




Research Article

Minimum Time Search in Real-World Scenarios Using Multiple UAVs with Onboard Orientable Cameras

Sara Pérez-Carabaza ¹, Eva Besada-Portas ¹,
José Antonio Lopez-Orozco,¹ and Gonzalo Pajares ²

¹Arquitectura de Computadores y Automática, Universidad Complutense de Madrid, Spain

²Ingeniería del Software e Inteligencia Artificial, Universidad Complutense de Madrid, Spain

Correspondence should be addressed to Sara Pérez-Carabaza; sapere04@ucm.es

Received 14 November 2018; Accepted 21 March 2019; Published 12 May 2019

Academic Editor: Grigore Stamatescu

Copyright © 2019 Sara Pérez-Carabaza et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a new evolutionary planner to determine the trajectories of several Unmanned Aerial Vehicles (UAVs) and the scan direction of their cameras for minimizing the expected detection time of a nondeterministically moving target of uncertain initial location. To achieve this, the planner can reorient the UAVs cameras and modify the UAVs heading, speed, and height with the purpose of making the UAV reach and the camera observe faster the areas with high probability of target presence. Besides, the planner uses a digital elevation model of the search region to capture its influence on the camera likelihood (changing the footprint dimensions and the probability of detection) and to help the operator to construct the initial belief of target presence and target motion model. The planner also lets the operator include intelligence information in the initial target belief and motion model, in order to let him/her model real-world scenarios systematically. All these characteristics let the planner adapt the UAV trajectories and sensor poses to the requirements of minimum time search operations over real-world scenarios, as the results of the paper, obtained over 3 scenarios built with the modeling aid-tools of the planner, show.

1. Introduction

The strong research interest in UAVs trajectory planning takes advantage of the UAVs capabilities to perform different types of military and civil missions, such as georeference [1], wildlife monitoring [2], or target tracking [3]. Moreover, the current developments of their onboard sensorial systems make them also ideal for performing risky long-endurance reconnaissance and surveillance operations. This work focuses on a type of Probabilistic Target Search Problem (PTSP), named minimum time search (MTS) [4], since reducing the time required by the UAVs and their onboard sensors to detect the target is a critical objective of the mission. The selected problem has several applications that include looking for survivors after natural disasters (e.g., after fires or earthquakes), search and rescue operations, or searching for military targets.

Approaches that tackle PTSPs determine the trajectories of the UAVs in spite of the uncertainty associated with

the target location and sensor capabilities. To do this, they probabilistically model the different uncertainty sources. In more detail, the information about the target position is modeled with an initial probability map (with the belief of target presence within the search area) and a probabilistic motion model, while the sensors uncertainty is modeled with detection likelihood functions. As an example, Figure 1 schematizes a PTSP where two UAVs equipped with electrooptic sensors (cameras) look for a static lost off-road vehicle in the mountains. The colored map at the bottom shows the altitude of the search area (with green being associated with valleys and brown with mountains), the grey shadowed map in the middle displays the target initial belief (darker areas, associated with lower altitudes, indicate a higher probability of target presence), and the blue/red polygons, respectively, show the current observed areas (within the cameras footprints) by the onboard sensor of each UAV. When the PTSP is also a MTS operation, the regions with higher target probabilities (darker grey areas)

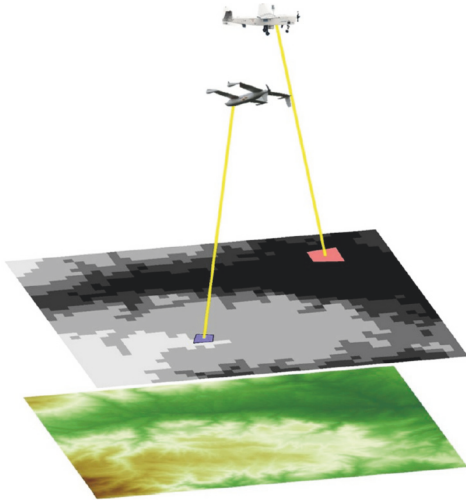


FIGURE 1: Schematic representation of the problem.

should be observed (falling within the cameras footprints) as soon as possible in order to minimize the target detection time.

One of the main goals of MTS planners is to reduce the target detection time, which can be achieved by optimizing the expected time of target detection [5–10]. Other PTSP approaches optimize alternative criteria, such as maximizing the probability of target detection [11–14] or minimizing its counterpart probability of nondetection [15, 16], maximizing the information gain [17], minimizing the system entropy [18], minimizing its uncertainty (areas with intermediate belief of target presence) [19], or optimizing normalized or discounted versions of the previous criteria [4, 20–22]. A common characteristic of the different approaches is that, in scenarios with bounded resources (e.g., limited flying time or fuel), they often obtain better results than predefined search patterns (e.g., spiral, lawnmower), as they adapt the UAV trajectories to the scenario specific target initial belief and motion [6, 20]. Besides, although the approaches that optimize the previous PTSP criteria can share the same elements and probabilistic models, MTS distinctiveness is the extreme influence of the visiting order of the high probability regions in the expected time of target detection, as prioritizing flying over high probability areas first increases the chances of finding the target earlier [4].

The NP-hard complexity of PTSP [23] is tackled with suboptimal algorithms and heuristics, such as gradient-based approaches [13, 15–17, 19], greedy methods [8, 12, 20], cross-entropy optimization [4, 7], Bayesian optimization algorithms [5], ant colony optimization [6, 9], or genetic algorithms [10]. Besides, streamlined formulations of the problem are typically accepted in order to further simplify the problem complexity. They range from considering static targets [8, 10, 13, 15, 16, 19, 20] instead of dynamic ones [4–7, 9, 11, 12, 14, 17, 21, 22] to modeling the sensors ideally [4–6, 8] instead of realistically (e.g., as radars [9–12] or downward-looking cameras [13, 14, 17, 19]) or to assuming that the UAVs fly following straight-lines according to the

eight cardinal directions [4–8] or optimized waypoints [17, 21, 22] instead of considering the physical maneuverability constraints induced by the dynamic models of the UAVs [9–16, 19, 20]. Additionally, in some cases (e.g., in [10, 13, 16, 17, 19, 20]) the approach uses a receding horizon controller to divide the UAVs trajectory into sequentially optimized sections, narrowing the optimization search space at the expense of constructing suboptimal myopic solutions. Finally, it is worth noting that the approaches are often tested over synthetic scenarios, built by the authors, without a clear relation with a real-world problem.

The previous simplifications and the lack of analysis over real-world scenarios reduce the applicability of the approaches to real-world problems. This is especially relevant within the subset of MTS approaches [4–10], as the majority of them have been developed for UAVs flying accordingly to the eight cardinal directions [4–8] or only tested over ideal sensors and synthetic unreal scenarios [4–6]. To pave the path of using MTS methods for real-world scenarios, this work extends the capabilities of the planner introduced in [10] by contributing to the following fields:

- (i) MTS mission definition, by combining intelligence and geographical information to construct the target probability models that will be used in real-world missions. This feature is inspired by software tools that use geographical information to build the target initial belief or motion model [24–26], monitor search missions [26, 27], or evaluate predefined search patterns [24, 28]. Its main benefit, as the results of this paper will show, will be the substitution of the synthetic scenarios used in previous MTS planners by real-world inspired ones.
- (ii) MTS planning, by optimizing simultaneously the UAV trajectory (by means of changing the UAV heading, speed, and height) and camera pose (azimuth and elevation). This new competence combines the UAV trajectory optimization capabilities of previous MTS planners (which usually manipulate only the UAV heading) with the sensor moving capabilities of only a few PTSP approaches [21, 22]. As this innovation supports a higher moving capability of the camera footprint and a quicker coverage of the high probability areas, it is especially relevant for MTS where the target has to be detected as soon as possible.
- (iii) Sensor characterization, by incrementing the realism of the camera detection behavior, modeling the effects of the terrain elevation (occlusions and target-camera distance variation), camera orientation, and target and sensor size in the footprint dimensions and sensor likelihood. Not only is the realism of the likelihood model crucial to shorten the differences between the simulated and real behavior of the camera, but it will also be regarded during the mission definition presented in the Results section in order to set up the scenarios correctly.

In short, this paper presents a new planner that optimizes the UAV trajectories and onboard cameras orientations in

real-world dynamic MTS scenarios, where the behavior of a dynamic target is probabilistically modeled from a novel perspective by combining intelligence and geographic information, and whose new camera likelihood model takes into account the effects of the terrain elevation in the camera detection capabilities. Besides, we extend the capabilities of the MTS planner presented in [10], which only manipulated the trajectory heading of a set of UAVs equipped with fixed radars in order to optimize the detection time of static targets in synthetically built scenarios, with the simultaneous optimization of UAVs trajectories and sensor poses over realistically built scenarios with dynamic targets. Moreover, the new planner also considers the extension to heterogeneous UAVs, which can have different parametrizations and start and leave the search mission at different times. Finally, this paper describes in detail, and in an algorithmic form, the functionality of the new planner with the purpose of clarifying the interaction between the different elements during the optimization process.

The remaining of this paper is organized as follows. The second section introduces the probabilistic formulation of the MTS problem, describes the novel approaches used to model the target initial state and motion behavior, and presents the new models used for the UAVs and their cameras. The third section details the new multisteped MTS planner presented in this paper, introducing its steps in an algorithmic form and analyzing its computational complexity. The fourth section compares the state-of-the-art of the closest related work with our new planner and highlights its differences with the previous planner in [10]. The fifth section analyzes its performance over three different real search scenarios and shows the benefits of letting the planner reorient the camera. And in the ending section, the conclusions are drawn and some open research questions discussed.

2. Minimum Time Search Definition

This section presents the probabilistic formulation of the MTS problem and describes the approaches used in this work to model the target, UAV, and sensor behavior.

2.1. General Problem Formulation. In the MTS problem presented in this paper, there is a set of U UAVs overflying a search area Ω (discretized into a grid G_Ω of $w_x \cdot w_y$ square cells) in order to detect the presence of a single target located within it. Besides, due to the uncertainty associated with the position and dynamics of the target and to the measurements of the onboard sensor of each UAV, the MTS problem is formulated within the probabilistic framework introduced in [5].

In more detail, the information about the target initial position (represented by random variable τ^0) is modeled with the initial target belief $b(\tau^0) \triangleq P(\tau^0)$, which is a probability mass function that represents the chances of finding the target at each $\tau^0 \in G_\Omega$. Besides, when the target is moving, the information about its dynamics is described with the target Markovian motion model $P(\tau^t | \tau^{t-T_\tau})$, which states the probability that the target moves from any cell $\tau^{t-T_\tau} \in G_\Omega$ to any cell $\tau^t \in G_\Omega$ during a time lapse T_τ . Moreover, the

detection capabilities of the onboard sensor of each UAV are described with the likelihood function $P(z_u^t | \tau^t, s_u^t)$, which returns the probability of detecting ($z_u^t = D$) or not detecting ($z_u^t = \bar{D}$) a target placed at τ^t from the UAV and sensor pose stored in the state variable s_u^t of the u -th UAV.

For those readers familiarized with the Recursive Bayesian Filter (RBF, [12]), the previous probability models $b(\tau^0)$, $P(\tau^t | \tau^{t-T_\tau})$, and $P(z_u^t | \tau^t, s_u^t)$ are used to obtain the target belief $b(\tau^t) \triangleq P(\tau^t | z_{1:U}^{0:t}, s_{1:U}^{0:t})$, which represents the chances of finding the target at time step t and at each cell $\tau^t \in G_\Omega$, given the trajectory of the UAVs and sensors poses $s_{1:U}^{0:t}$ and the sensor measurements $z_{1:U}^{0:t}$. The RBF process, stated in (1), calculates the current belief $b(\tau^t)$ from the previous time step belief $b(\tau^{t-T_\tau}) \triangleq P(\tau^{t-T_\tau} | z_{1:U}^{0:t-T_\tau}, s_{1:U}^{0:t-T_\tau})$ by (1) incorporating the current UAVs location and sensor poses $s_{1:U}^t$ and the last sensor measurements $z_{1:U}^t$ with the likelihood functions $P(z_u^t | \tau^t, s_u^t)$, and by (2) displacing and redistributing the target location probability with the target motion model $P(\tau^t | \tau^{t-T_\tau})$. Besides, (1) assumes that the sensors provide measurements only at time steps that are multiple of T_τ and its normalization factor γ is used to ensure that $\sum_{\tau^t \in G_\Omega} b(\tau^t) = 1$.

$$b(\tau^t) = \gamma \prod_{u=1:U} P(z_u^t | \tau^t, s_u^t) \sum_{\tau^{t-T_\tau} \in G_\Omega} P(\tau^t | \tau^{t-T_\tau}) b(\tau^{t-T_\tau}) \quad (1)$$

A MTS mission can be formulated as an optimization problem, where some criteria related to the previous probabilistic models and other mission objectives are evaluated in order to determine the best UAV trajectories and sensor poses $s_{1:U}^{0:t^{end}}$ during the duration of the mission t^{end} . A useful criterion for MTS is the Expected Time of Detection (ETD, [5–7, 9, 10]), which measures the average time of detection of the target when the UAVs and the sensors follow the trajectories and poses defined in a given $s_{1:U}^{0:t^{end}}$. It is calculable with (2)-(4), setting $\tilde{b}(\tau^t)$ to $b(\tau^0)$ for the initial case ($t = 0$). The first equation is similar to RBF equation (1), but as it lacks the normalization term γ and all measurements $z_{1:U}^{0:t}$ are nondetection, it obtains the “unnormalized belief” $\tilde{b}(\tau^t)$ assuming that the onboard sensors do not detect the target from the UAVs trajectories and sensor poses in $s_{1:U}^{0:t}$. The second one obtains $P_{nd}(s_{1:U}^{0:t}) \triangleq P(\bar{D}_{1:U}^{0:t} | s_{1:U}^{0:t})$, which is the probability of not detecting the target from $s_{1:U}^{0:t}$ and whose value decays as the sensors make nondetection observations in regions with probability of target presence. The third expression obtains the ETD from $s_{1:U}^{0:t^{end}}$ when $P_{nd}(s_{1:U}^{0:t})$ becomes 0 for some $t \leq t^{end}$ and underestimates its value when $P_{nd}(s_{1:U}^{0:t^{end}}) > 0$. Finally, it is worth noting that minimizing $ETD(s_{1:U}^{0:t^{end}})$ is a better option for MTS than maximizing the probability of detection along the whole trajectory $P_d(s_{1:U}^{0:t^{end}}) = 1 - P_{nd}(s_{1:U}^{0:t^{end}})$, as the MTS objective is to collect as much probability as possible sooner [4–6].

$$\begin{aligned} \bar{b}(\tau^t) &= \prod_{u=1:U} P(z_u^t = \bar{D} \mid \tau^t, s_u^t) \\ &\cdot \sum_{\tau^{t-T_\tau} \in G_\Omega} P(\tau^t \mid \tau^{t-T_\tau}) \bar{b}(\tau^{t-T_\tau}) \end{aligned} \quad (2)$$

$$P_{\text{nd}}(s_{1:U}^{0:t}) = \sum_{\tau^t \in G_\Omega} \bar{b}(\tau^t) \quad (3)$$

$$\text{ETD}(s_{1:U}^{0:t^{\text{end}}}) = \sum_{t=0}^{t^{\text{end}}} P_{\text{nd}}(s_{1:U}^{0:t}) \cdot T_\tau \quad (4)$$

Besides, in order ensure that the UAV trajectories and sensor poses $s_{1:U}^{0:t^{\text{end}}}$ calculated by our MTS planner are feasible from the maneuverability point of view, we exploit the UAV and sensor deterministic dynamic model of each UAV-sensor pair $\dot{s}_u^t = f(s_u^t, a_u^t)$, where a_u^t stands for the set of control actions (e.g., commanded UAV heading, height & speed, and sensor elevation & azimuth) and \dot{s}_u^t for the time derivative of the UAV location and sensor poses. In particular, our MTS planner uses this dynamic model to obtain the solutions (best UAV trajectories and sensor poses $s_{1:U}^{0:t^{\text{end}}}$) from the initial UAVs locations $s_{1:U}^0$ and the sequence of sets of control actions $a_{1:U}^{0:t^{\text{end}}}$ proposed, manipulated, and evaluated by our approach in order to optimize MTS missions. Finally, it is worth highlighting that due to the deterministic nature of the UAV and sensor dynamic model, there is an unambiguous relation between the best sequence of control actions and the best UAVs trajectories and sensor poses.

The realism of the four models (i.e., of the initial target belief $b(\tau^0)$, of the target motion model $P(\tau^t \mid \tau^{t-T_\tau})$, of the sensor likelihood $P(z_u^t \mid \tau^t, s_u^t)$, and of the UAV and sensor motion model $\dot{s}_u^t = f(s_u^t, a_u^t)$) is crucial to avoid discordances between the real ETD of a given $s_{1:U}^{0:t}$ and the ETD calculated by our approach. Hence, in the rest of this section we present the new models proposed in this paper to bring realism to the definition and solution of MTS missions performed by fixed-wing UAVs equipped with orientable cameras.

2.2. Initial Target Belief Definition. To construct the initial target belief $b(\tau^0)$, we merge knowledge coming from different sources, using a different probability layer $b^l(\tau^0)$ for each information source and performing with (5) an addition of the L probability layers weighted with their reliability/importance coefficients w_l . In other words, considering the first term in (5) is a normalization coefficient that ensures that $b(\tau^0)$ is a probability function (i.e., $\sum_{\tau^0 \in G_\Omega} b(\tau^0) = 1$), our initial target belief $b(\tau^0)$ is calculated as the mixture of the beliefs $b^l(\tau^0)$ associated with the different l -th information sources.

$$b(\tau^0) = \frac{1}{\sum_{l=1}^L w_l} \sum_{l=1}^L w_l b^l(\tau^0) \quad (5)$$

The layers can be associated with geographical information (e.g., terrain altitude, road maps) or intelligence/user-defined clues (e.g., last/habitual location areas of the target).

For the examples of this work, we consider the following two layers:

- (1) The terrain elevation probability layer $b^1(\tau^0)$, obtained with the following steps. First the digital elevation model (DEM, [29]) of Ω is automatically resampled to the size of the cells in G_Ω in order to obtain the average height of each cell in G_Ω . Next the user/operator is required to divide the existing elevations within the cells in G_Ω into consecutive ranges and to assign a chance of target presence to each range. Finally, the method automatically determines the cells in each elevation range and the probability associated with all $\tau^0 \in G_\Omega$, distributing the chances of target presence assigned by the operator. As this way of proceeding generates a geographical probability layer $b^1(\tau^0)$ where areas with similar altitude share the same initial belief, it automatically spreads uniformly the belief over different regions of the search area.
- (2) The intelligence probability layer $b^2(\tau^0)$. For this layer the operator has to graphically define a mixture (weighted addition) of Gaussians (centered in eligible locations of the search area Ω and spread according to selectable standard deviations) and of polygonal areas (defined by their external points placed in the desired locations of the search area Ω) with uniform probabilities (assigned by the operator). The weights of each element (each Gaussian and/or each polygonal area) within the intelligence probability layer $b^2(\tau^0)$ are also selected by the operator according to the information gathered about the last known location of the target.

As an illustrative example, we show the initial belief defined by an operator when looking for a drifting boat next to the coast in Areia Branca, Brazil. To obtain the probability elevation layer $b^1(\tau^0)$ shown at the bottom of Figure 2(b), where darker/lighter greys are associated with higher/lower probability cells, the operator analyzes the elevation map represented in Figure 2(a) and assigns high chances of target presence to the elevations by the coast (represented in dark green in the elevation map) to model a strong belief that the boat may have arrived a ground when the search mission starts, lower chances to the sea-level elevation (in blue) to model a moderate belief that the target is still navigating, and zero chances to higher inland elevations. In order to define the intelligence probability layer $b^2(\tau^0)$, represented at the top of Figure 2(b), the operator centers a moderate spread Gaussian in the last known position of the boat. Finally, the operator assigns a weight for each layer ($w_1 = 2$ and $w_2 = 1$) in order to produce the initial target belief $b(\tau^0)$ shown in Figure 2(c).

2.3. Target Motion Model Definition. To define that target motion model, we can distinguish two cases:

- (1) Scenarios with static targets. In this case, and due to the immobility of the target, $P(\tau^t \mid \tau^{t-T_\tau}) = 1$ if and

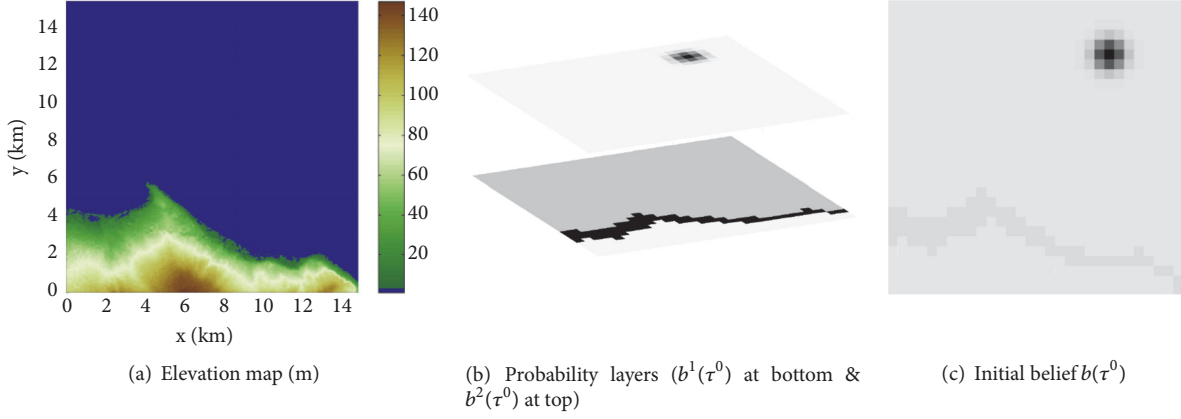


FIGURE 2: Initial target belief definition used for a drifting boat (in Areia Branca, Brazil).

only if τ^t and τ^{t-T_τ} refer to the same cell in G_Ω , and $P(\tau^t \mid \tau^{t-T_\tau}) = 0$ otherwise. Besides, in this case, the equations that contain the term $\sum_{\tau^{t-T_\tau} \in G_\Omega} P(\tau^t \mid \tau^{t-T_\tau})b(\tau^{t-T_\tau})$ can be simplified as $\sum_{\tau^{t-T_\tau} \in G_\Omega} P(\tau^t \mid \tau^{t-T_\tau})b(\tau^{t-T_\tau}) = b(\tau^{t-T_\tau})$.

(2) Scenarios with dynamic targets. To construct this type of target motion model, we combine different types of information (e.g., elevation data and sea currents). In this work, the operator has to provide the following:

- (i) The elevation range ER where the target is allowed to move within Ω . This allows the operator to indirectly assign the probabilities corresponding to the static target behavior to those cells that do not belong to the ER . In other words, $\forall \tau^{t-T_\tau} \notin moving(G_\Omega, ER)$, the target motion definition process automatically makes $P(\tau^t \mid \tau^{t-T_\tau}) = 1$ if $\tau^t = \tau^{t-T_\tau}$ and $P(\tau^t \mid \tau^{t-T_\tau}) = 0$ otherwise.
- (ii) K vectors $\mathbf{v}_{1:K}$, each with 9 values that represent, for a few selected cells $c_{1:K} \subset G_\Omega$, the chances of moving to their 8 neighbor cells $\mathcal{N}(c_k)$ and of staying at the same cell c_k . Besides, in order to force the target to stay within the search area Ω , some of the 9 values of the vectors \mathbf{v}_k that correspond to cells in the borders of G_Ω are automatically set to zero. With this information, the target motion definition process makes $P(\tau^t \mid \tau^{t-T_\tau}) = 0$ for all $\tau^t \notin \mathcal{N}(\tau^{t-T_\tau}) \cup \{\tau^{t-T_\tau}\}$ and computes $P(\tau^t \mid \tau^{t-T_\tau})$ applying the potential field method (according to the cardinal directions that make the target move from one cell to its neighbors) for all $\tau^t \in \mathcal{N}(\tau^{t-T_\tau}) \cup \{\tau^{t-T_\tau}\}$ and $\tau^{t-T_\tau} \in moving(G_\Omega, ER)$. In more detail, if $adj(\tau^{t-T_\tau}, d)$ with $d \in \{1 : 8\}$ stands for the adjacent cell to τ^{t-T_τ} in the d -th cardinal direction, $adj(\tau^{t-T_\tau}, 9) = \tau^{t-T_\tau}$ represents stay in the same cell, $dist(\tau^t, c_k)$ is the distance between cell τ^t and cell c_k , and $v_k[d]$ is the d -th element of \mathbf{v}_k , then $P(\tau^t =$

$adj(\tau^{t-T_\tau}, d) \mid \tau^{t-T_\tau}) \propto \sum_{k=1:K} v_k[d] / dist(\tau^t, c_k)$ and $\sum_{d \in \{1:9\}} P(\tau^t = adj(\tau^{t-T_\tau}, d) \mid \tau^{t-T_\tau}) = 1$. This way of proceeding allows the operator to define the distribution of the target probability around the neighborhood cells in only a few cells $c_{1:K} \subset G_\Omega$ and extend this definition to the remaining moving cells taking into account the distance between the selected cells and the others.

- (iii) The time period T_τ that has to be used to apply $P(\tau^t \mid \tau^{t-T_\tau})$ to the belief $b(\tau^{t-T_\tau})$. As the speed of the target is related to the quotient of the size of the cell and T_τ , this value is used by the planner to relate the simulation of the motion of the target to the simulation of the motion of the UAVs and sensors.

As an illustrative example, we also obtain the target motion model of the boat drifting example next to the cost in Areia Branca, Brazil. In this case, the operator defines \mathbf{v}_k for three cells $c_k \in G_\Omega$ in the sea, making $v_k[9] = 0.1$, $v_k[d] = 2$ for one of the cardinal directions d towards the coast, and $v_k[r] = 0$ for the remaining directions $r \in \{1 : 8\} - \{d\}$. Figure 3 represents with the starting point of the red arrows the location of the selected $c_{1:3}$ and with the arrows themselves the direction d where $v_k[d] = 2$ for each c_k . Besides, the operator also makes the moving $ER \leq 0$ to only let the target (i.e., the drifting boat) move in the sea. To summarize the result of the target motion definition process, the green arrows in Figure 3 represent the direction obtained by weighting, for each cell $\tau^{t-T_\tau} \in moving(G_\Omega, ER)$, the arrows in the 8-cardinal directions d with $P(\tau^t = adj(\tau^{t-T_\tau}, d) \mid \tau^{t-T_\tau})$. Besides, the lack of inland green arrows indicates the static behavior of the target in this area. In short, the green arrows show how the defined target motion makes the belief over the sea move towards the coast and remain unchanged inland. Finally, it is worth highlighting that, thanks to our target motion model definition approach, this complex model is automatically built just considering the elevation range ($ER \leq 0$) and the target movement probabilities in three cells of G_Ω .

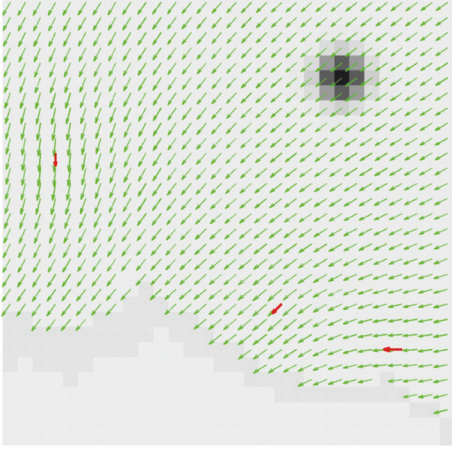


FIGURE 3: Target motion definition used for a drifting boat (in Areia Branca, Brazil).

2.4. Camera Likelihood with Terrain Occlusion. The likelihood of detecting the target with the onboard camera of each UAV in a given cell $\tau^t \in \Omega$ can be calculated scaling the Target Task Performance (TTP) metric [30] with the percentage of the selected cell within the camera footprint. This scaling, used to reduce the detection probability in the cells of the footprint border, can be modeled with (6), where $|\tau^t \cap \text{footprint}(s_u^t)|$ represents the area (size of the surface) of the common region of the cell τ^t and of the footprint of the camera (oriented and placed according to s_u^t), $|\tau^t|$ is the total area of cell τ^t , and $\text{TTPF}(\tau^t, s_u^t)$ is the target task performance function.

$$P(D_u^t | \tau^t, s_u^t) = \frac{|\tau^t \cap \text{footprint}(s_u^t)|}{|\tau^t|} \text{TTPF}(\tau^t, s_u^t) \quad (6)$$

To determine the footprint of the camera from the UAV location and camera pose in s_u^t , we follow the process schematized in Figure 4(a) and consisting of the following two steps. First, we calculate geometrically the camera footprint at sea level taking into account the following pieces of information within s_u^t : UAV location (x_u^t, y_u^t, h_u^t) and camera azimuth, elevation, and FOV $(\alpha_u^t, \eta_u^t, \text{FOV})$. To do this, we consider that the sensor location is the same as the UAV location (since its deviation from the location of the UAV mass center is negligible for the mission) and that its orientations are measured with respect to the vehicle coordinate frame (since the camera gimbal compensates the UAV attitude). Second, we approximate the corners of the real camera footprint by the intersections (obtained using the 3D Bresenham algorithm [31]) of the terrain elevation with the 4 lines that join the camera with the 4 corners of the sea-level camera footprint.

The value of the target task performance function TTPF for each cell τ^t within the sensor footprint from the UAV location and sensor pose in s_u^t is calculated with (7), where $V_T(\tau^t, s_u^t)$ is the cycle size of the target and $V_{50} = 2.03$ (according to [32, 33]) is the critical cycle size for detecting a small target with a likelihood of 50% (since when $V_T = V_{50}$,

$\text{TTPF} = 1/(1 + 1) = 0.5$). Moreover, the target cycle size is calculated with (8), where S_{target} is the real size of the target, $\beta(\tau_u^t)$ is the angle between the along-scan (vertical) and cross-scan (horizontal) directions of the footprint, and GSD_i (with $i \in \{\text{vertical}, \text{horizontal}\}$) is the ground sample distance (corresponding size of a pixel of the camera over terrain, obtainable as the ratio of the length of the footprint in each direction to its corresponding number of pixels). Taking into account the trigonometric relations between the UAV location and camera footprint, the angle $\beta(s_u^t)$ can be obtained solving (9) and the ground sample distance in each scan direction can be calculated with (10) and (11), where $h_{\text{terrain}}(\tau^t)$ is the height of the terrain at the target location τ^t and $\#_{\text{pixel}_i}$ is the number of pixels of the camera in the vertical or horizontal directions ($i \in \{v, h\}$).

$$\text{TTPF}(\tau^t, s_u^t) = \frac{(V_T(\tau^t, s_u^t)/V_{50})^{1.51+0.24V_T(\tau^t, s_u^t)/V_{50}}}{1 + (V_T(\tau^t, s_u^t)/V_{50})^{1.51+0.24V_T(\tau^t, s_u^t)/V_{50}}} \quad (7)$$

$$V_T(\tau^t, s_u^t) = \frac{S_{\text{target}}}{\sqrt{\text{GSD}_v(\tau^t, s_u^t) \cdot \text{GSD}_h(\tau^t, s_u^t) \cdot \sin(\beta(s_u^t))}} \quad (8)$$

$$\begin{aligned} & \text{tg}(\beta(s_u^t)) \\ &= \frac{\text{tg}^{-1}(\eta_u^t - \text{FOV}/2) - \text{tg}^{-1}(\eta_u^t + \text{FOV}/2)}{\text{tg}(\text{FOV}/2) [\sin^{-1}(\eta_u^t - \text{FOV}/2) - \sin^{-1}(\eta_u^t + \text{FOV}/2)]} \end{aligned} \quad (9)$$

$$\begin{aligned} & \text{GSD}_v(\tau^t, s_u^t) \\ &= \frac{h_u^t - h_{\text{terrain}}(\tau^t)}{\#_{\text{pixel}_v}} \left[\text{tg}^{-1}\left(\eta_u^t - \frac{\text{FOV}}{2}\right) - \text{tg}^{-1}\left(\eta_u^t + \frac{\text{FOV}}{2}\right) \right] \end{aligned} \quad (10)$$

$$\text{GSD}_h(\tau^t, s_u^t) = 2 \frac{h_u^t - h_{\text{terrain}}(\tau^t)}{\#_{\text{pixel}_h}} \frac{\text{tg}(\text{FOV}/2)}{\sin(\eta_u^t)} \quad (11)$$

As an illustrative example of the likelihood model, Figure 4(b) shows how the probability of detection, proportional to TTPF, is incremented as the cycle size of the target V_T grows. Hence, and due to (8)-(11), the probability of detection grows as the target size S_{target} is increased, as the UAV flying altitude with respect to the terrain elevation gets smaller (through $\text{GSD}_i(\tau^t, s_u^t)$), as the FOV of the camera is reduced (through $\beta(s_u^t)$ and $\text{GSD}_i(\tau^t, s_u^t)$), or as the camera elevation gets closer to the vertical one (through $\beta(s_u^t)$ and $\text{GSD}_i(\tau^t, s_u^t)$). In more detail, Figure 4(c) shows several TTPF curves, corresponding to different FOV and η_u^t values (indicated in the legend), versus the UAV flying altitude with respect to the terrain elevation (i.e., $h_u^t - h_{\text{terrain}}(\tau^t)$) for a target of $S_{\text{target}} = 5m$ (e.g., the boat in Areia Branca).

Finally, note that the camera likelihood function defined by (6)-(10) is able to observe partially the borders of the footprints and provide different values for targets of different sizes; cells observed from different UAV heights, camera azimuth, and elevation angles; etc. Hence, our model does not nullify the likelihood that those cells are partially within the footprint but their center falls outside the footprint as in [17, 19], and allows the planner to move the camera from the downward-looking pose assumed in other camera models tested in probabilistic target search problems [13, 14, 17].

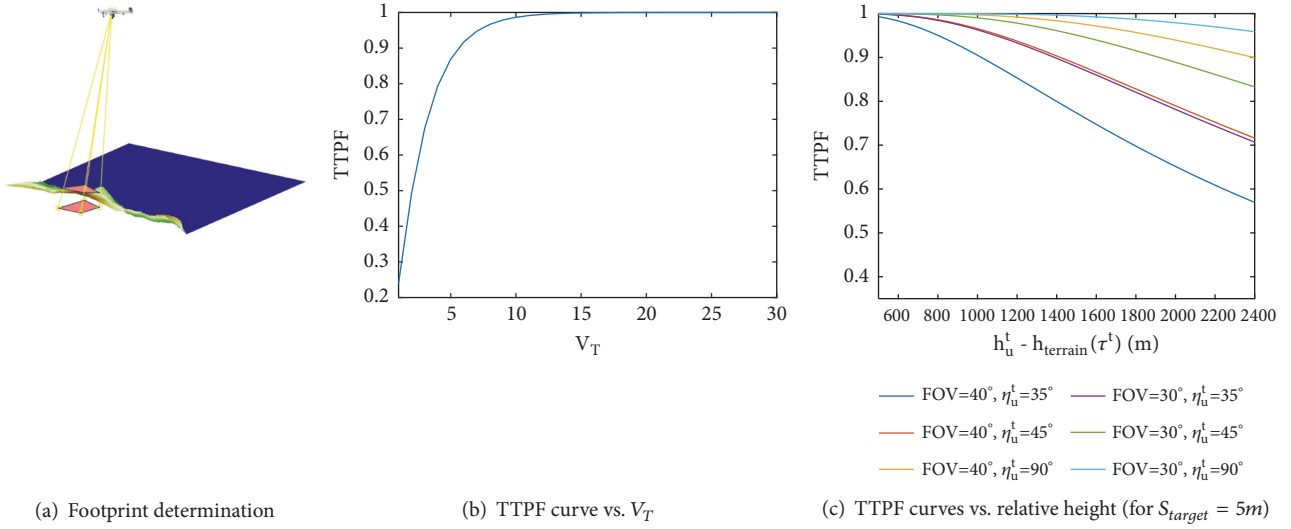


FIGURE 4: Camera likelihood model.

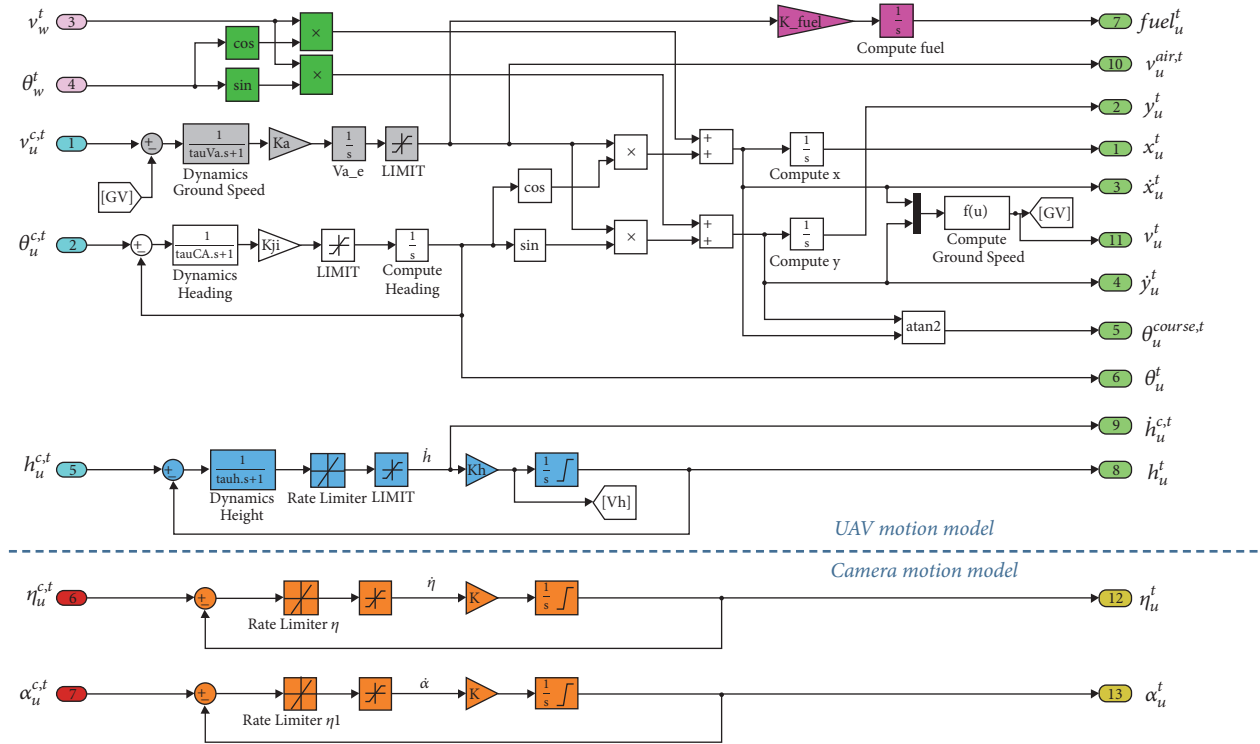


FIGURE 5: Simulink UAV and camera motion model

2.5. UAV and Sensor Dynamic Models. The UAV dynamics corresponds to a fixed-wing UAV modeled with the upper part of the nonlinear parametrizable Simulink model represented in Figure 5 or the differential equations of the appendix. The UAV motion variables within s_u^t , highlighted in light green on the right of the figure, are the UAV 3D location (x_u^t, y_u^t, h_u^t), 3D speeds ($\dot{x}_u^t, \dot{y}_u^t, \dot{h}_u^t$), heading (θ_u^t), course angle ($\theta_u^{course,t}$), air and ground velocity ($v_u^{air,t}$ and v_u^t), and fuel consumption ($fuel_u^t$). The UAV motion variables within the command a_u^t , highlighted in cyan on the left, are

the commanded UAV velocity ($v_u^{c,t}$), heading ($\theta_u^{c,t}$), and height ($h_u^{c,t}$). Additional inputs to the UAV dynamical model are the wind speed (v_w^t) and direction (θ_w^t), highlighted on the left in pink. Besides, in order to let the reader identify the UAV dynamics within the Simulink model, the blocks associated with its height dynamics are colored in blue, with its velocity in grey, with its lateral displacement in white, with the wind in green, and with the fuel in magenta.

The sensor dynamic model is represented at the bottom of Figure 5 and corresponds to a gimbaled camera whose scan

Require: sc	▷ Structure with scenario models and times: $b(\tau^0), P(\tau^t \tau^{t-1}), P(z^t \tau^t, s_u^t), f(s_u^t, a_u^t), t_{1:U}^{start}, t_{1:U}^{end}$
Require: op	▷ Structure with optimizer parameters
Require: $s_{1:U}^{t_{start}}$	▷ Initial UAVs locations and sensor poses $\forall u = \{1 : U\}$
Require: Q	▷ Duration of each subsequence
1: $t^{end} = \max(sc.t_{1:U}^{end})$	▷ Determine the duration of the mission
2: $R = \lceil t^{end}/Q \rceil$	▷ Determine the number of sections required for the multi-stepped planner
3: $t^{end} = Q \cdot R$	▷ Round the duration of the mission
4: $sc.T_b = obtainT(sc.T_{1:U}^z, sc.T_\tau)$	▷ Determine minimum time step to update $\tilde{b}(\tau^t)$.
5: $t \leftarrow 0$	▷ Set starting time
6: $\tilde{b}(\tau^t) = sc.b(\tau^0)$	▷ Initialize $\tilde{b}(\tau^t)$
7: $FCV^t = [0, 0, 0, 0, 0, 0]$	▷ Initialize fitness criteria vector
8: for $r=1:R$ do	▷ For each subsequence
9: $[s_{1:U}^{t:t+Q}, FCV^{t+Q}, \tilde{b}(\tau^{t+Q})] \leftarrow \mathbf{optimGA}(sc, op, s_{1:U}^t, \tilde{b}(\tau^t), t, Q, FCV^t)$	▷ Obtain best subsequence
10: $t \leftarrow t + Q$	▷ Update time for the following section
11: end for	
12: return $s_{1:U}^{0:t^{end}}$	▷ Complete UAV trajectory and sensor poses

ALGORITHM 1: Multisteped GA-based planner.

direction can be changed commanding its elevation ($\eta_u^{c,t}$) and azimuth ($\alpha_u^{c,t}$) angles, with the model inputs highlighted in red on the left of the figure. Besides, the camera motion variables within s_u^t are the outputs of the model highlighted in yellow on the right (camera elevation η_u^t and azimuth α_u^t) and the blocks of the camera dynamics are colored in orange.

The whole model includes the usual limitations related to the UAV speed, height, and heading and to the camera elevation and azimuth. Its different parameters (provided in an external input file, eligible by the operator) allow adapting the movement of the UAV and of the camera to the behavior of different real-world aircraft and camera gimbals. Finally, the model integration, performed from each UAV and camera initial state s_u^0 using the values of a sequence of commands $a_u^{0:t^{end}}$, allows obtaining the $s_u^{0:t^{end}}$ with a given simulation resolution time step T_s .

3. MTS Planner

The MTS planner presented in this work is an extension of the Genetic Algorithm (GA) planner introduced in [10]. This section details the main elements of the new planner, introduces its algorithmic description following a top-down strategy, and discusses its computational cost.

3.1. Multisteped GA-Based Planner. The MTS planner obtains the UAVs trajectories $s_{1:U}^{0:t^{end}}$ for a given mission time t^{end} following a receding horizon controller approach [10, 13, 16, 17, 19, 20]. More concretely, the planner divides $s_{1:U}^{0:t^{end}}$ into R different sections of duration Q and optimizes each section ($s_{1:U}^{Q(r-1):Qr}$ with $r = \{1 : R\}$) sequentially, using a GA whose inputs are the final state of the last section $s_{1:U}^{Q(r-1)}$ and its “unnormalized belief” $\tilde{b}(\tau^{Q(r-1)})$, and whose outputs are the new $s_{1:U}^{Q(r-1):Qr}$ and $\tilde{b}(\tau^{Qr})$. Besides, it incorporates the possibility of letting each UAV engage and leave the MTS mission at different time instants t_u^{start} and t_u^{end} (without loss

of generality, we assume that for at least one UAV, $t_u^{start} = 0$) and of using different time steps for the measures of each UAV T_u^z and for the target update T_τ . Hence, to obtain sequentially the sections of the UAV trajectories (and of the cameras orientations), the new planner obtains the overall mission ending time, calculates the number of sections R required for the mission, and considers the UAVs and camera immobile and disabled for those time steps where they are not engaged in the MTS mission (i.e., $s_u^{0:t_{start}} = s_u^{t_{start}}$ and $s_u^{t_{end}:t^{end}} = s_u^{t_{end}}$). Besides, it also obtains the minimum time step T_b required to update the “unnormalized belief” $\tilde{b}(\tau^t)$ during the evaluation of the ETD taking into account the time steps of target motion model T_τ and of the camera of each UAV T_u^z .

The algorithmic description of the new multisteped planner is sketched in the pseudocode of Algorithm 1. It starts computing the mission duration t^{end} , the number of sections R into which $s_{1:U}^{0:t^{end}}$ will be divided, and the time lapse T_b required to update the “unnormalized belief” $\tilde{b}(\tau^t)$. Next, it initializes the “unnormalized belief” $\tilde{b}(\tau^0)$ and the fitness criteria vector FCV^0 , which stores the different values of the criteria used by the GA to decide if a solution (proposal of UAV trajectories and sensor poses in $s_{1:U}^{0:t}$) is better than another. Finally, it sequentially optimizes each of the sections of the trajectory (i.e., each $s_{1:U}^{t:t+Q}$ with $t = Q(r-1)$ and $r = \{1 : R\}$) using the optimizer in $\mathbf{optimGA}(\cdot)$, which will be described in detail in the following section.

3.2. GA Optimizer. This section details the most relevant aspects of the GA used to determine the best UAV trajectory and sensor pose for each section ($s_{1:U}^{t:t+Q}$ with $t = Q(r-1)$) of the whole solution $s_{1:U}^{0:t^{end}}$. We start presenting the GA genotype, continue with the main GA steps and operators, and finish with the evaluation of each possible solution $s_{1:U}^{0:t^{end}}$.

3.2.1. GA Decision Variables. The decision variables directly manipulated by the genetic operators (i.e., the GA genotype)


```

1: procedure OPTIMGA( $sc, op, \bar{b}(\tau^t), s_{1:U}^t, t, Q, FCV^t$ )
2:    $J \leftarrow op.J$  ▷ Get population size
3:    $N_{1:U} \leftarrow obtainActionNumber(t, Q, T_a, sc, s_{1:U}^{start}, sc, t_{1:U}^{start})$  ▷ Obtain number of actions for each UAV
4:    ${}^{1:J}a_{1:U}^{t:t+Q} \leftarrow initialize(J, N_{1:U}, op.init)$  ▷ Generate J sequences with the required actions
5:   for  $j=1:J$  do
6:      ${}^jFCV^{t+Q} \leftarrow sim\&Eval(sc, \bar{b}(\tau^t), s_{1:U}^t, t, Q, FCV^t, {}^j a_{1:U}^{t:t+Q})$  ▷ Simulate and evaluate
7:   end for
8:    $M \leftarrow 1$  ▷ Initialize counter of the algorithm iterations (generations)
9:   while  $getCompTime() \leq opt.CompTime$  do ▷ Main optimization loop
10:     ${}^{1:J}_{par}a_{1:U}^{t:t+Q} \leftarrow selectParents({}^{1:J}_{par}a_{1:U}^{t:t+Q}, {}^{1:J}_{par}FCV^{t+Q})$  ▷ Select pair of parents (tournament selection)
11:     ${}^{1:J}_{child}a_{1:U}^{t:t+Q} \leftarrow crossover({}^{1:J}_{par}a_{1:U}^{t:t+Q}, op.xover)$  ▷ Perform crossover of parents (single point crossover)
12:     ${}^{1:J}_{child}a_{1:U}^{t:t+Q} \leftarrow mutate({}^{1:J}_{child}a_{1:U}^{t:t+Q}, op.mut)$  ▷ Mutate children (incremental gaussian mutation)
13:    for  $j=1:J$  do
14:       ${}^j_{child}FCV^{t+Q} \leftarrow sim\&Eval(sc, \bar{b}(\tau^t), s_{1:U}^t, t, Q, FCV^t, {}^j_{child}a_{1:U}^{t:t+Q})$  ▷ Simulate and evaluate
15:    end for
16:     $[{}^{1:J}_{par}a_{1:U}^{t:t+Q}, {}^{1:J}_{par}FCV^{t+Q}] \leftarrow recombine({}^{1:J}_{par}a_{1:U}^{t:t+Q}, {}^{1:J}_{par}FCV^{t+Q}, {}^{1:J}_{child}a_{1:U}^{t:t+Q}, {}^{1:J}_{child}FCV^{t+Q})$ 
17:     $M \leftarrow M + 1$  ▷ Increment the GA generation number
18:  end while
19:   ${}^{best}_{1:U}a_{1:U}^{t:t+Q} \leftarrow best({}^{1:J}_{par}a_{1:U}^{t:t+Q}, {}^{1:J}_{par}FCV^{t+Q})$  ▷ Select best solution
20:   $[{}^{best}_{1:U}FCV^{t+Q}, {}^{best}_{1:U}s_{1:U}^{t:t+Q}, {}^{best}_{1:U}\bar{b}(\tau^{t+Q})] \leftarrow sim\&Eval(sc, \bar{b}(\tau^t), s_{1:U}^t, t, Q, FCV^t, {}^{best}_{1:U}a_{1:U}^{t:t+Q})$  ▷ Sim. and evaluate it
21:  return  ${}^{best}_{1:U}a_{1:U}^{t:t+Q}, {}^{best}_{1:U}FCV^{t+Q}, {}^{best}_{1:U}\bar{b}(\tau^{t+Q})$  ▷ Return best solution
22: end procedure

```

ALGORITHM 2: optimGA: NSGA-II for MTS.

of the MTS planner are the subsequence of control actions $a_{1:U}^{Q(r-1):Qr}$ used to obtain, by means of the UAV motion model, the r -th subsection of the UAV trajectories $s_{1:U}^{Q(r-1):Qr}$ (i.e., the GA phenotype or solution). Besides, the actions in a_u^t are the commanded UAV heading $\theta_u^{c,t}$, speed $v_u^{c,t}$, and height $h_u^{c,t}$ and the camera elevation $\eta_u^{c,t}$ and azimuth $\alpha_u^{c,t}$. Additionally, the number of decision variables is reduced by applying each action constantly during a time lapse T_a and by adapting the actions required for each UAV within each r -th subsection to the time steps where the UAV is engaged in the MTS mission. In other words, the GA algorithm only obtains the values of Q/T_a sets of actions (UAV heading, speed, and height; camera elevation and azimuth) for each UAV engaged in the mission during the r -th trajectory subsection.

3.2.2. GA General Description. The GA that sequentially optimizes each of the trajectory sections $s_{1:U}^{Q(r-1):Qr}$ based on the set of command actions ($a_u^t = [\theta_u^{c,t}, v_u^{c,t}, h_u^{c,t}, \eta_u^{c,t}, \alpha_u^{c,t}]$) and fitness criteria (which will be explained in the following section) maintains the operators, parameters, and structure of a standard GA and includes a new step to precalculate the number of actions required by each UAV.

To present it in an algorithmic form, we extend the notation of the paper with left superindexes ($1 : J$ or $best$) and left subindexes (par and $child$). The first superindex is used to indicate that there is a population or group of J action sequences ${}^{1:J}a_{1:U}^{t:t+Q}$ and fitness criteria vectors ${}^{1:J}FCV^{t+Q}$. The second one is used to identify the best action sequence ${}^{best}_{1:U}a_{1:U}^{t:t+Q}$, UAV trajectories and sensor poses ${}^{best}_{1:U}s_{1:U}^{t:t+Q}$, and fitness criteria vector ${}^{best}_{1:U}FCV^{t+Q}$ within the population of $1 : J$ solutions. And the left subindexes are incorporated

to the previous notation when the algorithm needs to distinguish between the generic population elements (${}^{1:J}a_{1:U}^{t:t+Q}$, ${}^{1:J}FCV^{t+Q}$) and those associated with the parents selected by the GA (${}^{1:J}_{par}a_{1:U}^{t:t+Q}$) and with their children (${}^{1:J}_{child}a_{1:U}^{t:t+Q}$, ${}^{1:J}_{child}FCV^{t+Q}$).

The GA in Algorithm 2 performs the following steps. First, step 3 computes the number of actions of T_a duration required by each UAV in the current subsection of the trajectory, taking into account the current time step t and each UAV starting and ending engaging time (t_u^{start} and t_u^{end}). Step 4 initializes a random population of the J action sequences ${}^{1:J}a_{1:U}^{t:t+Q}$ required for the UAVs engaged in the mission, using a parametrizable uniform distribution. Steps 5-7 simulate the UAV trajectories and sensor poses from the sequence of actions and evaluate them according to the criteria presented in the following section. In step 9, the GA enters an optimization loop which ends when a fixed computation time has passed. Next, step 10 selects J parents of action sequences from the current population using binary tournament selection; step 11 creates the children by combining pairs of parents using a single-point crossover; step 12 slightly modifies the children with an incremental Gaussian noise mutation; steps 13-15 simulate and evaluate them; and step 16 selects the survival population from the previous and the mutated children using NSGA-II recombination [34]. Once the computation time has finished, step 19 identifies the best solution in the population; step 20 obtains the FCV , trajectories and sensor poses, and “unnormalized belief” associated with it; and step 21 returns them.

It is worth noting that there are several steps whose behavior can be configured using the information in the opt

```

1: procedure SIM&EVAL( $sc, \tilde{b}(\tau^t), s_{1:U}^t, t, Q, FCV^t, a_{1:U}^{t:t+Q}$ )
2:    $t^{init} \leftarrow t$ 
3:    $t^{last} \leftarrow t + Q$ 
4:    $\triangleright$  Simulation of the proposed sequence of actions to obtain UAV trajectory and fuel consumption
5:   for  $u=1:U$  do
6:      $[s_u^{t:t+Q}, fuel_u] \leftarrow sim(sc, f(\cdot), s_u^t, a_u^{t:t+Q}, t, t + Q, \triangleright$  Simulate each UAV and sensor movement
7:        $sc.t_u^{start}, sc.t_u^{end}, sc.T_a, sc.T_s)$ 
8:   end for
9:    $\triangleright$  Evaluation of the trajectory obtained from the proposed sequence of actions
10:   $FCV^t[COL] \leftarrow FCV^t[COL] + \sum_{u=1:U} \sum_{l=u+1:U} collision(sc, s_u^{t:t+Q}, s_l^{t:t+Q}) \triangleright$  Update collision criterion
11:   $FCV^t[NFZ] \leftarrow FCV^t[NFZ] + \sum_{u=1:U} withinNFZ(s_u^{t:t+Q}) \triangleright$  Update fuel consumption
12:   $FCV^t[SMOOTH] \leftarrow FCV^t[SMOOTH] + \sum_{u=1:U} smooth(a_u^{t:t+Q}) \triangleright$  Update smooth criterion
13:   $FCV^t[FUEL] \leftarrow FCV^t[FUEL] + \sum_{u=1:U} fuel_u \triangleright$  Update fuel consumption
14:   $\triangleright$  Simulation of the target and evaluation of the probabilistic criteria
15:   $t \leftarrow t^{init} + sc.T_b$ 
16:  while  $t \leq t^{last}$  do
17:    if  $motionTarget(t, sc.T_r)$  then
18:       $\tilde{b}(\tau^{t-sc.T_r}) = \tilde{b}(\tau^t) \triangleright$  Rename variable for the target transition
19:       $\tilde{b}(\tau^t) \leftarrow \sum_{\tau^{t-sc.T_r} \in G_\Omega} P(\tau^t | \tau^{t-sc.T_r}) \tilde{b}(\tau^{t-sc.T_r}) \triangleright$  Target motion part of Equation (2)
20:    end if
21:     $\tilde{b}(\tau^t) \leftarrow \prod_{u \in measuring(u,t,sc)} P(z_u^t = \bar{D} | \tau^t, s_u^t) \tilde{b}(\tau^t) \triangleright$  Measurement update part of Equation (2)
22:     $P_{nd}(s_{1:U}^{0:t}) \leftarrow \sum_{\tau^t \in G_\Omega} \tilde{b}(\tau^t) \triangleright$  Equation (3)
23:     $FCV^t[ETD] \leftarrow FCV^t[ETD] + P_{nd}(s_{1:U}^{0:t}) * sc.T_b \triangleright$  Update ET, with equation (4) in sequential form
24:     $t \leftarrow t + sc.T_b$ 
25:  end while
26:   $FCV^t[MYOP] = \sum_{\tau^t \in G_\Omega} \prod_{u \in engaged(u,t,sc)} (1 - \lambda^{distance(\tau^t, s_u^t)}) \tilde{b}(\tau^t) \triangleright$  Calculate myopia reduction criterion
27:  return  $FCV^t, s_{1:U}^{start:t^{end}}, \tilde{b}(\tau^t)$ 
28: end procedure

```

ALGORITHM 3: sim&Eval: simulation and evaluation for MTS.

configuration variable. The initialization parameters $opt.init$ in step 4 allow, for each UAV and type of action, (1) selecting different bounds for the uniform distribution and (2) enabling/disabling its optimization. These options let our MTS planner generate trajectories with some fixed behavior (e.g., fixed height or fixed camera orientations). The stop condition parameter $opt.CompTime$ in step 9 allows fixing the computation time of the GA algorithm. The crossover parameter $opt.xover$ in step 11 allows selecting the percentage of parents that undergo the single-point crossover or are directly copied as children. Finally, the mutation parameter $opt.mut$ in step 12 allows selecting the Gaussian noise that is added to all variables in $a_{1:U}^{t:t+Q}$ and the Gaussian noise that is added to a few variables uniformly selected in $^{1:J} a_{1:U}^{t:t+Q}$ with a probability $1/|a_{1:U}^{t:t+Q}|$, where $|a_{1:U}^{t:t+Q}|$ is the number of decision variables in $a_{1:U}^{t:t+Q}$.

Besides, there are three steps of the GA that use the fitness criteria variable $^{1:J} FCV^{t+Q}$ to select solutions of the population to become pairs of parents (step 10), survive in the next generation (step 16), and become the best solution (step 19). Its behavior will be further detailed in the following section, after explaining the optimization criteria used in our MTS planner.

3.2.3. Simulation and Evaluation Process. In order to evaluate one of the solutions, subsequence of action commands $^J a_{1:U}^{t:t+Q}$,

manipulated and proposed by the GA, first, the motion model in Figure 5 or in the appendix has to be simulated for each UAV, and, second, the fitness criteria have to be evaluated. The planner has also to take into account the UAVs mission engaging time $[t_u^{start}, t_u^{end}] \forall u = \{1 : U\}$ and the possible differences among the time lapses of the problem (T_r, T_a, T_s, T_u^z). In the remainder of this section we describe the simulation process and the fitness criteria evaluation, and the steps of the $sim&Eval$ function, presented in Algorithm 3 and used by the GA in Algorithm 2.

The simulation of each UAV is performed in step 6 of Algorithm 3 with the $sim(\cdot)$ function, taking into account the Simulink or appendix model $f(s_u^t, a_u^t)$, the initial UAV location s_u^t , and the timely-equispaced set of actions in $a_u^{t:t+Q}$. Moreover, during the simulation each set of actions a_u^t proposed by the GA is considered constant during the time lapse T_a and the resulting UAV trajectory and sensor poses $s_u^{t:t+Q}$ are created with sampling simulation time of T_s . We use two different time lapses in the simulator (for T_s and T_a) to reduce the number of decision variables $a_u^{t:t+Q}$ manipulated by the planner, while permitting a high time resolution for evaluating certain aspects of the UAVs trajectories $s_{1:U}^{t:t+Q}$. Finally, it is worth noting that $sim(\cdot)$ also obtains, from the UAV motion model, the fuel consumption of the UAV during the corresponding trajectory subsection and checks if the UAV is engaged in the mission in order to only simulate its

displacement in the simulation time steps that simultaneously belong to $[t, t + Q]$ and $[t_u^{start}, t_u^{end}]$.

The fitness criteria of the MTS planner consists in fulfilling two constraints (UAV collision and nonflying zones avoidance) and in optimizing four indexes (command smoothness, fuel consumption, expected detection time, and myopia reduction criterion), whose values are stored in the different elements of the Fitness Criterion Vector FCV^t . Besides, in order to distinguish among the problem time periods T_a and T_s within the expressions of the evaluation criteria, we make $[t : \Delta : t + Q]$ stand for the time vector with elements equi-spaced by the selected Δ and $k \in [t : \Delta : t + Q] \wedge [a, b]$ stand for the time steps k within the set $[t : \Delta : t + Q]$ that fall within the limits $[a, b]$. Moreover, the *tag* used in Algorithm 3 within $FCV^t[tag]$ expressions permits the identification of the element of the vector FCV^t associated with each criterion. Finally, the details of each criterion are presented below:

- (i) *UAV collision avoidance*. This criterion counts the number of trajectory time instants at which the distance between the locations of all pairs of UAVs is smaller than the security collision distance d_{COL} . The criterion, evaluated in line 10 of Algorithm and detailed in (12), consists in checking the inter-UAV distances at time steps where both vehicles are engaged in the mission.

$$\begin{aligned} & collision(sc, s_u^{t:t+Q}, s_l^{t:t+Q}) \\ &= \sum_{k \in [t:T_s:t+Q] \wedge [sc.t^{start}, sc.t^{end}] \wedge [sc.t^{start}, sc.t^{end}]} (x_u^k - x_l^k)^2 \\ &+ (y_u^k - y_l^k)^2 + (h_u^k - h_l^k)^2 \leq (sc.d_{COL})^2 \end{aligned} \quad (12)$$

- (ii) *Nonflying zones (NFZ) avoidance*. This criterion counts the number of trajectory time instants at which the locations of any UAV overfly a selected set of forbidden overflying cells of G_Ω . Again, the criterion, evaluated in line 11 of Algorithm 3 and detailed in (13), considers locations at the time steps where the UAVs are in the mission.

$$\begin{aligned} & withinNFZ(sc, s_u^{t:t+Q}) \\ &= \sum_{k \in [t:T_s:t+Q] \wedge [sc.t^{start}, sc.t^{end}]} inNFZ(x_u^k, y_u^k) \end{aligned} \quad (13)$$

- (iii) *Actions smoothness*. This criterion, evaluated in line 12 of Algorithm 3, measures the smoothness of the commands for each UAV within $a_u^{t:t+Q}$. It accounts, with (14) and (15), for the consecutive changes of the five types of commands only during the time interval where the UAVs are engaged in the mission. The magnitude of the accounted change, calculated with (16), makes the smooth criterion only account for those changes in action that imply a modification of sign in the signal with a value proportional to the

square of the difference. Therefore, the smoothness criterion penalizes big changes of values and sign in the sequence of each type of command.

$$\begin{aligned} & smooth(sc, a_u^{t:t+Q}) \\ &= \sum_{k \in [t:T_a:t+Q-T_a] \wedge [sc.t^{start}, sc.t^{end}]} changeA(a_u^k, a_u^{k+T_a}) \end{aligned} \quad (14)$$

$$\begin{aligned} & changeA(a_u^t, a_u^k) \\ &= ch(\theta_u^{c,t}, \theta_u^{c,k}) + ch(h_u^{c,t}, h_u^{c,k}) + ch(v_u^{c,t}, v_u^{c,k}) \\ &+ ch(\eta_u^{c,t}, \eta_u^{c,k}) + ch(\alpha_u^{c,t}, \alpha_u^{c,k}) \end{aligned} \quad (15)$$

$$ch(a, b) = (a - b)^2 (|\text{sign}(a) - \text{sign}(b)| == 2) \quad (16)$$

- (iv) *Fuel consumption*. Its value is directly returned by the UAV motion model and affected by the UAV speed.

- (v) *Expected time of detection*. The main objective of the planner, described already in the general problem formulation section, is implemented in an iterative form within lines 16 and 25 of Algorithm 3. The process implements (2)-(4) after the following modifications. On the one hand, the expression in line 19 includes the target motion model transition operation for the time steps where the motion model has to be applied (i.e., when $motionTarget(t, T_\tau)$ is true, because the target is dynamic and the last target motion update was performed T_τ seconds ago). On the other hand, the ‘‘unnormalized belief’’ update expression in line 21 only includes the likelihood models of the UAVs at the time steps t where the UAV is engaged in the mission $[t_u^{start}, t_u^{end}]$ and its sensor has to take a measure (according to its T_u^z).

- (vi) *Myopia reduction criterion*. The planner can create myopic solutions due to its multistep behavior, which only considers the probability regions achievable from the starting position of each section of the trajectory. In other words, by optimizing only the previous criteria, the planner becomes greedy, ignoring during the optimization of each trajectory section the benefits of getting closer to a high probability area which is currently unreachable. To reduce this myopic behavior, the planner evaluates, in line 26 of Algorithm 3, the suitability of the final UAVs positions and sensor poses $s_{1:U}^{rQ}$ of a trajectory section for the optimization of future trajectory sections by weighting the remaining unnormalized belief $\tilde{b}(\tau^{rQ})$ with a function that takes into account the distance of each cell τ^t in G_Ω to the center of the camera footprint associated with the final location and camera pose of each UAV s_u^t . Besides, this evaluation is only performed for the UAVs that are still engaged in the mission. In other words, the planner evaluates the suitability of the final camera footprints of the searching UAVs for optimizing the remaining sections of the trajectories.

As we have previously mentioned, the FCV^t values obtained after the simulation of each trajectory section are later used in 3 different steps of the GA. The way of proceeding in each step is the following:

- (i) The binary tournament selection operator chooses randomly, with an uniform probability, two solutions j of the population; among them it selects as a parent the one with lower constraint violation (i.e., with lower ${}^jFCV^t[\text{COL}] + {}^jFCV^t[\text{NFZ}]$), and when tied, the one that Pareto-dominates the other four optimization objectives (i.e., ${}^jFCV^t[\text{SMOOTH}]$, ${}^jFCV^t[\text{FUEL}]$, ${}^jFCV^t[\text{ETD}]$, ${}^jFCV^t[\text{MYOP}]$).
- (ii) The survival/recombination operator organizes the solutions in a similar way: first according to its constraint violation level and second according to the Pareto-dominance of the remaining four optimization objectives. Hence, as a whole, the GA tries to identify the solution that violates the constraints less times and, within them, those solutions that Pareto-dominate the other objective criteria.
- (iii) The selection of a unique best solution is required at the final step of the GA to let the multisteped planner use its final position as the starting point of next trajectory section. To do this, the GA selects, among the solutions in the best Pareto-front of the last population obtained by the recombination operator, the one with smallest rounded value of myopia reduction criterion (first), smallest ETD (second, if tied in the first), smallest command smoothness (third, if tied in the previous), and small fuel consumption (fourth). The priority levels used for the selections process reflect the importance order of the objectives. Moreover, the myopia criterion is preferred to ETD, because $FCV^t[\text{MYOP}]$ accounts both for the probability gathered by the trajectory (through $\tilde{b}(\tau^t)$) and for the distance to all the cells with remaining “unnormalized belief”, while $FCV^t[\text{ETD}]$ only considers the reachable probability within each trajectory section $s_{1:U}^{t:t+Q}$.

3.3. Computational Cost of the MTS Planner. Finally, we calculate the computational complexity of our planner. To do this, we will consider that all UAVs start and end the mission at the same time steps ($t_{1:U}^{\text{start}} = 0$, $t_{1:U}^{\text{end}} = t^{\text{end}}$) and that the values of quotients Q/T_s , Q/T_a , Q/T_τ , Q/T_u^z , t^{end}/Q are natural numbers.

We start studying Algorithm 3 and estimate the computation complexity of the UAV and sensor motion simulation (cc_{sim}) and of the evaluation of each fitness criterion (cc_{tag}). The computational cost of the UAV and sensor motion simulation is proportional to the number of UAVs U and to the steps Q/T_s in which the simulator has to return UAV and sensor poses, $cc_{\text{sim}} = O(U \cdot Q/T_s)$. The computation cost of the collision avoidance criterion is proportional to the number of possible UAV combinations $U(U - 1)$ and to the steps Q/T_s in which the UAV trajectory and sensor poses have been discretized. Hence, $cc_{\text{COL}} = O(U^2 \cdot Q/T_s)$. For the NFZ

criterion, we can follow a similar process, but accounting only for the number of UAVs. Hence $cc_{\text{NFZ}} = O(U \cdot Q/T_s)$. In the case of the control smoothness, we have to consider the number of UAVs U , the number of steps Q/T_a in which the action sequence is discretized, and the number of types of actions that are being optimized A . Hence $cc_{\text{SMOOTH}} = O(U \cdot A \cdot Q/T_a)$. As the fuel consumption is calculated during the simulation of the UAV and sensor motion, its computational complexity is already included in cc_{sim} . To estimate the complexity associated with the ETD computation, we can observe that the cost associated with the target motion is proportional to Q/T_τ and $(w_x w_y)^2$, and cost associated with the sensor update is proportional to $(w_x w_y) \sum_{u=1}^U Q/T_u^z$. The first cost is associated to the operations in the target transition $\sum_{\tau^t - T_\tau \in G_0} P(\tau^t \mid \tau^{t-T_\tau}) \tilde{b}(\tau^{t-T_\tau})$, which can be calculated, a number of Q/T_τ times, as the multiplication of two matrixes of $(w_x w_y) \times (w_x w_y)$ and $(w_x w_y) \times 1$ size. The second cost is associated to the update of each sensor $P(z_u^t \mid \tau^t, s_u^t) \tilde{b}(\tau^t)$, which is calculated a number of Q/T_u^z times and is proportional to $w_x w_y$. Hence, $cc_{\text{ETD}} = O((w_x w_y)^2 Q/T_\tau + (w_x w_y) \sum_{u=1}^U Q/T_u^z)$. Finally, as the computational cost of the myopia reduction criterion is proportional to U and to $w_x w_y$, its complexity $cc_{\text{MYOP}} = O(U \cdot w_x w_y)$. The addition of the previous complexities, after simplifying the final expression, makes the expression of the computational complexity of the simulation and evaluation function become (17).

$$cc_{\text{sim\&Eval}} = O\left(U^2 \frac{Q}{T_s} + U \cdot A \cdot \frac{Q}{T_a} + (w_x w_y)^2 \frac{Q}{T_\tau} + (w_x w_y) \left(U + \sum_{u=1}^U \frac{Q}{T_u^z} \right) \right) \quad (17)$$

The expression shows that the complexity of the simulation and evaluation function $cc_{\text{sim\&Eval}}$ grows with the increment of the duration Q of each evaluation, the number of UAVs U , the number of decision variables manipulated by the planner A , and the number of cells of the grid of the search space $w_x w_y$. Besides, $cc_{\text{sim\&Eval}}$ also grows as the time steps of the simulation T_s , of the actions T_a , of the target T_τ , and of the camera measurements T_u^z are decremented. Hence, in order to have a reasonable computational cost, it is necessary to achieve a compromise among all those parameters. Besides, the quadratic dependency of $cc_{\text{sim\&Eval}}$ with $w_x w_y$ (i.e., with the size of the grid of the search space) is worth noting. This quadratic cost is especially relevant, because a moderate size search space (e.g., of 30 x 30 cells in the Areia Branca example) has already significantly bigger values of $w_x \cdot w_y$ (e.g., 900) than those used in other parameters (e.g., $U = 2$, $A = 3$, $Q/T_s = 50$, $Q/T_a = Q/T_u^z = 5$, $Q/T_\tau = 1$).

To account for the computational complexity of the planner, we still have to consider the computational cost of the GA in Algorithm 2 and of the multisteped approach in Algorithm 1. For the GA, we compute the cost of the main loop of Algorithm 2. Considering a population size of J solutions, $U \cdot A \cdot Q/T_a$ decision variables, and M iterations of the loop, the cost of the $cc_{\text{GA}_{\text{selfparent}}} = O(M \cdot J)$, $cc_{\text{GA}_{\text{xover}}} =$

$cc_{GA_{mut}} = O(M \cdot J \cdot U \cdot A \cdot Q/T_a)$, $cc_{GA_{sim\&Eval}} = M \cdot J \cdot cc_{sim\&Eval}$, and $cc_{GA_{recombine}} = O(MJ^2)$. This happens because computational cost of the tournament selection of each pair of parents is proportional to a constant, the cost of the single-point crossover and of the incremental mutation of a solution is proportional to the number of decision variables within it, and the cost of the recombination step of the population, which includes the sorting of the solutions, is proportional to J^2 . Hence, the overall GA complexity can be calculated as $cc_{GA} = M \cdot J(cc_{sim\&Eval} + O(U \cdot A \cdot Q/T_a + J))$. Finally, as the multisteped GA-based planner in Algorithm 1 calls the GA $R = t^{end}/Q$ times, its overall computation complexity becomes $cc_{MS} = t^{end}/Q \cdot M \cdot J(cc_{sim\&Eval} + O(U \cdot A \cdot Q/T_a + J))$, calculable with (18), obtained by substituting within it $cc_{sim\&Eval}$ by (17).

$$cc_{MS} = O\left(\frac{t^{end}}{Q} \cdot M \cdot J \left(J + U^2 \frac{Q}{T_s} + U \cdot A \cdot \frac{Q}{T_a} + (w_x w_y)^2 \frac{Q}{T_\tau} + (w_x w_y) \left(U + \sum_{u=1}^U \frac{Q}{T_u^z} \right) \right) \right) \quad (18)$$

The overall computational complexity shows the same dependencies observed for the computational complexity of the simulation and evaluation and a (multiplying) growing behavior proportional to the population size J , the number of generations in GA M , and the number of subsections in which the trajectory is divided t^{end}/Q . Besides, as Q appears in the denominator of t^{end}/Q that multiplies the numerator of different quotients with the time steps, there is a part of the computational cost that can be considered independent of Q and of the number of sections in $R = t^{end}/Q$ in which the trajectories are divided, and dependent on the quotient of the duration of the mission t^{end} and the different time steps (T_s , T_a , T_τ , and T_u^z).

Finally, it is worth noting that the complexity analysis, which is useful to consider the effects of selecting certain variables of the system in the computation cost, does not provide information about (1) the time required for evaluating the solutions or (2) the convergence speed of the multisteped planner and of each GA run, whose behavior is highly dependent on the probability models (initial belief, target motion model, and sensor detection function) of each scenario. To analyze the convergence speed, in the Results section we will analyze the temporal performance of our new planner (implemented in Matlab and using its Parallel Toolbox in the evaluation loops, lines 5-7 and 13-15, of the GA algorithm) over different scenarios to be able to determine, empirically, the time needed to optimize them (in a 2.5 GHz Intel Core i7 with 6 GB RAM PC with Windows 7). To provide some insight related to the time required for evaluating the solutions within the planner, we have measured the values of the different steps of the GA algorithm in the scenarios of the following section and observed that the mean computation time of each population step of the GA (loop in line 9 of Algorithm 2) is 0.46 seconds for the first scenario, 0.49 seconds for the second one, and 1.05 seconds for the third one; that the evaluation of the solutions (**sim&Eval** function)

takes 99% of the time; and that the UAV motion simulation takes 43% of the time of the **sim&Eval** function, while the rest of it takes 53%.

4. Related Work

This section compares the new planner presented in this paper with existing approaches and formulations of PTSP. Instead of performing an exhaustive state-of-the-art, we present a compilation of the works that have motivated this paper. In detail, we focus on two types of approaches: methods that optimize the ETD or other criteria where the visiting order of the cells is important [4–10], or more generic PTSP approaches that obtain either smooth trajectories of the UAVs [11–17, 20] or that also optimize the sensor orientation [21, 22]. Hence, we leave out of this comparison other works such as [18, 35–42] that do not share those properties. The comparison is summarized in Table 1, where the first column identifies each work and the remaining ones show the properties highlighted in the comparison, and where the works are grouped row-wise by the type of criteria they optimize (which is shown in the tenth column). In more detail:

Target column shows which approaches deal with **multiple** and/or **moving** targets. Although including both properties permits handling a wider range of scenarios, it is worth noting that none of the ETD strategies [5–7, 9, 10], at the bottom of the table, can still handle multiple targets.

UAVs column shows which planners are developed for **multiple** UAVs and their type of **dynamic model**, grouped under the following labels: *cell-to-cell* indicates that the UAVs move from one cell of the belief to another cell according to a finite set of cardinal directions, *waypoint* indicates that the UAVs move according to segments defined by the waypoints selected by the planner, *simplified* indicates that the UAV motion model $\dot{s}_u^t = f(s_u^t, a_u^t)$ is linearized or consists in a constant speed steering model, and *complex* indicates that the planner works with a Simulink or differential motion model as ours. The benefits of multi-UAV approaches with complex UAV motion models are the possibilities of, on the one hand, observing different regions of the space simultaneously and, on the other hand, adapting the UAVs trajectories to the actual maneuvering capabilities of the UAVs.

Sensor column shows the **type** of sensor, its observation **range**, and its **efficiency**, as well as the planner capability of **moving** it. The algorithms under comparison use mainly downward-looking (*DL*) cameras and radars as well as generic sensors, whose behavior is identifiable by the labels in its observation range (a single *cell*, a cell of the *quadtrees* used to store the belief, and a *limited* or *wide* group of cells) and efficiency (*ideal* when the likelihood is always either 1 or 0, and *FN* and *FP* if it, respectively, considers false negatives and false positives). It is also worth noting that *wide* range sensors are often modeled with

TABLE 1: Works comparison.

Work	Targets		UAVs		Sensor			Optimization		
	Multiple	Moving	Multiple	Dynamic Model	Type	Range	Efficiency	Moving	Criteria	Approach
[11]	✓	✓	✓	simplified	radar (DL)	wide	FN		$\sum_k w_k P_d^k$	SQP
[12]		✓		simplified	radar (DL)	wide	FN		P_d	Greedy
[13]			✓	simplified	camera (DL)	limited	FN		P_d	Gradient
[14]		✓	✓	simplified	camera (DL)	limited	FN		P_d	Max-sum
[15]			✓	simplified	generic	wide	FN		P_{nd}	Gradient
[16]	✓		✓	simplified	generic	wide	FN		$\prod_{k=1}^Q P_{nd}^k$	Gradient
[17]	✓	✓	✓	waypoints	camera (DL)	limited	FN/FP		IG	Gradient
[19]	✓		✓	simplified	camera (DL)	limited	FN/FP		Uncertainty	Gradient
[20]				waypoints	generic	quadtree	FN/FP		$IG + dist.$	Greedy
[21]	✓	✓	✓	waypoints	camera	limited	FN	✓	$P_d/dist.$	Graph
[22]	✓	✓	✓	simplified	camera	limited	FN	✓	Discounted Uncertainty	MAB
[4]		✓	✓	cell-to-cell	generic	cell	ideal		Discounted P_d	CEO
[5]		✓	✓	cell-to-cell	generic	cell	ideal		ETD	BOA
[6]		✓	✓	cell-to-cell	generic	cell	ideal		ETD	ACO _{MMAS}
[7]		✓	✓	cell-to-cell	radar, sonar, camera				ETD	CEO
[8]				cell-to-cell	generic	cell	ideal		MCSTD	Greedy
[9]		✓	✓	complex	radar (DL)	wide	FN/FP		ETD with constraints	ACO _R
[10]			✓	complex	radar (DL)	wide	FN/FP		ETD and others	GA
This work		✓	✓	complex	camera	limited	FN	✓	ETD and others	GA

decaying likelihood functions with the distance [9–12, 16], while the remaining sensors are often modeled with constant likelihood functions [4–6, 8, 13, 19, 21]. Besides, although camera sensors are used in several works [13, 14, 17, 21, 22], only a few approaches (including the one presented in this paper) reorient them to observe different regions of the search space easier.

Optimization column shows the **criteria** and the **approach** used to improve them within each planner.

On the one hand, the optimization criteria are used to organize the works row-wise in four groups: optimizing the probability of detection (maximizing P_d or minimizing its counterpart P_{nd}), maximizing the information gain (IG), considering the trajectory distance or the time (by including a time discounting factor) in previous TPSP criteria, and minimizing the expected time of detection (ETD or the Monte Carlo Simulation Time of Detection MCSTD). Besides, in the table, the notation is extended with a right superindex in P_d^k and P_{nd}^k in order to represent the corresponding probability of each (k) target (each with an initial belief and target motion model independent of the others). And to evaluate MCST for a given $s_{iu}^{0:t_{iu}^{end}}$, several Monte Carlo simulations are run (each one

consisting in sampling the target initial position τ^0 from the initial belief $b(\tau^0)$ and in measuring the time step of the trajectory where $s_{iu}^{0:t_{iu}^{end}}$ overflies the sample τ^0) and the average value of the target detection times of all the simulation is calculated. In this regard it is worth noting that optimizing either P_d , P_{nd} , IG, or the uncertainty by itself is not the best option for MTS, where it is necessary to collect the maximum probability of detection as soon as possible. Besides, although including the distance (and hence making the approaches prefer solutions that optimize the PTSP criterion with shorter trajectories) or the time (by including a discounting factor that gives higher importance to the optimization of the criterion in the initial steps) in the timeless objectives alleviates the problem, those solutions [4, 20–22] are biased by the strategies and the parameters used to include them. Hence, they do not optimize ETD as well as the approaches in the last group [5] do.

On the other hand, the optimization approach column shows the variability of methods used to handle the high complexity of PTSP, ranging from *greedy* or *local* approaches to *gradient* or *graph* based techniques, *max-sum* algorithms, Sequential Quadratic Programming (QSP), Multiarmed Bandit algorithms

(MAB), Cross-Entropy Optimization (CEO), Ant Colony Optimization (ACO), or Genetic Algorithms (GA). In this regard it is worth noting that the selected approach is often associated with the underlying type of decision variables (discrete when the UAV dynamic model is cell-to-cell, continuous otherwise), the type and number of criteria functions, and the authors expertise. The main benefit of the selected GA optimizer, inherited from the planner in [10], comes from its capabilities, tested by many researchers over different real-world problems, to handle many-decision multicriteria optimization problems. In this regard, it is worth noting that the selection of a GA has been especially useful to add several constraints and optimization criteria to ETD and to incorporate the five types of decision variables (actions) in the new planner.

If we analyze the table in detail, we can see that this work is the most general within the ETD group, as it allows the planner to obtain the best camera orientations for the best UAV trajectory, deal with moving target scenarios, and optimize multiple criteria. Moreover, the optimization approach in [8], which evaluates the expected time of detection via MCSTD, is directly targeted at initial Gaussian beliefs, while ours is targeted and tested against generic initial beliefs. Regarding the UAV motion model and the quantity of optimization criteria, this work is also more general than the ones classified in the other three groups. Its main drawback, inherited from existing ETD planners, is its single-target support. However, as we will show in one of the examples, this limitation can be shortened by using as initial belief the joint belief of the targets and making the UAVs continue the search mission until all the targets are found.

Besides, a more detailed comparison with its closest MTS state-of-the-art planners ([9, 10]) shows the following facts. First, the new planner optimizes five action types (UAV orientation, speed, and height; sensor azimuth and elevation) while the planners in [9, 10] only optimize the UAV orientation. Moreover, as the good performance of the ACO-based planner in [9] is supported by the use of a heuristic for generating promising UAV headings, its extension is not straightforward as it will require building heuristics for generating promising UAV speeds and heights, and promising camera orientations. Second, the camera introduced in this paper challenges the planner as its limited range likelihood enhances the stepped and multimodal behavior of $ETD(s_{1:U}^{0:t^{end}})$, while the soft decaying wide range likelihood of the radars in [9, 10] smooths $ETD(s_{1:U}^{0:t^{end}})$, helping those planners. Finally, although the new planner and the one in [10] share the multisteped GA-optimization support, UAV motion model, constraints, and objective functions, the new planner (and evaluation functions) extends the capabilities of the planner in [10] by allowing obtaining solutions (UAV trajectories plus sensor poses) for scenarios with dynamic targets, using five different types of actions, with different UAV motion model parameterizations, and with different UAV mission engaging and camera-capture times.

An additional contribution of this work to MTS search, not included in the table due to space limitation, is the methodology followed to build realistic scenarios for testing the planner. Moreover, the next section details how to set up the scenario correctly, as there is a strong influence of some parameters (e.g., the size of the target, the camera FOV, and the height and size of the search region) on others (such as the recommended UAV flying altitude, camera likelihood, and search cell size) and, hence, on the planner performance.

Finally, for comparison purposes with existing approaches, we can set up the new planner with the capabilities of the one presented in [10], as it is a direct extension of it and the new planner allows selecting which decision variables have to be optimized. We do not compare against the other works of the ETD group, as the planner in [9] is not capable of optimizing all the criteria used in this work, and the remaining ones [5–8] are designed for UAV motion models that move from cell to cell and for optimizing a single criterion.

5. Results

In this section the new planner is tested over three real-world inspired scenarios. We first explain their common properties and the analysis setup, and afterwards we detail the individual characteristics of each scenario and the solutions obtained by our planner under two different configurations: optimizing the UAVs trajectories and sensor poses, and only optimizing the UAV orientation (as in [10]). Therefore, this second configuration is used as a baseline to compare against and observe the benefits of this work proposal of optimizing multiple types of actions (including the sensor poses).

5.1. Common Scenario and Planner Setup. On the one hand, to study the benefits of optimizing multiple decision variables, we make the GA of our planner optimize: (1) only the commanded UAV heading $\theta_u^{c,t}$ (as in the previous planner in [10]) or (2) the commanded UAV heading $\theta_u^{c,t}$, UAV speed $s_u^{c,t}$, and camera azimuth $\alpha_u^{c,t}$. For shortening, hereafter the first configuration will be labeled as GA_θ and the second as $GA_{\theta,v,\alpha}$. Besides, in this work, we prefix the values of the remaining decision variables (commanded UAV height $h_u^{c,t}$ and commanded camera elevation $\eta_u^{c,t}$) to facilitate the analysis of the results. Finally, for comparison purposes, in GA_θ we prefix the UAV speed and camera azimuth (i.e., the two extra decision variables of $GA_{\theta,v,\alpha}$) to the mean value of the permitted ranges in $GA_{\theta,v,\alpha}$.

On the other hand, and due to the stochastic nature of a GA, we run each planner configuration 20 times and represent the evolution (against the computation time) of the mean and variance of the $ETD(s^{0:t^{end}})$ obtained at different runs. Note that we only show the ETD as it is in fact the most relevant objective for this planner. Besides, as $ETD(s^{0:t^{end}})$ is not available until the last section of the trajectory is being optimized, for the computation times where the multisteped planner has not yet started the final section optimization, we approximate the value of $ETD(s^{0:t^{end}})$ with the following expression, which accounts for the ETD up to the section that

is being optimized plus the worst case (associated with the case in which the UAVs do not collect more beliefs after $s_{1:U}^r$) for the remaining time ($t^{end} - (rQ + T_r)$).

$$\text{ETD}_a \left(s_{1:U}^{0:t^{end}} \right) \approx \text{ETD} \left(s_{1:U}^{0:rQ} \right) + \left(t^{end} - (rQ + T_r) \right) \cdot P_{nd} \left(s_{1:U}^{0:rQ} \right) \quad (19)$$

It is worth noting that the approximation in (19) induces the stepped behavior of the ETD curves presented in the paper in Figures 7(d), 9(d), and 10(d). This happens because the contribution of the worst case term in the ETD_a , which is associated with the belief that can/has not yet been collected by the UAVs, is usually decremented after a new section of the trajectory is being optimized, because as soon as the new probability of nondetection $P_{nd}(s_{1:U}^{0:(r+1)Q})$ is lower than the one of the previous section $P_{nd}(s_{1:U}^{0:rQ})$, there is at least a decrement of $(t^{end} - (Q(r+1) + T_r))(P_{nd}(s_{1:U}^{0:rQ}) - P_{nd}(s_{1:U}^{0:(r+1)Q}))$ in ETD_a . Moreover, in the initial sections, where r is lower, for a similar reduction in the nondetection probability, the decrement in the represented value of the ETD_a is often bigger.

The results of the planner for each scenario are, respectively, schematized in Figures 7, 9, and 10. In all of them, the graphic at the left-top (a) represents the initial $b(\tau^0)$ and UAVs initial locations (colored airplanes), the graphic underneath (d) shows the evolution of the $\text{ETD}_a(s_{1:U}^{0:t^{end}})$ mean and standard deviation, and the second and third columns display additional information of a representative solution obtained by each planner configuration (only heading in the second column, and heading + speed + azimuth in the third one). In more detail, the top row graphics of the second and third column show the $\tilde{b}(\tau^{end})$, $\text{ETD}(s_{1:U}^{0:t^{end}})$, and the probability of detection $P_d(s_{1:U}^{0:t^{end}})$ of a representative solutions of the corresponding planner configuration, while the graphics at the bottom row also represent the UAVs trajectories (with the red and blue lines), the camera footprints (with the shadowed red and blue polygons), and the camera aiming point without terrain elevation (by the end of the yellow line that starts at the UAV location of each measurement).

Additionally, the elevation information used to construct the initial beliefs for each scenario is available in [43]. Besides, the wind speed $u_w^t = 0$ in all the scenarios and all the cameras has a resolution of 2160x3840 pixels and a fixed elevation angle of 45°. Moreover, the camera FOV and UAV altitude $h_u^{c,t}$ are precalculated and maintained constant for each mission to ensure a high detection likelihood during the whole search. To achieve this, the FOV and $h_u^{c,t}$ are adjusted to make acceptable the value of TTPF for the worst possible case, which occurs at the highest distance between the target and camera and, hence, at the lowest terrain elevation. Additionally, the size of the cells in G_Ω is calculated for each mission to ensure that the footprint covers more than one cell in the worst case (at the highest terrain elevation). Finally, all missions share the frequency of the camera captures and command changes ($T_u^z = T_a = 10$ s), the simulation step

($T_s = 1$ s), the duration of the trajectory sections ($Q = 50$ s), and the collision security distance ($d_{COL} = 100$ m).

Lastly, we inherit several values of the planner parameters from [10]. The population size $J = 50$, the probability of crossover is 0.8, the probability of mutation $1/(\sum_{u=1:U} N_u)$, and the high-lower additive mutation standard deviation is 1.0-0.1. Besides, the myopia parameter $\lambda = 0.95$.

5.2. Looking for a Static Off-Road Vehicle in the Mountains.

The target of the first scenario (proposed by Airbus) is an off-road vehicle broken down in Gador mountains (Spain), within a search area of 10x10km² whose elevation varies, according to the elevation map in Figure 6(a), from 626 to 2037m. The elevation probability layer, represented at the bottom of Figure 6(b), is obtained by dividing the elevation range into four fractions and assigning lower probabilities to higher elevations. The intelligence layer, displayed at the top of Figure 6(b), consists in two Gaussians centered at the valleys oftener visited by the vehicle. The initial belief, displayed in Figure 6(c), is obtained merging both layers with $w_1 = 15$ and $w_2 = 1$, since this weights combination makes the two Gaussians be only 1/16 (i.e., 6.25%) of $b(\tau^0)$. Besides, the only UAV engaged in this MTS operation, from $t_1^{start} = 0$ to $t_1^{end} = 450$ s, starts the mission at the right top corner of the scenario and can fly at $h_1^t \in [500, 3000]$ m (with respect to the sea level) and $v_1^t \in [30, 60]$ m/s, with camera azimuth $\alpha_1^t \in [-30^\circ, 30^\circ]$. We select a camera FOV=25° and UAV $h_1^{c,t} = 3000$ m to ensure that the sensor likelihood varies, according to the elevations in Ω and target size ($S_{target} \approx 5$ m), from 0.90 to 0.99. Besides, as the estimated smallest complete footprint, according to the UAV constant height and maximum terrain elevation, is 590x482m², we divide Ω into a grid of 30x30 cells, each of 333x333m².

The results of running both configurations of the planner for this scenario are presented in Figure 7. The graphic of the ETD evolution in Figure 7(d), which represents the approximated ETD_a up to the optimization of the last section of the trajectory, clearly shows the stepped behavior associated with the inclusion of new trajectory sections $s_{1:U}^{Q(r-1):Qr}$ in the optimization of the planner. It is worth noting that the duration of the ETD steps can be easily clipped reducing the computational time of the GA stop condition, as each GA run within the multisteped planner converges to its ETD plateau long before its designated time has finished. Besides, the ETD evolution graphic also shows that the $\text{GA}_{\theta,v,\alpha}$ produces better results (with lower ETD) during all the optimization processes than GA_θ .

Furthermore, if we analyze the additional information of the representative solutions of both configurations, presented in Figures 7(b), 7(c), 7(e), and 7(f), we can observe that $\text{GA}_{\theta,v,\alpha}$ permits the UAV + camera to collect more $P_d(s_{1:U}^{0:t^{end}})$ than GA_θ (51% vs. 43%), because the UAV adapts its speed to collect slowly the two Gaussians and arrive quicker to the southeast region of Ω . Besides, the footprints obtained with $\text{GA}_{\theta,v,\alpha}$ are more scattered than those obtained by GA_θ , due to the possibility of changing the camera azimuth. Finally, it is worth noting how the shape of the footprints changes and gets closer to the camera aiming point at zero elevation in

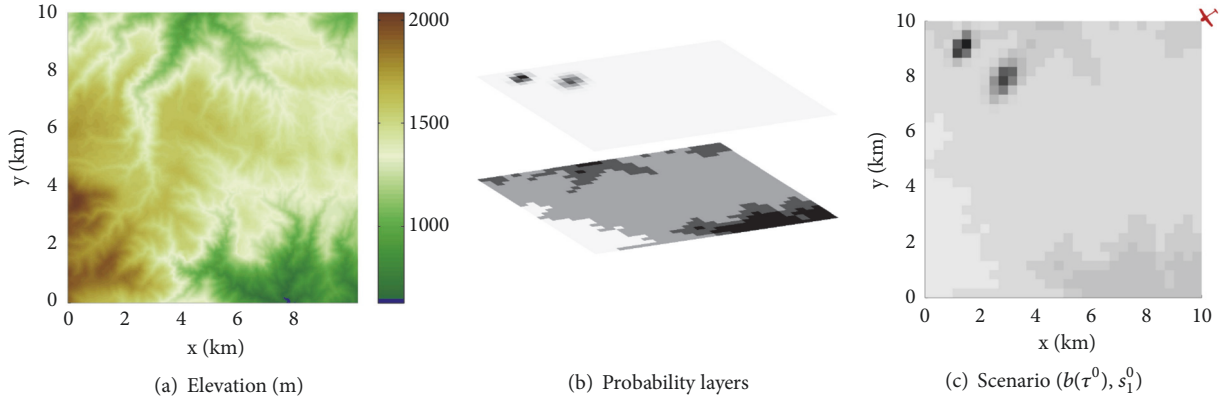


FIGURE 6: Scenario for searching for a lost vehicle in Gador mountains (Almeria, Spain).

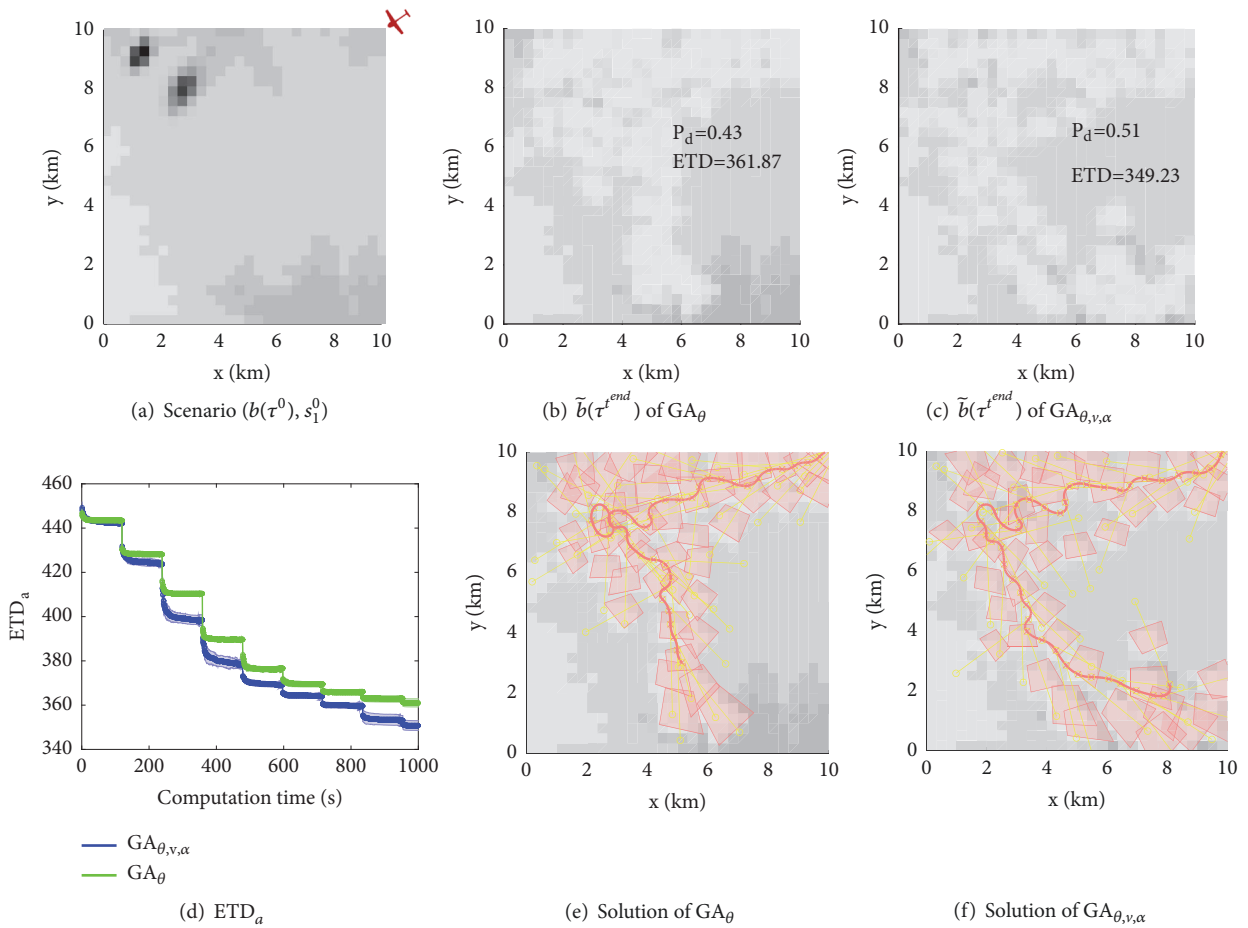


FIGURE 7: Results for searching for a lost vehicle in Gador mountains (Almeria, Spain).

the valleys (at the north and at the south west) than in the mountains.

5.3. *Looking for Survivors after an Earthquake.* In the second scenario, inspired by the earthquake of Haiti in 2010, we search survivors at Carrefour (Port-au-Prince) in a rectangular area of $10 \times 7 \text{ km}^2$, whose elevation varies, according to the elevation map in Figure 8(a), from sea level to 875m.

To use our single-target search planner for a multitarget scenario, we only have to make the initial belief of our scenario the joint-probability of all the survivors and make the UAVs perform the search until all the targets are found [16]. The elevation probability layer, represented at the bottom of Figure 8(b), is defined by dividing the elevation range into three fractions and assigning 0 probability of survivors presence at the sea, high probability at the coast, and lower

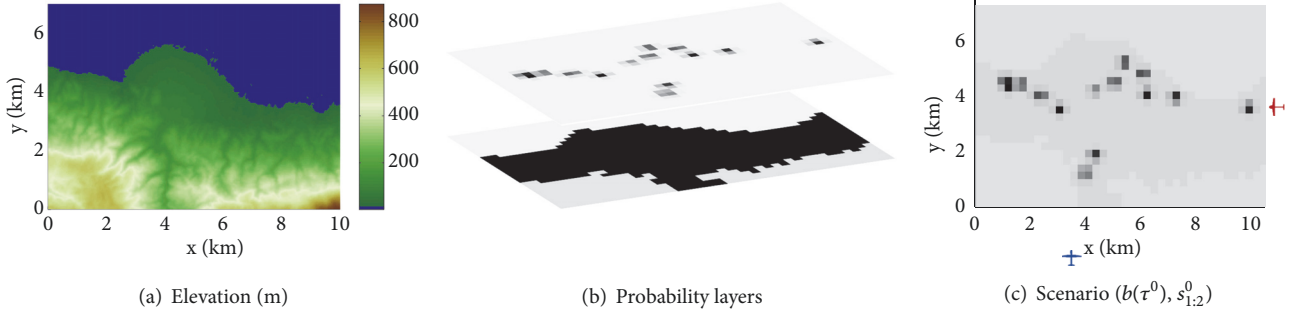


FIGURE 8: Scenario of searching for survivors after an earthquake in Carrefour (Port-au-Prince, Haiti).

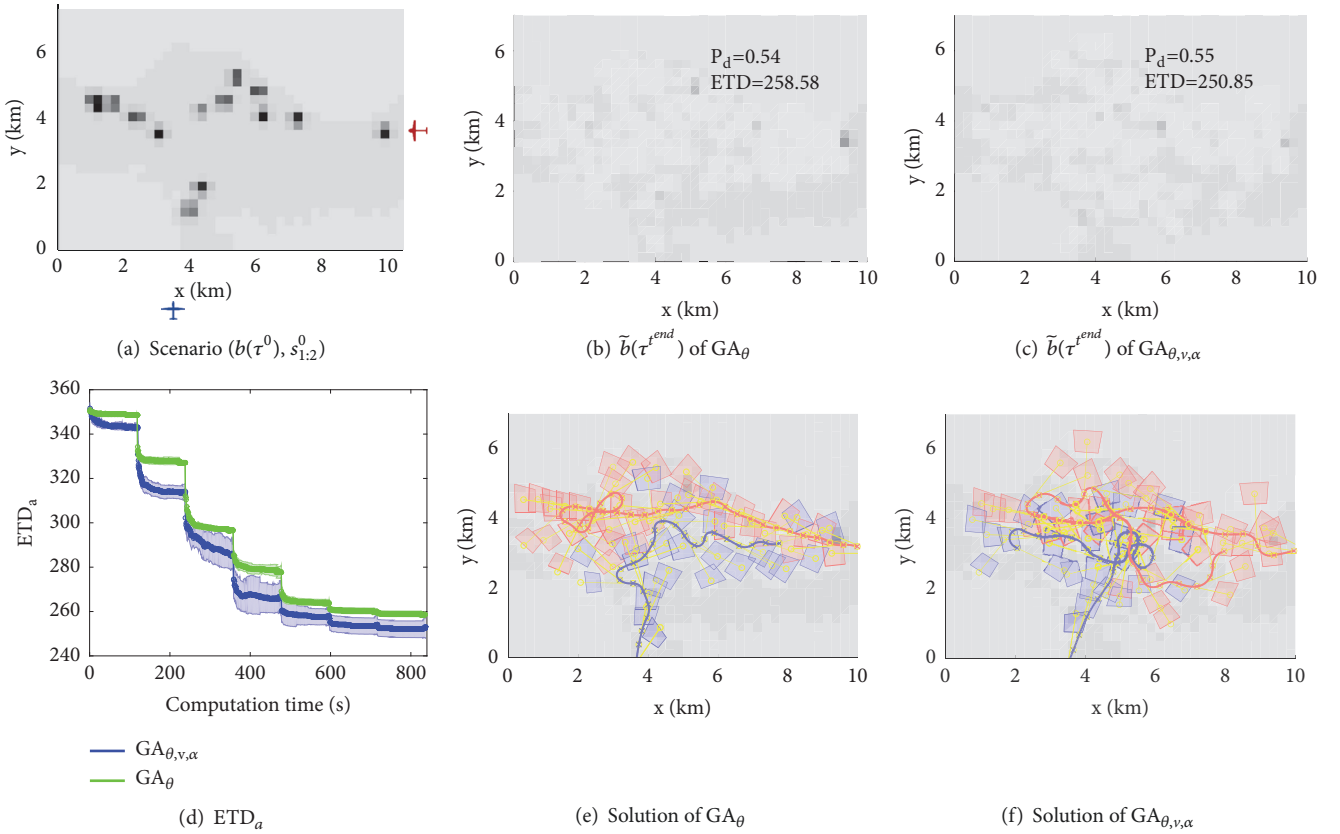


FIGURE 9: Results of searching for survivors after an earthquake in Carrefour (Port-au-Prince, Haiti).

probability to elevations bigger than 400m. The intelligence layer, represented at the top of Figure 8(b), indicates the probability of survivors based on building damage assessment. This information, available in [44] and used in [45] for another search problem, was obtained by several public and private entities, comparing previous and a posteriori aerial imagery of the earthquake area. The initial belief, represented in Figure 8(c), is obtained merging both layers with $w_1 = 1$ and $w_2 = 1$, in order to make the intelligence information represent 1/2 (i.e., 50%) of $b(\tau^0)$. The first UAV (in red in Figure 8(c)) arrives at the search area Ω from the East at $t_1^{start} = 0s$, while the second UAV (in blue) does it from the south at $t_2^{start} = 50s$. Both UAVs remain in the mission

up to $t_{1,2}^{end} = 350s$ and can fly at $h_{1,2}^t \in [500, 3000]m$ and $v_{1,2}^t \in [30, 60]m/s$, with camera azimuth $\alpha_{1,2}^t \in [-30^\circ, 30^\circ]$. However, due to the survivors size ($S_{target} \approx 2m$), both UAVs are forced to fly at lower altitudes $h_{1,2}^{c,t} = 1500m$ with a smaller FOV=20° than in the first scenario, in order to ensure an acceptable camera likelihood that varies from 0.82 to 0.99 (respectively, at the lowest and highest elevations of the search region). Besides, as the estimated smallest complete footprint in this scenario is $383 \times 313m^2$, the scenario is divided into a grid of 40×28 cells, each of them of $250 \times 250m^2$.

The results of running both configurations of the planner for this scenario are presented in Figure 9. The ETD evolution graph in Figure 9(d) shows again that the proposed $GA_{\theta,v,\alpha}$

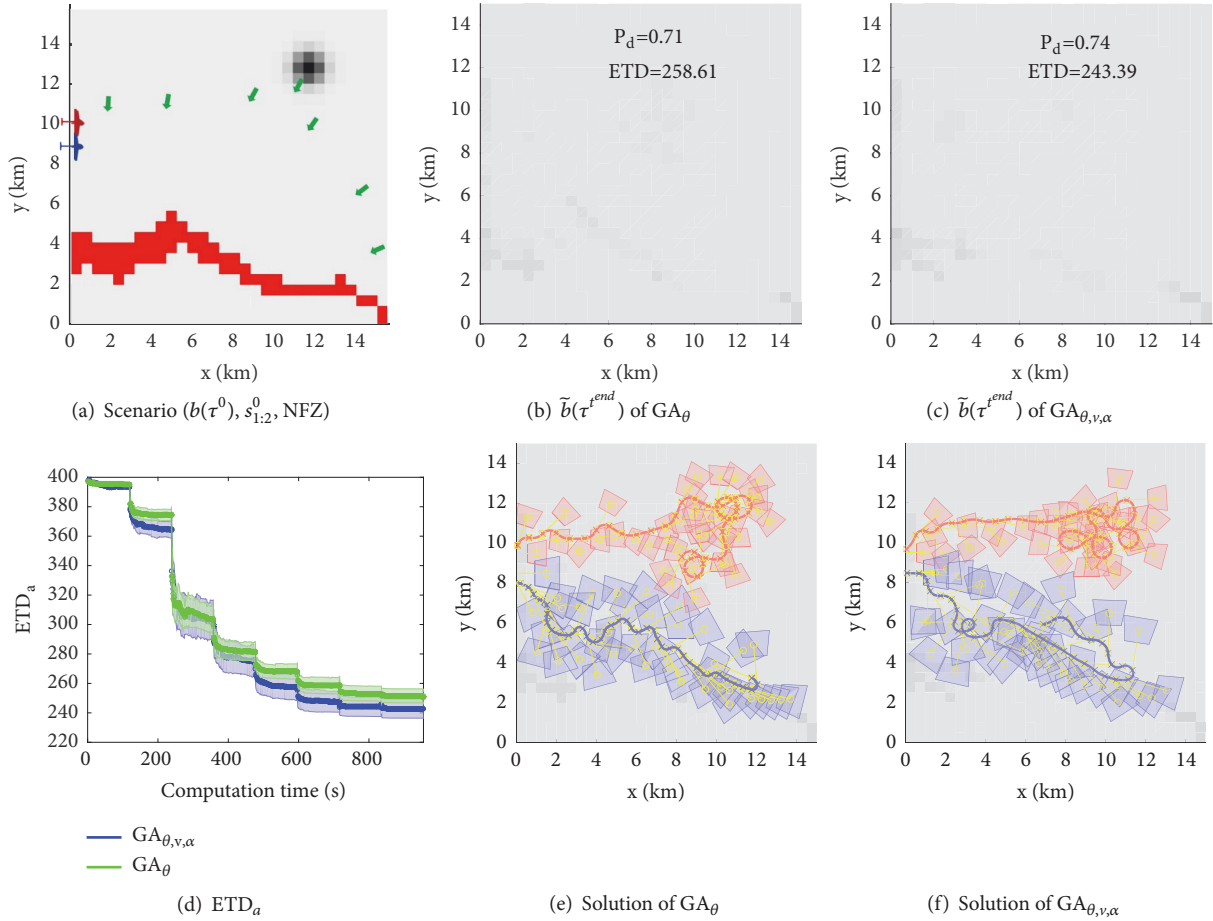


FIGURE 10: Results of searching for a drifting boat (Areia Branca, Brazil).

outperforms GA_θ regarding the ETD. This happens because both UAVs of $\text{GA}_{\theta,v,\alpha}$ can arrive quicker to the high probability areas (in the damaged buildings). It is worth noting also that the probabilities of detection $P_d(s_{1:U}^{0:t^{end}})$ at the end of the mission of two representative solutions of each GA are similar (54% in GA_θ vs. 55% in $\text{GA}_{\theta,v,\alpha}$), because both solutions manage to get the same belief during the development of the mission. In more detail, the analysis of $\tilde{b}(\tau^{t^{end}})$ and of the trajectories in the second and third column of Figure 9 shows that the solution by $\text{GA}_{\theta,v,\alpha}$ collects quicker the concentrated intelligence belief than the elevation belief, while the solution by GA_θ gets quicker the elevation belief than part of the intelligence belief. However, the similarity in the values of the final probability of detection does not imply that the solutions are equivalent for our planner, as its purpose is to minimize the expected time of detection (which is better for $\text{GA}_{\theta,v,\alpha}$ than for GA_θ) and not the probability of detection.

Another interesting result of this scenario is the higher variance in the ETD of $\text{GA}_{\theta,v,\alpha}$. This is caused by the uniformity of elevation probability layer, which hardens the scenario for multisteped planners based on generic heuristics as ours. The effect is bigger in the ETD of $\text{GA}_{\theta,v,\alpha}$ than in the ETD of GA_θ , since its capability to manipulate more decision variables allows the planner to obtain more solutions

that are similar to the myopic criterium and different from the ETD perspective.

5.4. Looking for a Drifting Boat in the Sea. The target of the last scenario is a drifting boat ($S_{target} \approx 5\text{m}$) close to the coast (Areia Branca, Brazil) within a square area of $15 \times 15 \text{km}^2$, whose elevation varies from sea level to 147m . The construction of its initial belief and target motion models has already been described in the Minimum Time Search Definition section.

Besides in this scenario, there is a NFZ (e.g., restricted military area) over the coast cells marked in red in Figure 10(a), where the green arrows summarize the tendency of the target motion model. Both UAVs enter the search area Ω from the west at $t_{1,2}^{start} = 0$ and stay in the mission up to $t_{1,U}^{end} = 400\text{s}$. Their flying capabilities are different: the first UAV (in red) can do it at $h_1^t \in [500, 1500]\text{m}$ and $v_1^t \in [60, 90]\text{m/s}$, while the second (in blue) can do it at $h_2^t \in [500, 2000]\text{m}$ and $v_2^t \in [30, 60]\text{m/s}$. The camera azimuth of both $\alpha_{1,2}^t \in [-90^\circ, 90^\circ]$. Besides, both UAVs are required to fly at their highest altitude with a $\text{FOV} = 30^\circ$ to ensure that the sensor likelihood of the first UAV varies from 0.95 to 0.97 and of the second one from 0.89 to 0.91. Additionally, as the estimated smallest complete footprint is $990 \times 821 \text{m}^2$,

we divide the search area Ω into a grid of 30x30 cells of 500x500m². Finally, as the drifting boat is a dynamic target, we also have to select its motion lapse. Taking into account the size of the cells, we make $T_\tau = 50$ s with the purpose of letting the target advance at approximately 10m/s.

The results of running both configurations of the planner for this scenario are presented in Figure 10. Again, the ETD evolution graph in Figure 10(d) shows that $GA_{\theta,v,\alpha}$ produces overall better ETD results than GA_θ , at the expense of a bigger variability of its ETD. The initial closer mean values of the ETD results in $GA_{\theta,v,\alpha}$ and GA_θ are due to the fact that the $b(\tau^0)$ of this scenario and the initial locations of the UAVs let their cameras initially observe areas with the same probability independently of the UAVs speed and camera azimuth. The benefit on ETD comes later, when the UAVs with changing velocity can arrive quicker to the high probability areas.

For this case, the $P_d(s_{1:U}^{0:t^{end}})$ of the representative solution of $GA_{\theta,v,\alpha}$ is also higher than its counterpart in GA_θ (74% vs. 71%). That is, the planner with $GA_{\theta,v,\alpha}$ lets the UAVs collect quicker more probability. Several benefits of changing $v_u^{c,t}$ and $\alpha_u^{c,t}$ are clearly observed in the solutions displayed in the second and third columns of Figure 10. For $GA_{\theta,v,\alpha}$ configuration, not only does the red UAV arrive sooner to the moving Gaussian, but also the blue UAV orients the camera to cover the coast at its right while flying straightforward, while in GA_θ the red UAV follows the track left by the Gaussian and the blue UAV has to change its orientation several times to be able to collect the belief on the coast. Hence, $GA_{\theta,v,\alpha}$ obtains safer straighter flights, where the UAV headings changes proposed by GA_θ are substituted by camera reorientations. It is also worth noting that in both solutions the red footprint is smaller than the blue, because the first UAV flies at lower altitude than the second.

6. Conclusions

This paper presents a new planner for real-world scenarios that optimizes the UAV trajectories and poses of its onboard camera for minimum time search missions with uncertain UAV target locations and movements, and noisy camera-based target detection. This new planner extends the capabilities of the one presented in [10] by (1) permitting the change of UAV heading, speed, and height and of camera azimuth and elevation angles; (2) supporting the use of a DEM and of intelligence information of the search area to define the target behavior (initial belief and motion); (3) modeling realistically (considering the target size, sensor specifications, and DEM) the cameras used by the UAVs to search the target; and (4) handling scenarios with moving targets. Besides, we also present an algorithmic description of the planner and analyze the effects of its different elements in its computational complexity. Finally, we (1) show how to model in detail 3 real-world inspired scenarios (looking for an off-road vehicle in the mountains, survivors in an earthquake, and a drifting boat by the coast) taking into account the influence of different variables (e.g., target size, camera FOV, search area and grid size, and UAV height) and (2) analyze the performance and benefits of the new planner capabilities to tackle them.

As future work, we plan to further extend the capabilities of the planner. On the one hand, we would like to let it handle the definition of the probabilistic behavior of multiple targets and the optimization of the expected detection time of all of them. On the other hand, we are considering treating each action type differently, by including weights in the smoothness functions and spending different computational resources in each type (e.g., using a different action step T_a for each type or permitting the multisteped planner to modify each of them in different runs of the GA with a different time computation limit for each type). Besides, we are also exploring the possibility of substituting the GA that supports the optimization of our planner by other metaheuristics, to include new heuristics in the planner for speeding up the optimization, or to generate solutions supported partially by predefined patterns (e.g., spirals, lawnmower) often used in general probabilistic target search problems.

Appendix

In this appendix we sketch the differential equations that model the UAV motion ((A.2)-(A.8)) and sensor motion ((A.2)-(A.8)). In more detail, we consider that the velocity of fuel consumption \dot{fuel}_u^t is proportional to the UAV air speed $v_u^{air,t}$; that the horizontal speeds \dot{x}_u^t and \dot{y}_u^t are related to the UAV air speed $v_u^{air,t}$ and orientation θ_u^t , and to the wind speed v_{wind}^t and orientation θ_{wind}^t ; that the derivatives of the UAV air acceleration $\dot{v}_u^{air,t}$, the UAV angular acceleration $\dot{\theta}_u^t$, and the UAV height acceleration \dot{h}_u^t are related to their time constants $(\tau_v, \tau_\theta, \tau_h)$ and commanded speed $v_u^{c,t}$, orientation $\theta_u^{c,t}$, and altitude $h_u^{c,t}$; and that the camera elevation and azimuth speeds $(\dot{\eta}_u^t, \dot{\alpha}_u^t)$ are related to the commanded camera poses $(\eta_u^{c,t}, \alpha_u^{c,t})$. Besides, although not stated in the expressions, the UAV air speed $v_u^{air,t}$, angular speed $\dot{\theta}_u^t$, and vertical speed \dot{h}_u^t and the camera elevation and azimuth speeds $(\dot{\eta}_u^t, \dot{\alpha}_u^t)$ are bounded, and the rate of change of the last three $(\dot{h}_u^t, \dot{\eta}_u^t, \dot{\alpha}_u^t)$ is limited (see the Simulink model in Figure 5 for further details).

$$\dot{fuel}_u^t = K_{fuel} v_u^{air,t} \quad (A.1)$$

$$\dot{x}_u^t = v_u^{air,t} \cdot \cos(\theta_u^t) + v_{wind}^t \cdot \cos(\theta_{wind}^t) \quad (A.2)$$

$$\dot{y}_u^t = v_u^{air,t} \cdot \sin(\theta_u^t) + v_{wind}^t \cdot \sin(\theta_{wind}^t) \quad (A.3)$$

$$v_u^t = \sqrt{(\dot{x}_u^t)^2 + (\dot{y}_u^t)^2} \quad (A.4)$$

$$\dot{v}_u^{air,t} = \frac{K_v (v_u^{c,t} - v_u^t) - \dot{v}_u^{air,t}}{\tau_v} \quad (A.5)$$

$$\dot{\theta}_u^t = \frac{K_\theta (\theta_u^{c,t} - \theta_u^t) - \dot{\theta}_u^t}{\tau_\theta} \quad (A.6)$$

$$\dot{h}_u^t = \frac{K_h (h_u^{c,t} - h_u^t) - \dot{h}_u^t}{\tau_h} \quad (A.7)$$

$$\dot{\eta}_u^t = K_\eta (\eta_u^{c,t} - \eta_u^t) \quad (A.8)$$

$$\dot{\alpha}_u^t = K_\alpha (\alpha_u^{c,t} - \alpha_u^t) \quad (\text{A.9})$$

The model parameters ($K_{fuel}, K_v, \tau_v, K_\theta, \tau_\theta, K_h, \tau_h, K_\eta$ and K_α) and the limits in the speeds and rate changes allow adapting the movement of the UAV and of the camera to the behavior of different real-world aircraft and camera gimbals. Finally, this differential model can be simulated, after rewriting the expression in a state-space fashion, using different tools (such as the ODE solvers in Matlab or a Runge-Kutta algorithm).

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Airbus under SAVIER AER-30459 project.

References

- [1] S. Wang, L. Ding, Z. Chen, and A. Dou, "A rapid UAV image georeference algorithm developed for emergency response," *Journal of Sensors*, vol. 2018, Article ID 8617843, 10 pages, 2018.
- [2] J. Linchant, J. Lisein, J. Semeki, P. Lejeune, and C. Vermeulen, "Are unmanned aircraft systems (UASs) the future of wildlife monitoring? A review of accomplishments and challenges," *Mammal Review*, vol. 45, no. 4, pp. 239–252, 2015.
- [3] G. W. Pulford, "Taxonomy of multiple target tracking methods," *IEEE Proceedings-Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 291–304, 2005.
- [4] P. Lanillos, E. Besada-Portas, G. Pajares, and J. J. Ruz, "Minimum time search for lost targets using cross entropy optimization," in *Proceedings of the 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems, IROS 2012*, pp. 602–609, Portugal, October 2012.
- [5] P. Lanillos, J. Yáñez-Zuluaga, J. J. Ruz, and E. Besada-Portas, "A Bayesian approach for constrained multi-agent minimum time search in uncertain dynamic domains," in *Proceedings of the 2013 15th Genetic and Evolutionary Computation Conference, GECCO 2013*, pp. 391–398, Netherlands, July 2013.
- [6] S. Perez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz, "Ant colony optimization for multi-UAV minimum time search in uncertain domains," *Applied Soft Computing*, vol. 62, pp. 789–806, 2018.
- [7] P. Lanillos, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz, "Minimum time search in uncertain dynamic domains with complex sensorial platforms," *Sensors*, vol. 14, no. 8, pp. 14131–14179, 2014.
- [8] M. Meghjani, S. Manjanna, and G. Dudek, "Multi-target rendezvous search," in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2016*, pp. 2596–2603, Republic of Korea, October 2016.
- [9] S. Perez-Carabaza, J. Bermudez-Ortega, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. De La Cruz, "A Multi-UAV minimum time search planner based on ACOR," in *Proceedings of the 2017 Genetic and Evolutionary Computation Conference, GECCO 2017*, pp. 35–42, ACM, New York, NY, USA, 2017.
- [10] S. Perez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. De La Cruz, "A real world multi-UAV evolutionary planner for minimum time target detection," in *Proceedings of the 2016 Genetic and Evolutionary Computation Conference, GECCO 2016*, pp. 981–988, USA, July 2016.
- [11] E. M. Wong, F. Bourgault, and T. Furukawa, "Multi-vehicle Bayesian search for multiple lost targets," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 3169–3174, Spain, April 2005.
- [12] F. Bourgault, T. Furukawa, and H. Durrant-Whyte, "Optimal search for a lost target in a bayesian world," in *Field and Service Robotics, Recent Advances in Research and Applications*, pp. 209–222, 2006.
- [13] J. Tisdale, Z. W. Kim, and J. K. Hedrick, "Autonomous UAV path planning and estimation: an online path planning framework for cooperative search and localization," *IEEE Robotics and Automation Magazine*, vol. 16, no. 2, pp. 35–42, 2009.
- [14] F. M. Delle Fave, Z. Xu, A. Rogers, and N. R. Jennings, "Decentralised coordination of unmanned aerial vehicles for target search using the max-sum algorithm," in *Proceedings of the Workshop on Agents in Real Time and Environment*, pp. 35–44, 2010.
- [15] S. K. Gan and S. Sukkariéh, "Multi-UAV target search using explicit decentralized gradient-based negotiation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.
- [16] P. Lanillos, S. K. Gan, E. Besada-Portas, G. Pajares, and S. Sukkariéh, "Multi-UAV target search using decentralized gradient-based negotiation with expected observation," *Information Sciences*, vol. 282, pp. 92–110, 2014.
- [17] J. Hu, L. Xie, J. Xu, and Z. Xu, "Multi-agent cooperative target search," *Sensors*, vol. 14, no. 6, pp. 9408–9428, 2014.
- [18] Y. Yang, A. A. Minai, and M. M. Polycarpou, "Decentralized cooperative search in UAV's using opportunistic learning," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit 2002*, USA, August 2002.
- [19] M. Zhang, J. Song, L. Huang, and C. Zhang, "Distributed cooperative search with collision avoidance for a team of unmanned aerial vehicles using gradient optimization," *Journal of Aerospace Engineering*, vol. 30, no. 1, 2016.
- [20] S. Carpin, D. Burch, N. Basilico, T. H. Chung, and M. Kölsch, "Variable resolution search with quadrotors: theory and practice," *Journal of Field Robotics*, vol. 30, no. 5, pp. 685–701, 2013.
- [21] G. Collins, J. Riehl, and P. S. Vegdahl, "A UAV routing and sensor control optimization algorithm for target search," in *Proceedings of SPIE*, pp. 1–13, 2007.
- [22] P. Skoglar, U. Orguner, D. T. ornqvist, and F. Gustafsson, "Road target search and tracking with gimballed vision sensor on an unmanned aerial vehicle," *Remote Sensing*, vol. 4, no. 7, pp. 2076–2111, 2012.
- [23] K. E. Trummel and J. R. Weisinger, "The complexity of the optimal searcher path problem," *Operations Research*, vol. 34, no. 2, pp. 324–327, 1986.
- [24] I. Abi-Zeid and J. R. Frost, "SARPlan: A decision support system for Canadian search and rescue operations," *European Journal of Operational Research*, vol. 162, no. 3, pp. 630–653, 2005.

- [25] Ø. Breivik and A. A. Allen, "An operational search and rescue model for the Norwegian Sea and the North Sea," *Journal of Marine Systems*, vol. 69, no. 1-2, pp. 99–113, 2008.
- [26] L. Lin and M. A. Goodrich, "A Bayesian approach to modeling lost person behaviors based on terrain features in wilderness search and rescue," *Computational and Mathematical Organization Theory*, vol. 16, no. 3, pp. 300–323, 2010.
- [27] M. Wysokinski, R. Marcjan, and J. Dajda, "Decision support software for search rescue operations," *Procedia Computer Science*, vol. 35, pp. 776–785, 2014.
- [28] BMT, "Search and rescue information system," <https://www.bmtscd.com/products/saris/>.
- [29] Z. Li, C. Zhu, and C. Gold, *Digital Terrain Modeling: Principles and Methodology*, CRC Press, 2004.
- [30] R. H. Vollmerhausen, E. Jacobs, and R. G. Driggers, "New metric for predicting target acquisition performance," *Optical Engineering*, vol. 43, no. 11, pp. 2806–2818, 2004.
- [31] A. Watt, *3D Computer Graphics*, Pearson, 2000.
- [32] J. Hixson, E. Jacobs, and R. Vollmerhausen, "Target detection cycle criteria when using the targeting task performance metric," *Electro-Optical and Infrared Systems: Technology and Applications*, vol. 5612, pp. 275–283, 2004.
- [33] J. Johnson, "Analysis of image forming systems," in *Proceedings of the Image Intensifier Symposium*, pp. 249–273, 1958.
- [34] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proceedings of the VI International Conference on Parallel Problem Solving from Nature*, pp. 849–858, 2000.
- [35] L. Lin and M. A. Goodrich, "UAV intelligent path planning for wilderness search and rescue," in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 709–714, USA, October 2009.
- [36] N. Lo, J. Berger, and M. Noel, "Toward optimizing static target search path planning," in *Proceedings of the 2012 5th IEEE Symposium on Computational Intelligence for Security and Defence Applications, CISDA 2012*, pp. 1–7, Canada, July 2012.
- [37] J. Berger, N. Lo, and M. Noel, "A new multi-target, multi-agent search-and-rescue path planning approach," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 8, no. 6, pp. 978–987, 2014.
- [38] A. Khan, E. Yanmaz, and B. Rinner, "Information exchange and decision making in micro aerial vehicle networks for cooperative search," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 4, pp. 335–347, 2015.
- [39] C.-J. Ru, X.-M. Qi, and X.-N. Guan, "Distributed cooperative search control method of multiple UAVs for moving target," *International Journal of Aerospace Engineering*, vol. 2015, Article ID 317953, 12 pages, 2015.
- [40] J. Berger, N. Lo, and M. Barkaoui, "Static target search path planning optimization with heterogeneous agents," *Annals of Operations Research*, vol. 244, no. 2, pp. 1–18, 2016.
- [41] P. Yao, H. Wang, and H. Ji, "Gaussian mixture model and receding horizon control for multiple UAV search in complex environment," *Nonlinear Dynamics*, vol. 88, no. 2, pp. 903–919, 2017.
- [42] 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, Article ID 6879419, 17 pages, 2018.
- [43] U. S. G. S., "Elevation data source," <https://earthexplorer.usgs.gov/>.
- [44] H. Earthquake, "Remote sensing based building damage assessment data," 2010.
- [45] Z. Beck, L. Teacy, A. Rogers, and N. Jennings, "Online planning for collaborative search and rescue by heterogeneous robot teams," in *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems*, 2016.

