. Unmanned Aerial Vehicle with Human Detection and People Counter Using YOLO v5 and Thermal Camera for Search Operations. In Proceedings of the 2022 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Shah Alam, Malaysia, 25 June 2022; pp. 113–118. [[CrossRef](#)]
19. Kochenderfer, M.J. *Decision Making under Uncertainty: Theory and Application*; Lincoln Laboratory Series; MIT Press: Cambridge, MA, USA, 2015.
20. 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. [[CrossRef](#)]
21. Galvez-Serna, J.; Vanegas, F.; Gonzalez, F.; Flannery, D. Towards a Probabilistic Based Autonomous UAV Mission Planning for Planetary Exploration. In Proceedings of the 2021 IEEE Aerospace Conference (50100), Big Sky, MT, USA, 6–13 March 2021; pp. 1–8. [[CrossRef](#)]
22. Eaton, C.M.; Chong, E.K.; Maciejewski, A.A. Robust UAV path planning using POMDP with limited FOV sensor. In Proceedings of the 2017 IEEE Conference on Control Technology and Applications (CCTA), Maui, HI, USA, 27–30 August 2017; pp. 1530–1535. [[CrossRef](#)]
23. Vanegas, F.; Gonzalez, F. Enabling UAV navigation with sensor and environmental uncertainty in cluttered and GPS-denied environments. *Sensors* **2016**, *16*, 666. [[CrossRef](#)] [[PubMed](#)]
24. 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. [[CrossRef](#)]
25. Sandino, J.; Caccetta, P.A.; Sanderson, C.; Maire, F.; Gonzalez, F. Reducing Object Detection Uncertainty from RGB and Thermal Data for UAV Outdoor Surveillance. In Proceedings of the 2022 IEEE Aerospace Conference (AERO), Big Sky, MT, USA, 5–12 March 2022; Volume 2022.
26. Kurniawati, H.; Yadav, V. *An Online POMDP Solver for Uncertainty Planning in Dynamic Environment*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 611–629. [[CrossRef](#)]
27. Russell, S.J.S.J. *Artificial Intelligence: A Modern Approach*, 4th ed.; Pearson: London, UK, 2021.
28. Kaelbling, L.P.; Littman, M.L.; Cassandra, A.R. Planning and acting in partially observable stochastic domains. *Artif. Intell.* **1998**, *101*, 99–134. [[CrossRef](#)]
29. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; The MIT Press: Cambridge, MA, USA, 2006.
30. Optitrack. OptiTrack for Robotics. Available online: <https://optitrack.com/applications/robotics/> (accessed on 5 May 2023).
31. OSR Foundation. Robot Operating System. Available online: <https://www.ros.org/> (accessed on 5 May 2023).
32. Meier, L.; Honegger, D.; Pollefeys, M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 6235–6240. [[CrossRef](#)]
33. Ermakov, V. Mavros. Available online: <http://wiki.ros.org/mavros> (accessed on 5 May 2023).
34. Koubâa, 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. [[CrossRef](#)]
35. Klimenko, D.; Song, J.; Kurniawati, H. *TAPIR: A Software Toolkit for Approximating and Adapting POMDP Solutions Online*; Australian National University: Canberra, Australia, 2014.

36. Chovancová, A.; Fico, T.; Chovanec, L.; Hubinsk, P. Mathematical Modelling and Parameter Identification of Quadrotor (A Survey). *Procedia Eng.* **2014**, *96*, 172–181. [[CrossRef](#)]
37. Hornung, A.; Wurm, K.; Bennewitz, M.; Stachniss, C.; Burgard, W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robot.* **2013**, *34*, 189–206. [[CrossRef](#)]
38. Ultralytics. YOLOv5. Available online: <https://github.com/ultralytics/yolov5> (accessed on 15 May 2023).
39. Roboflow Universe Projects. People Detection—Thermal Dataset. 2022. Available online: <https://universe.roboflow.com/roboflow-universe-projects/people-detection-thermal> (accessed on 3 February 2023).
40. Kohlbrecher, S.; von Stryk, O.; Meyer, J.; Klingauf, U. A flexible and scalable SLAM system with full 3D motion estimation. In Proceedings of the 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 1–5 November 2011; pp. 155–160. [[CrossRef](#)]
41. Faust, J.; Hershberger, D.; Gossow, D.; Woodall, W.; Haschke, R. Rviz. Available online: <https://github.com/ros-visualization/rviz> (accessed on 21 November 2023).
42. Boiteau, S.; Vanegas, F.; Sandino, J.; Gonzalez, F.; Galvez-Serna, J. Autonomous UAV Navigation for Target Detection in Visually Degraded and GPS Denied Environments. In Proceedings of the 2023 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2023; pp. 1–10. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.