

Integrated Guidance and Gimbal Control for Coverage Planning with Visibility Constraints

Savvas Papaioannou, Panayiotis Kolios, Theocharis Theocharides,
Christos G. Panayiotou and Marios M. Polycarpou

Abstract—Coverage path planning with unmanned aerial vehicles (UAVs) is a core task for many services and applications including search and rescue, precision agriculture, infrastructure inspection and surveillance. This work proposes an integrated guidance and gimbal control coverage path planning (CPP) approach, in which the mobility and gimbal inputs of an autonomous UAV agent are jointly controlled and optimized to achieve full coverage of a given object of interest, according to a specified set of optimality criteria. The proposed approach uses a set of visibility constraints to integrate the physical behavior of sensor signals (i.e., camera-rays) into the coverage planning process, thus generating optimized coverage trajectories that take into account which parts of the scene are visible through the agent's camera at any point in time. The integrated guidance and gimbal control CPP problem is posed in this work as a constrained optimal control problem which is then solved using mixed integer programming (MIP) optimization. Extensive numerical experiments demonstrate the effectiveness of the proposed approach.

Index Terms—Guidance and control, coverage planning, trajectory planning, autonomous agents, unmanned aerial vehicles (UAVs).

I. INTRODUCTION

OVER the last years we have witnessed an accelerated demand for Unmanned Aerial Vehicles (UAVs) and Unmanned Aerial Systems (UASs) in various application domains including search and rescue [1]–[6], precision agriculture [7], package delivery [8], wildfire monitoring [9] and security [10]–[12]. This high demand is mainly fueled by the recent advancements in automation technology, avionics and intelligent systems, in combination with the proliferation and cost reduction of electronic components. Amongst the most crucial capabilities for a fully autonomous UAV system is that of path/trajectory planning [13], which play a pivotal role in designing and executing automated flight plans, required by the majority of application scenarios. The path planning problem encapsulates algorithms that compute trajectories between the desired start and goal locations. Moreover, for many tasks such as structure inspection, target search, and surveillance, there is a great need for efficient and automated coverage path planning (CPP) [14] techniques. Coverage path planning consists of finding a path (or trajectory) which allows an autonomous agent (e.g., a UAV) to cover every point (i.e., the point must be included within the agent's sensor footprint) within a certain area of interest.

Despite the overall technological progress in CPP techniques over the last decades, there is still work to be done for

constructing solutions to the level of realism that can support practical autonomous UAV operations. As discussed in more detail in Sec. II, the vast majority of CPP approaches mainly consider ground vehicles or robots with static and fixed sensors (i.e., the sensor mounted on the robot is not controllable and the size of the sensor's footprint does not change). These assumptions reduce the CPP problem to a path-planning problem, where a) the area of interest is first decomposed into a finite number of cells (where usually each cell has size equal to the sensor's footprint), and b) the path that passes through all cells is generated with a path-finding algorithm, thus achieving full coverage. In some approaches (e.g., [15]) the generated path is adapted to the robot's dynamic/kinematic model at a second stage. This two-stage approach however, usually produces sub-optimal results, in terms of coverage performance. In addition, UAVs are usually equipped with a gimbal sensor, which potentially exhibits a dynamic sensing range e.g., a pan-tilt-zoom (PTZ) camera. Therefore, in order to optimize coverage, necessitates the implementation of CPP techniques with the ability to optimize not only the UAV's trajectory, but also the control inputs of the onboard gimbal sensor.

Specifically, in this paper we investigate the coverage path planning problem for a known 2D region/object of interest, with a UAV agent, which exhibits a controllable gimbal sensor with dynamic sensing range. In particular, we propose an integrated guidance and gimbal control coverage path planning approach, where we jointly control and optimize a) the UAV's mobility input governed by its kinematic model and b) the UAV's gimbal sensor, in order to produce optimal coverage trajectories. The CPP problem is posed as a constrained optimal control problem, where the goal is to optimize a set of mission-specific objectives subject to coverage constraints. As opposed to the majority of existing techniques, in this work we consider the CPP problem in the presence of state-dependent visibility-constraints. In particular, the proposed approach integrates ray-casting into the planning process in order to determine which parts of the region/object of interest are visible through the UAV's sensor at any point in time, thus enabling the generation of realistic UAV trajectories.

Specifically, the contributions of this work are the following:

- We propose an integrated guidance and gimbal control CPP approach for the problem of coverage planning in the presence of kinematic and sensing constraints including state-dependent visibility constraints i.e., we are simulating the physical behavior of sensor signals in order to account for the parts of the scene that are blocked by obstacles and thus cannot be observed by the UAV's sensor at a given pose.

The authors are with the KIOS Research and Innovation Centre of Excellence (KIOS CoE) and the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, 1678, Cyprus. {papaioannou.savvas, pkolios, ttheocharides, christosp, mpolycar}@ucy.ac.cy

- We formulate the CPP problem as a constrained optimal control problem, in which the UAV mobility and gimbal inputs are jointly controlled and optimized according to a specified set of optimality criteria, and we solve it using mixed integer programming (MIP). The performance of the proposed approach is demonstrated through a series of numerical experiments.

The rest of the paper is organized as follows. Section II summarizes the related work on coverage path planning with ground and aerial vehicles. Then, Section III develops the system model, Section IV defines the problem tackled and Section V discusses the details of the proposed coverage planning approach. Finally, Section VI evaluates the proposed approach and Section VII concludes the paper and discusses future work.

II. RELATED WORK

In coverage path planning (CPP) we are interested in determining the path/trajectory that enables an autonomous agent to observe with its sensor every point in a given environment. Early works e.g., [16], [17], treated the CPP problem as a path-planning problem, by decomposing the environment into several non-overlapping cells, and then employing a path-planning algorithm [18] to find the path that passes through every cell. Notably, in [16], [17] the authors propose cellular decomposition coverage algorithms, where the free-space of the environment is decomposed into non-intersecting regions or cells, which can be covered by the robot, one by one using simple back-and-forth motions, thus covering the whole area. Extensions [19]–[21] of this approach have investigated the coverage/sweeping pattern, the sensor's footprint and the region traversal order for optimal coverage. In [22], the authors propose a CPP approach based on spanning trees, for covering, in linear time, a continuous planar area with a mobile robot equipped with a square-shaped sensor footprint. In the subsequent work [23], the planar area to be covered is incrementally sub-divided into disjoint cells on-line, allowing for real-time operation. The spanning tree approach presented in [22] and [23] is extended in [24] for multi-robot systems. In [25], the boustrophedon cellular decomposition algorithm (BDC) [26] is used to compute a graph-based representation of the environment, and then a coverage clustering algorithm is proposed which divides the coverage path among multiple robots. The problem of coverage path planning has also been investigated with multiple ground robots in [27]. The authors of [27] propose an approach that minimizes the total coverage time by transforming the CPP problem into a network flow problem. In [28], a graph-based coverage algorithm is proposed for a CPP problem variation, which enables a team of robots to visit a set of predefined locations according to a specified frequency distribution. Another CPP variation is investigated in [29], [30]. The authors propose decentralized, adaptive control laws to drive a team of mobile robots to an optimal sensing configuration which achieves maximum coverage of the area of interest. The same problem is investigated in [31] with heterogeneous robots that exhibit different sensor footprints.

Interestingly, the CPP problem has recently gained significant attention due to its importance in various UAV-based applications, including emergency response [5], [6], [32],

critical infrastructure inspection [33] and surveillance [34]. A UAV-based coverage path planning approach utilizing exact cellular decomposition in polygonal convex areas is proposed in [35], whereas in [15] the CPP problem is adapted for a fixed-wing UAV. In [15] the ground plane is first decomposed into several non-overlapping cells which are then connected together forming a piecewise-linear coverage path. At a second stage a UAV-specific motion controller is used to convert the generated path into a smooth trajectory which the UAV can execute. An information theoretic coverage path planning approach for a single fixed-wing UAV is presented in [36]. The aircraft maintains a coverage map which uses to it make optimized control decisions on-line, in order to achieve global coverage of the environment. In [37], the authors propose a clustering-based CPP approach for searching multiple regions of interest with multiple heterogeneous UAVs, by classifying the various regions into clusters and assigning the clusters to the UAVs according to their capabilities. In [38]–[40] CPP techniques for energy-constraint multi-UAV systems are proposed. Moreover, in [41], a distributed coverage control approach is proposed for a multi-UAV system. Specifically, a coverage reference trajectory is computed using a Voronoi-based partitioning scheme, and then a distributed control law is proposed for guiding the UAVs to follow the reference trajectory, thus achieving full coverage. Finally, the CPP problem has also been investigated more recently with learning based techniques [42], [43]. In [42], an end-to-end deep reinforcement learning CPP approach is proposed for an autonomous UAV agent. The authors utilize a double deep Q-network to learn a coverage control policy that takes into account the UAV's battery constraints. On the other hand in [43], a coverage path planning system based on supervised imitation learning is proposed for a team of UAV agents. This approach plans coordinated coverage trajectories which allow unexplored cells in the environment to be visited by at least one UAV agent.

The problem of coverage path planning (CPP) is also related to the view planning problem (VPP) and its variations, where the objective is to find the minimum number of viewpoints that completely cover the area of an object of interest. The relationship between these two problems is showcased in [44], in which the CPP problem is posed as a view planning problem. Specifically, the authors in [44] propose an algorithm based on the traveling salesman problem (TSP), which incorporates visibility constraints and finds the tour of minimum length which allows a UAV equipped with a fixed downward facing camera to inspect a finite sets of points on the terrain. On the other hand, the authors in [45] propose a variation of the original VPP problem, termed traveling VPP, where the objective is the minimization of the combined view and traveling cost i.e., the cost to minimize combines the view cost which is proportional to the number of viewpoints planned, and the traveling cost which accounts for the total distance that the robot needed to travel in order to cover all points of interest. In general, VPP approaches operate in a discrete state-space setting and are mostly concerned with the selection of an optimal sequence of sensing actions (taken from the finite set of all admissible actions) which achieve full coverage of the object of interest. In contrast, the proposed approach can be used to generate continuous trajectories which are governed by kinematic and sensing constraints. The interested reader is

directed to [14], [46], [47] for a more detailed examination of the various coverage path planning techniques in the literature, including the view planning problem.

To summarize, in comparison with the existing techniques, in this work we propose a coverage planning approach which integrates ray-casting into the planning process, in order to simulate the physical behavior of sensor signals and thus determine which parts of the scene are visible through the UAV's onboard camera. The proposed approach takes into account both the kinematic and the sensing constraints of the UAV agent, to achieve full coverage of an object of interest in the presence of obstacles. In particular, the coverage path planning problem is posed in this work as a constrained open-loop optimal control problem which incorporates the UAV's kinematic and sensing model to produce optimal coverage trajectories in accordance with the specified mission objectives. Finally, the proposed mathematical formulation can be solved optimally with off-the-shelf mixed integer programming (MIP) optimization tools [48].

III. SYSTEM MODEL

A. Agent Kinematic Model

This work assumes that an autonomous agent (e.g., a UAV), is represented as a point-mass object which maneuvers inside a bounded surveillance area $\mathcal{W} \subset \mathbb{R}^2$. The agent kinematics are governed by the following discrete-time linear model [49]:

$$x_t = \Phi x_{t-1} + \Gamma u_{t-1}, \quad (1)$$

where $x_t = [p_t(x), p_t(y), v_t(x), v_t(y)]^\top$ denotes the agent's state at time t in cartesian (x, y) coordinates, which consists of the agent's position $[p_t(x), p_t(y)]^\top \in \mathbb{R}^2$ and velocity $[v_t(x), v_t(y)]^\top \in \mathbb{R}^2$ in the x and y directions. The term $u_t = [f_t(x), f_t(y)]^\top \in \mathbb{R}^2$ denotes the control input, i.e., the amount of force applied in each dimension in order to direct the agent according to the mission objectives. The matrices Φ and Γ are given by:

$$\Phi = \begin{bmatrix} \mathbf{I}_{2 \times 2} & \delta t \cdot \mathbf{I}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \phi \cdot \mathbf{I}_{2 \times 2} \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \mathbf{0}_{2 \times 2} \\ \gamma \cdot \mathbf{I}_{2 \times 2} \end{bmatrix}, \quad (2)$$

where δt is the sampling interval, $\mathbf{I}_{2 \times 2}$ and $\mathbf{0}_{2 \times 2}$ are the 2 by 2 identity matrix and zero matrix respectively, with ϕ and γ given by $\phi = (1 - \eta)$ and $\gamma = m^{-1} \delta t$, where the parameter η is used to model the (air) drag coefficient and m is the agent mass. Given a known initial agent state x_0 , and a set of control inputs $\{u_t | t = 0, \dots, T-1\}$, inside a finite planning horizon of length T , the agent trajectory can be obtained for time-steps $t = [1, \dots, T]$ by the recursive application of Eqn. (1) as:

$$x_t = \Phi^t x_0 + \sum_{\tau=0}^{t-1} \Phi^{t-\tau-1} \Gamma u_\tau. \quad (3)$$

Therefore, the agent's trajectory can be designed and optimized to meet the desired mission objectives by appropriately selecting the control inputs $\{u_t | t = 0, \dots, T-1\}$, inside the given planning horizon of length T . We should point out here that although the agent kinematic model in Eqn. (1) does not fully captures the low-level UAV aerodynamics (which are platform dependent), it allows us to design and construct high-level (i.e., mission-level) coverage trajectories which in turn can be used as desired reference trajectories to be tracked with

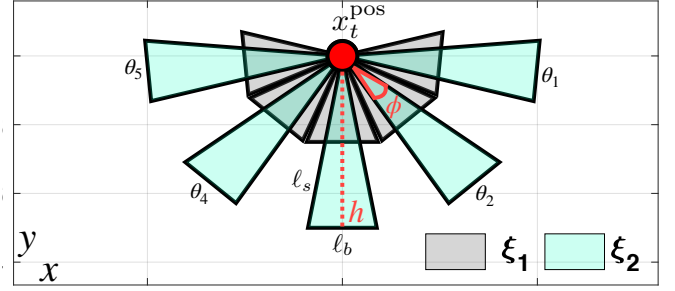


Fig. 1. The figure illustrates all the possible sensor FoV configurations for two different zoom-levels ξ_1 and ξ_2 , and for 5 different rotation angles $\theta_1, \dots, \theta_5$, when the agent's position is equal to x_t^{pos} . In this example, the total number of FoV configurations is equal to $|\bar{\Theta}| \times |\bar{\Xi}| = 10$.

low-level closed-loop guidance controllers found on-board the UAVs [50]–[52].

B. Agent Sensing Model

The agent is equipped with a gimbaled camera with optical zoom, which is used for sensing its surroundings and performing various tasks e.g., searching objects/regions of interest, detecting targets, etc. The camera field-of-view (FoV) or sensing footprint is modeled in this work as an isosceles triangle [53], [54] parameterized by its angle at the apex φ and its height h , which are used to model the FoV angle opening and sensing range respectively. We should point out here that any convex 2D shape can be used to model the camera FoV. Using the parameters φ and h the camera FoV side length (ℓ_s) and base length (ℓ_b) are computed as:

$$\ell_s = h \times \cos(\varphi/2)^{-1}, \text{ and, } \ell_b = 2\ell_s \times \sin(\varphi/2). \quad (4)$$

Therefore, the set of vertices ($\mathcal{V}_o$) of the triangular FoV camera projection, for an agent centered at the origin, and facing downwards are given by $\mathcal{V}_o = [v_1, v_2, v_3]^\top$, where $v_1 = [0, 0]^\top$, $v_2 = [-\ell_b/2, -h]^\top$ and $v_3 = [\ell_b/2, -h]^\top$ so that:

$$\mathcal{V}_o = \begin{bmatrix} 0 & -\ell_b/2 & \ell_b/2 \\ 0 & -h & -h \end{bmatrix}, \quad (5)$$

The camera FoV can be rotated (on the xy -plane) around the agent's position $x^{\text{pos}} = [p(x), p(y)]^\top$ by an angle $\theta \in \bar{\Theta}$ (with respect to x -axis), by performing a geometric transformation consisting of a rotation operation followed by a translation:

$$\mathcal{V} = R(\theta) \mathcal{V}_o + x^{\text{pos}}, \quad (6)$$

where $\mathcal{V}$ is the rotated camera FoV in terms of its vertices, θ is the control signal and $R(\theta)$ is a 2D rotation matrix given by:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (7)$$

We should mention here that in this work the rotation angle θ takes its values from a finite set of all admissible rotation angles $\bar{\Theta} = \{\theta_1, \dots, \theta_{|\bar{\Theta}|}\}$, where $|\bar{\Theta}|$ denotes the set cardinality. The agent can be placed in any desired position and orientation (i.e., pose) at some time-step t , by adjusting the control signals u_t and $\theta_t \in |\bar{\Theta}|$ i.e., $\mathcal{V}_t = R(\theta_t) \mathcal{V}_o + x_t^{\text{pos}}$. The agent's onboard camera is also equipped with an optical zoom functionality, which can alter the FoV characteristics in order to better

suit the mission objectives and constraints. In particular, it is assumed that a zoom-in operation narrows down the FoV (i.e., reduces the FoV angle opening φ) however, it increases the sensing range h , as shown in Fig. 1. In particular, we assume that the camera exhibits a finite set of predefined zoom levels denoted as $\Xi = \{\xi_1, \dots, \xi_{|\Xi|} | \xi_i \in \mathbb{R}, \xi_i \geq 1\}$, which are used to scale the FoV parameters. The zoom-in functionality is thus defined for a particular zoom level $\xi \in \Xi$ as:

$$h' = h \times \xi, \text{ and, } \varphi' = \varphi \times \xi^{-1}, \quad (8)$$

where (h', φ') are the new parameters for the FoV angle and range, after applying the optical zoom level ξ . A visual representation of the camera model is illustrated in Fig. 1.

We can now denote the agent's sensing state at time-step t as $S_t(\theta_t, x_t^{\text{pos}}, \xi_t)$, which jointly accounts for the agent's position and orientation. The notation $S_t(\theta_t, x_t^{\text{pos}}, \xi_t)$, is used here to denote the area inside the agent's sensing range determined by the FoV vertices (i.e., the convex hull) as computed by Eqn. (6). Therefore, the total area covered by the agent's FoV within a planning horizon of length T can thus be obtained by:

$$S_{1:T} = \bigcup_{t=1}^T S_t(\theta_t, x_t^{\text{pos}}, \xi_t), \quad (9)$$

where $\theta_t \in \bar{\Theta}$, $\xi_t \in \bar{\Xi}$ and the agent position x_t^{pos} has been computed from the application of a set of mobility controls inputs $\{u_\tau | \tau = 0, \dots, t-1\}$ according to Eqn. (3).

We should point out here that the agent kinematic and sensing models described above can easily be extended to 3 dimensions (i.e., the agent kinematics in Eqn. (1) can be extended in 3D by accounting for the z dimension, the triangular 2D FoV translates to a regular pyramid in 3D, the 2D rotation matrix R becomes a 3D rotation matrix in 3D, etc). Consequently, the proposed approach discussed in detail in Sec.V can also be extended in 3D environments with some modifications. However, in order to make the analysis of the proposed approach easier to follow and more intuitive, the problem in this paper has been formulated in a two dimensional space, which already has some key challenges.

IV. PROBLEM STATEMENT

Let an arbitrary bounded convex planar region $\mathcal{C} \subset \mathcal{W}$ to denote a single object or region of interest, that we wish to cover with our autonomous agent, with boundary $\partial\mathcal{C}$, as shown in Fig. 2. The proposed approach can be used to generate the coverage plan for the area enclosed in $\mathcal{C}$ in the case where the region $\mathcal{C}$ is traversable. On the other hand when $\mathcal{C}$ is not traversable (i.e., $\mathcal{C}$ represents an inaccessible, to the agent, object or region of interest), the proposed technique is used to generate the coverage plan for the region's boundary $\partial\mathcal{C}$. For brevity, we will formulate the problem assuming the latter scenario (i.e., generating coverage plans for covering the boundary of a region/object of interest), however the proposed formulation can be applied for both scenarios. In a high level form, the problem tackled in this work can be formulated as follows:

In (P1) we are interested in finding the agent's mobility (i.e., $\mathbf{U}_T = \{u_0, \dots, u_{T-1}\}$) and sensor (i.e., $\boldsymbol{\Theta}_T = \{\theta_1, \dots, \theta_T\}$, $\boldsymbol{\Xi}_T = \{\xi_1, \dots, \xi_T\}$) control inputs, over the planning horizon

Problem (P1): High-level Controller

$$\arg \min_{\mathbf{U}_T, \boldsymbol{\Theta}_T, \boldsymbol{\Xi}_T} \mathcal{J}_{\text{coverage}}(\mathbf{X}_T, \mathbf{U}_T, \boldsymbol{\Theta}_T, \boldsymbol{\Xi}_T) \quad (10a)$$

subject to: $t = [1, \dots, T]$

$$x_t = \Phi^t x_0 + \sum_{\tau=0}^{t-1} \Phi^{t-\tau-1} \Gamma u_\tau \quad (10b)$$

$$x_0, x_t^{\text{pos}} \notin \psi, \quad \forall \psi \in \Psi \quad (10c)$$

$$\partial\mathcal{C} \in \mathcal{S}'_{1:T} \quad (10d)$$

$$x_0, x_t \in \mathcal{X}, \quad u_t \in \mathcal{U}, \quad \theta_t \in \bar{\Theta}, \quad \xi_t \in \bar{\Xi} \quad (10e)$$

of length T , i.e., $\mathcal{T} = \{1, \dots, T\}$, which optimize a certain set of optimality criteria encoded in the state-dependent multi-objective cost function $\mathcal{J}_{\text{coverage}}(\mathbf{X}_T, \mathbf{U}_T, \boldsymbol{\Theta}_T, \boldsymbol{\Xi}_T)$, where $\mathbf{X}_T = \{x_1, \dots, x_T\}$, subject to the set of constraints shown in Eqn. (10b)-(10e). In particular, Eqn. (10b) represents the agent's kinematic constraints as described in Sec. III-A. Then, Eqn. (10c) represents obstacle avoidance constraints with a specified set of obstacles Ψ , and the constraint in (10d) (i.e., coverage constraint) is used to guarantee that during the planning horizon the whole boundary of the region/object of interest will be covered by the agent's sensor FoV.

We should mention here that the notation $\mathcal{S}'_{1:T} \subseteq \mathcal{S}_{1:T}$ refers to the reduced FoV coverage obtained when obstacles block the sensor signals (i.e., camera-rays) from passing through, thus creating occlusions. In this work we use a set of visibility constraints to distinguish between parts of the scene $p \subset \partial\mathcal{C}$ that belong to the visible field-of-view i.e., $p \in \mathcal{S}'_{1:T}$ and parts p that are occluded. In order to model the visible field-of-view $\mathcal{S}'_{1:T}$ we use ray-casting to simulate the physical behavior of the camera-rays and account for the occluded regions. Therefore, the constraint in Eqn. (10d), enforces the generation of coverage trajectories, which take into account which parts of the scene are visible through the agent's camera at any point in time. Finally, the constraints in Eqn. (10e) restrict the agent's state and control inputs within the desired bounds. In the next section, we discuss in more detail how we have tackled the problem discussed above.

We should mention here that in this work the following assumptions are made: a) the agent has self-localization capability (e.g., via accurate GPS positioning), b) the environment (i.e., object of interest, obstacles, etc.) is known a-priori, and c) the visual data acquisition process is noise-free. However, in certain scenarios in which the assumptions above no longer apply, the agent's visual localization accuracy at the planning time is of essence in generating optimal coverage trajectories. In such scenarios the proposed approach can be combined with active visual localization techniques [55], in order to improve the agent's visual localizability, and generate accurate coverage trajectories.

V. INTEGRATED GUIDANCE AND GIMBAL CONTROL COVERAGE PLANNING

In this section we design a mixed integer quadratic program (MIQP) in order to tackle the optimal control problem of integrated guidance and gimbal control coverage planning, as described in problem (P1).

A. Preliminaries

The proposed approach first proceeds by sampling points $p \in \partial\mathcal{C}$ on the region's boundary, generating the set of points $\mathcal{P} = \{p_1, \dots, p_{|\mathcal{P}|}\} \subset \partial\mathcal{C}$, where $|\mathcal{P}|$ is the set cardinality, thus creating a discrete representation of the region's boundary that needs to be covered, as depicted in Fig. 2. Equivalently, a set of points $\mathcal{P} \subset \mathcal{C}$ are sampled from $\mathcal{C}$ in the scenario where the region of interest is traversable. Essentially, the object of interest is represented in this work as a point-cloud, which in practice can be obtained with a number of scene/object reconstruction techniques [56], [57]. The coverage constraint shown in Eqn. (10d) can now be written as:

$$p \in \mathcal{S}'_{1:T}, \forall p \in \mathcal{P}, \quad (11)$$

and thus we are looking to find the optimal agent mobility and sensor control inputs over the planning horizon, which satisfy the constraint in Eqn. (11), i.e., the set of points $\mathcal{P}$ must be covered by the agent's sensor FoV. We should mention here that the methodology used to generate $\mathcal{P}$ (e.g., systematic selection or sampling procedure) is up to the designer and in accordance to the problem requirements.

In the special scenario examined in this work, in which the region (or object) of interest $\mathcal{C}$ is not traversable, and thus acting as an obstacle to the agent's trajectory, a set of obstacle avoidance constraints are implemented i.e., Eqn. (10c) to prevent collisions between the agent and the obstacle. Intuitively, in such scenarios the agent must avoid crossing the region's boundary $\partial\mathcal{C}$. We denote as $\Delta\mathcal{C}$, the piece-wise linear approximation of the boundary $\partial\mathcal{C}$, that contains the line segments $L_{p_i} = L_{i,j} = \{p_i + r(p_j - p_i) | r \in [0, 1], j \neq i\}$, which are formed when connecting together the pair of points $(p_i, p_j)_{i \neq j} \in \mathcal{P}$, such that the resulting line segments $L_{i,j}$ belong to the boundary of the convex hull of $\mathcal{P}$, as shown in Fig. 2.

Let us assume that $|\mathcal{P}|$ points $\{p_1, \dots, p_{|\mathcal{P}|}\}$ have been sampled from $\partial\mathcal{C}$, and that $\Delta\mathcal{C}$ contains $|\mathcal{P}|$ line segments $\{L_{p_1}, \dots, L_{p_{|\mathcal{P}|}}\}$, where each line segment $L_{p_i}, i = 1, \dots, |\mathcal{P}|$ lies on the line $\tilde{L}_{p_i} = \{x \in \mathbb{R}^{(2,1)} | \alpha_i^\top x = \beta_i\}$, where the line coefficients in the vector α_i determine the outward normal to the i th line segment and β_i is a constant. The area inside the region of interest $\mathcal{C}$ is thus modeled as a convex polygonal obstacle with boundary $\Delta\mathcal{C}$ defined by $|\mathcal{P}|$ linear equations $\tilde{L}_{p_i}, i = 1, \dots, |\mathcal{P}|$. A collision occurs when at some point in time $t \in \mathcal{T}$, the agent's position x_t^{pos} resides within the area defined by the region's boundary $\Delta\mathcal{C}$ or equivalently the following system of linear inequalities is satisfied:

$$\alpha_i^\top x_t^{\text{pos}} < \beta_i, \forall i \in [1, \dots, |\mathcal{P}|]. \quad (12)$$

Hence, a collision can be avoided with $\mathcal{C}$ at time t i.e., $x_t^{\text{pos}} \notin \mathcal{C}$, when $\exists i \in [1, \dots, |\mathcal{P}|] : \alpha_i^\top x_t^{\text{pos}} \geq \beta_i$. This is implemented with a set of mixed integer linear constraints as shown in Eqn. (13a)-(13c).

$$-\alpha_i^\top x_t^{\text{pos}} - Mb_{t,i}^{\text{collision}} \leq -\beta_i, \forall t, i, \quad (13a)$$

$$\sum_{i=1}^{|\mathcal{P}|} b_{t,i}^{\text{collision}} \leq (|\mathcal{P}| - 1), \forall t, \quad (13b)$$

$$b_{t,i}^{\text{collision}} \in \{0, 1\}, \forall t, i. \quad (13c)$$

Specifically, the constraint in Eqn. (13a), uses the binary variable $b_{t,i}^{\text{collision}}$ to determine whether the i th inequality i.e., $\alpha_i^\top x_t^{\text{pos}} < \beta_i$ of Eqn. (12) is satisfied at some time $t \in \mathcal{T}$ by setting $b_{t,i}^{\text{collision}} = 1$, where M is a large positive constant. Then, the constraint in Eqn. (13b) makes sure that for any time-step t , the binary variable $b_{t,i}^{\text{collision}}$ is activated less than $|\mathcal{P}| - 1$ times, to ensure that the agent's position x_t^{pos} does not reside inside the obstacle. These constraints, can be applied for any number of convex polygonal obstacles $\psi \in \Psi$, that need to be avoided, by augmenting the variable $b_{t,i}^{\text{collision}}$ with an additional index to indicate the obstacle number.

B. Visibility Constraints

In the previous section we have described how the region's boundary $\partial\mathcal{C}$ that needs to be covered, is defined as a piece-wise linear approximation $\Delta\mathcal{C}$, and we have also shown how the non-traversable area inside the region of interest $\mathcal{C}$, is modeled with a set of obstacle avoidance constraints, which are used to prevent the agent from passing-through. In this section, we devise a set of visibility constraints, which allows us to determine which parts of the region's boundary are visible (i.e., not blocked by an obstacle) through the agent's camera, given a certain agent pose.

In this work, we use the term camera-rays to denote the light rays that are captured by the camera's optical sensor. Without loss of generality, let us assume that at each time-step t , a finite set of camera-rays enter the optical axis, denoted as $\mathcal{K}_{\theta_t, x_t^{\text{pos}}, \xi_t} = \{K_1, \dots, K_{|\mathcal{K}|}\}$, where $\theta_t, x_t^{\text{pos}}, \xi_t$ determine the agent's pose and subsequently the FoV configuration, as illustrated in Fig. 3(a)-(b).

The individual ray K_i is given by the line-segment $K_i = \{x_t^{\text{pos}} + s(\kappa_i - x_t^{\text{pos}}) | s \in [0, 1]\}$, where x_t^{pos} is the ray's origin given by the agent's position at time t and $\kappa_i \in \mathbb{R}^2$ is a fixed point on the base of the triangle which defines the camera FoV and determines the ray's end point. We can now define the notion of visibility as follows: The point $p_i \in \mathcal{P}, i \in [1, \dots, |\mathcal{P}|]$ on the region's boundary $\partial\mathcal{C}$, which exists on the line-segment $L_{p_i} \in \Delta\mathcal{C}$ is said to be visible at time-step t i.e., $p_i \in \mathcal{S}'_t(\theta_t, x_t^{\text{pos}}, \xi_t)$ when:

$$p_i \in \mathcal{S}_t(\theta_t, x_t^{\text{pos}}, \xi_t) \wedge \exists K \in \mathcal{K}_{\theta_t, x_t^{\text{pos}}, \xi_t} : (K \otimes \Delta\mathcal{C}) = L_{p_i}, \quad (14)$$

where the operation $K \otimes \Delta\mathcal{C}$ is defined as the intersection of the camera-ray K , with the set of line segments in $\Delta\mathcal{C}$, and returns the nearest (i.e., closest distance with respect to the ray's origin) line-segment $L \in \Delta\mathcal{C}$ which the ray K intersects with. In the case where a ray K exhibits no intersections with any line-segment, $\emptyset$ is returned instead. In essence, the point p_i is visible at time t , when both constraints in Eqn. (14) are satisfied i.e., a) p_i is included inside the agent's camera FoV $\mathcal{S}_t(\theta_t, x_t^{\text{pos}}, \xi_t)$ and b) there exists a camera-ray K which first intersects with the line-segment L_{p_i} which contains the point p_i . In other words there is a camera-ray K which does not intersect with any parts of the boundary (i.e., line segments) $L \in \Delta\mathcal{C}$ prior to L_{p_i} , as illustrated in Fig. 3(b).

Let the camera-ray K (i.e., $K = \{x_t^{\text{pos}} + s(\kappa - x_t^{\text{pos}}) | s \in [0, 1]\}$) to have x and y cartesian coordinates given by $x_t^{\text{pos}}(x) + s[\kappa(x) - x_t^{\text{pos}}(x)]$ and $x_t^{\text{pos}}(y) + s[\kappa(y) - x_t^{\text{pos}}(y)]$ respectively. Also, let the x and y cartesian coordinates of the a line-segment $L_{p_i} = L_{i,j} \in \Delta\mathcal{C}$ on the boundary (i.e., $L_{i,j} =$

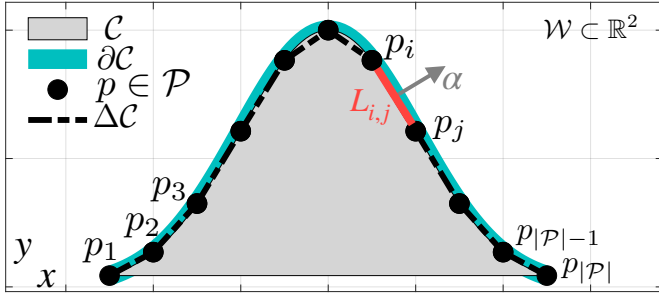


Fig. 2. In the scenario above we are interested in finding the optimal mobility and sensor control inputs for a single UAV agent, which enable full coverage of the boundary ∂C of a given region of interest C . ΔC denotes the piece-wise linear approximation of the boundary which is composed of a finite number of line segments. The points on the boundary p_i and p_j are connected with the line segment $L_{i,j} \in \Delta C$, where α denotes its outward normal vector. Full coverage is achieved when every point $p \in \mathcal{P}$ is included inside the agent's sensor FoV.

$\{p_i + r(p_j - p_i) | r \in [0, 1], j \neq i\}$ to be $p_i(x) + r[p_j(x) - p_i(x)]$ and $p_i(y) + r[p_j(y) - p_i(y)]$ respectively. The intersection $K \otimes L_{i,j}$ of the camera-ray with the line segment can be computed by solving the following system of linear equations for the two unknowns i.e., (s, r) :

$$\begin{bmatrix} \kappa(x) - x_t^{\text{pos}}(x) & p_i(x) - p_j(x) \\ \kappa(y) - x_t^{\text{pos}}(y) & p_i(y) - p_j(y) \end{bmatrix} \begin{bmatrix} s \\ r \end{bmatrix} = \begin{bmatrix} p_i(x) - \kappa(x) \\ p_i(y) - \kappa(y) \end{bmatrix}. \quad (15)$$

An intersection exists if the pair $(s, r) \in [0, 1]$, and the intersection point can be recovered by substituting either s or r into the respective line-segment equations.

Subsequently, the ray-casting process described above must be performed for each camera-ray K , amongst all sets of possible camera-ray configurations $\mathcal{K}_{\theta_t, x_t^{\text{pos}}, \xi_t}$ and line-segments of the boundary ΔC , to determine which parts of the scene (i.e., points on the region's boundary), are visible through the agent's camera at each time-step $t \in \mathcal{T}$. This makes the ray-casting computation very computationally expensive. Observe here that the agent's position x_t^{pos} is a continuous variable, which depends on the unknown mobility control inputs i.e., Eqn. (3). More importantly the direct implementation of Eqn. (14)-(15) requires the inclusion of non-linear and non-convex constraints in the control problem tackled, which are very hard to be handled efficiently. For this reason, in this work an alternative approximate procedure is employed, which allows the ray-casting functionality described above, to be integrated into a mixed integer quadratic program (MIQP) which in turn can be solved to optimality with readily available optimization tools.

In essence the surveillance area $\mathcal{W}$ is first decomposed into a finite number of cells, and then the agent's visible FoV is computed within each cell, for all possible camera-ray configurations. This enables the proposed approach to learn a set of state-dependent visibility constraints which simulate the physical behavior of camera-rays originating within each cell, and which can be embedded into a mixed integer quadratic program.

Let the rectangular grid $\mathcal{G} = \{c_1, \dots, c_{|\mathcal{G}|}\}$ to denote the discretized representation of the surveillance area $\mathcal{W}$, which is composed of cells $c_i, i = 1, \dots, |\mathcal{G}|$ such that $\bigcup_{i=1}^{|\mathcal{G}|} c_i = \mathcal{G}$. To implement the logical conjunction of the visibility constraint

in Eqn. (14) we introduce 3 binary variables namely b^{S_t} , $b^{x_t^{\text{pos}}}$ and b^K , which are defined as follows:

$$b_{p,m,t}^{S_t} = 1, \text{ iff } \exists t \in \mathcal{T}, m \in \mathcal{M} : p \in S_t(m, x_t^{\text{pos}}), \quad (16)$$

$$b_{c,t}^{x_t^{\text{pos}}} = 1, \text{ iff } \exists t \in \mathcal{T} : x_t^{\text{pos}} \in c, \quad (17)$$

$$b_{c,p}^K = 1, \text{ iff } \exists K \in \mathcal{K}_{\forall \theta, \xi}^c : (K \otimes \Delta C) = L_p, \quad (18)$$

where $\mathcal{M}$ denotes the set of all pairwise combinations $m = (\theta, \xi)$ of $\theta \in \Theta$ and $\xi \in \Xi$. Hence, the total number of FoV configurations is equal to $|\mathcal{M}| = |\Theta| \times |\Xi|$ and thus $S_t(\theta_t, x_t^{\text{pos}}, \xi_t)$ is abbreviated as $S_t(m, x_t^{\text{pos}})$, $m \in \mathcal{M}$. Above, $c \in \mathcal{G}$ is a rectangular cell, part of the surveillance area $\mathcal{W}$ and $p \in \partial \mathcal{P}$ is a point on the region's boundary ∂C which is also connected to some line-segment $L_p \in \Delta C$.

The binary variable $b_{p,m,t}^{S_t}$ in Eqn. (16) is activated when the point p is included inside the $m_{\text{th}} \in \mathcal{M}$ FoV configuration when the agent's position is x_t^{pos} , at time-step t .

Then, the binary variable $b_{c,t}^{x_t^{\text{pos}}}$ in Eqn. (17) shows at which cell c the agent with position x_t^{pos} resides at any point in time t . Finally the constraint in Eqn. (18) indicates with the binary variable $b_{c,p}^K$ if the point p is visible when the agent is inside cell c , where the notation $\mathcal{K}_{\forall \theta, \xi}^c$ indicates the sets of camera-ray configurations for all possible combinations of sensor inputs θ and ξ .

To be more specific, $b_{c,p}^K$ is learned offline, by pre-computing for each cell $c \in \mathcal{G}$ and for each camera-ray $K \in \mathcal{K}_{\forall \theta, \xi}^c$ the visible part of the boundary ΔC via the ray-casting process discussed in the previous paragraph. This is illustrated in Fig. 3(c).

We should note here that the agent position is sampled uniformly n_s times from within each cell c , generating a set of camera-ray configurations $\{^i \mathcal{K}_{\forall \theta, \xi}^c | i = [1, \dots, n_s]\}$. Therefore for each candidate agent position $x_i^{\text{pos}} \in i = [1, \dots, n_s]$ we seek to find the visible points on the boundary.

Thus, a point p is visible at some time-step $t \in \mathcal{T}$, for some combination of sensor input controls $m \in \mathcal{M}$, and agent position x_t^{pos} i.e., $p \in S'_t(m, x_t^{\text{pos}})$ when:

$$\exists m, t : b_{p,m,t}^{S_t} \wedge \left[\bigvee_{c=1}^{|\mathcal{G}|} (b_{c,t}^{x_t^{\text{pos}}} \wedge b_{c,p}^K) \right] = 1. \quad (19)$$

Specifically a point p is visible when both parts of the conjunction in Eqn. (19) are true: a) the point is included inside the agent's sensor FoV which is determined by the agent position x_t^{pos} and sensor control inputs θ and ξ , and encoded by the binary variable $b_{p,m,t}^{S_t}$ and b) the agent with position x_t^{pos} resides inside the cell $c \in \mathcal{G}$ (encoded by $b_{c,t}^{x_t^{\text{pos}}}$) at time t , from which originates a camera-ray $K \in \mathcal{K}_{\forall \theta, \xi}^c$ which first intercepts the line L_p which contains point p (i.e., encoded in the learned variable $b_{c,p}^K$). We should note here that the individual terms of the logical disjunction inside the square brackets are mutually exclusive as the agent cannot occupy two distinct cells at the same time. The ray-casting information encoded in $b_{c,p}^K$ has been learned offline for each cell c , and thus during the optimization phase, we need to find the agent's pose which results in a FoV configuration which observes point p (i.e., $b_{p,m,t}^{S_t}$) and also determine whether the agent resides inside a cell $c \in \mathcal{G}$ (indicated by $b_{c,t}^{x_t^{\text{pos}}}$) from which point p is visible (as indicated by $b_{c,p}^K$). In the next

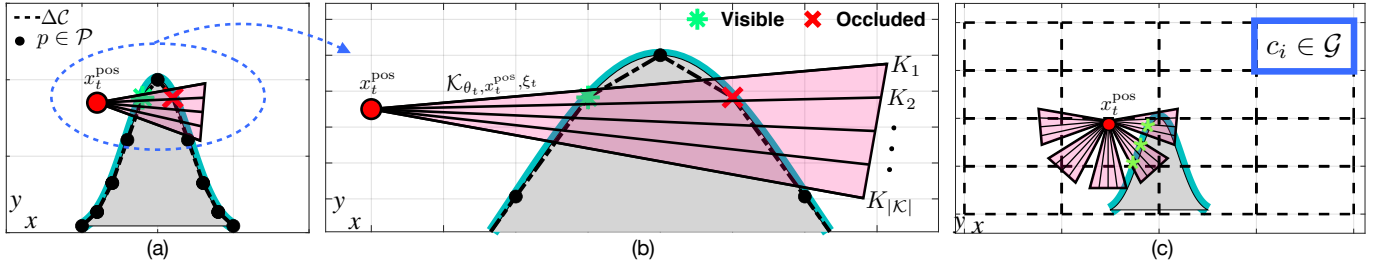


Fig. 3. The figure illustrates how the proposed approach tackles the visibility problem by simulating the physical behavior of camera rays. (a) In the coverage problem investigated in this work, every point $p \in \mathcal{P}$ on the boundary approximation $\Delta\mathcal{C}$ of the region of interest must be covered by the agent's sensor FoV, shown with the pink triangle. (b) The sensor's FoV is composed of a finite number of camera-rays $K_1, \dots, K_{|\mathcal{K}|} \in \mathcal{K}_{\theta_t, x_t^{pos}, \xi_t}$ parameterized by the agent's pose as shown above. A point p is visible through the agent's camera when it resides inside the camera FoV and exists some ray K which first intersects the line segment L_p which contains the point p . (c) We learn a set of visibility constraints by decomposing the surveillance area into a number of cells $c_1, \dots, c_{|\mathcal{G}|}$, thus creating a rectangular grid $\mathcal{G}$. For each cell we learn the visible parts of the scene by checking the intersection of the boundary's line segments with the camera-rays for all possible combinations of camera-ray configurations $\mathcal{K}_{\theta_t, x_t^{pos}, \xi_t}$.

section, we will show how the above constraints are embedded in the proposed coverage controller.

C. Coverage Controller

The complete formulation of the proposed integrated guidance and sensor control coverage planning approach is shown in problem (P2). Specifically, in this section we will show how the high-level problem shown in (P1) is converted into a mixed integer quadratic program (MIQP), which can be solved exactly using readily available optimization tools [48]. To summarize, our goal in (P2) is to jointly find the mobility and sensor control inputs inside a planning horizon, which optimize a mission-specific objective function i.e., $\mathcal{J}_{\text{coverage}}$ subject to visibility and coverage constraints.

1) *Constraints:* Guidance control is achieved by appropriately selecting the agent's mobility control inputs $\mathbf{U}_T = \{u_t : t \in [0, \dots, T-1]\}$ governed by its kinematic constraints i.e. Eqn. (20b). On the other hand, sensor control is achieved via the constraints in Eqn. (20c)-(20d). More specifically, in Eqn. (20c) we construct the FoV configurations for all possible pairwise combinations $\{m = (\theta, \xi)\} \in \mathcal{M}$ of the sensor inputs i.e., rotational angle (θ) and zoom-level (ξ) . Specifically, the continuous variable $\mathcal{V}_m'$ represents a 2 by 3 matrix containing the sensor's FoV vertices for the m_{th} FoV configuration. In essence for each zoom-level $\xi \in \Xi$ (which determines the FoV parameters ϕ and h), the FoV is rotated at the origin for each admissible angle $\theta \in \Theta$, thus creating a total of $|\mathcal{M}| = |\Theta| \times |\Xi|$ FoV configurations indexed by m as shown. Subsequently, all the FoV configurations are translated to the agent's position x_t^{pos} at time t as shown in Eqn. (20d). Therefore, the UAV's pose at each time-step inside the planning horizon is completely specified by the constraints in Eqn.(20b)-(20d).

The constraint in Eqn. (20e), uses the function $\mathcal{L}(\cdot)$ which takes as input the vertices of the m_{th} FoV configuration at some time t i.e., $\mathcal{V}_{m,t}$ and returns a set of linear constraints of the form:

$$A_{n,m,t}^{\mathcal{V}} \times x \leq B_{n,m,t}^{\mathcal{V}}, \quad (21)$$

where $n = [1, \dots, 3]$ and thus $A_{m,t}^{\mathcal{V}}$ is a 3 by 2 matrix, $B_{m,t}^{\mathcal{V}}$ is a 3 by 1 column vector, and x is column vector representing an arbitrary point in $\mathbb{R}^2$. Given two vertices of the triangular FoV, the function $\mathcal{L}(\cdot)$ works by finding

Problem (P2): Coverage Controller

$$\arg \min_{\mathbf{U}_T, \Theta_T, \Xi_T} \mathcal{J}_{\text{coverage}}(\mathbf{X}_T, \mathbf{U}_T, \Theta_T, \Xi_T) \quad (20a)$$

subject to: $t = [1, \dots, T]$

$$x_t = \Phi^t x_0 + \sum_{\tau=0}^{t-1} \Phi^{t-\tau-1} \Gamma u_{\tau} \quad \forall t \quad (20b)$$

$$\mathcal{V}_m' = R(\theta) \mathcal{V}_o(\xi) \quad \forall \{(\theta, \xi)\} \in \mathcal{M} \quad (20c)$$

$$\mathcal{V}_{m,t} = \mathcal{V}_m' + x_t^{pos} \quad \forall m \quad (20d)$$

$$A_{n,m,t}^{\mathcal{V}}, B_{n,m,t}^{\mathcal{V}} = \mathcal{L}(\mathcal{V}_{m,t}) \quad \forall m, n = [1, \dots, 3] \quad (20e)$$

$$A_{n,m,t}^{\mathcal{V}} \times p + \quad (20f)$$

$$b_{n,p,m,t}(M - B_{n,m,t}^{\mathcal{V}}) \leq M \quad \forall n, p, m, t$$

$$3b_{p,m,t}^{S_t} - \sum_{n=1}^3 b_{n,p,m,t} \leq 0 \quad \forall p, m, t \quad (20g)$$

$$A_{k,c}^{\mathcal{G}}, B_{k,c}^{\mathcal{G}} = \mathcal{L}(\mathcal{G}_c) \quad \forall c, k = [1, \dots, 4] \quad (20h)$$

$$A_{k,c}^{\mathcal{G}} \times x_t^{pos} + \quad (20i)$$

$$\tilde{b}_{k,c,t}(M - B_{k,c}^{\mathcal{G}}) \leq M \quad \forall k, c, t$$

$$4b_{c,t}^{x_t^{pos}} - \sum_{k=1}^4 \tilde{b}_{k,c,t} \leq 0 \quad \forall c, t \quad (20j)$$

$$\sum_{m=1}^{|\mathcal{M}|} \mathcal{F}_{m,t}^{\text{sel}} = 1 \quad \forall t \quad (20k)$$

$$b_{p,m,t}^{S_t'} = \mathcal{F}_{m,t}^{\text{sel}} \wedge \quad (20l)$$

$$b_{p,m,t}^{S_t} \wedge \left[\bigvee_{c=1}^{|\mathcal{G}|} (b_{c,t}^{x_t^{pos}} \wedge b_{c,p}^{\mathcal{K}}) \right] \quad \forall p, m, t$$

$$\sum_{t=1}^T \sum_{m=1}^{|\mathcal{M}|} b_{p,m,t}^{S_t'} \geq 1 \quad \forall p \quad (20m)$$

$$x_0, x_t^{pos} \notin \psi, \quad \forall \psi \in \Psi \quad (20n)$$

$$x_0, x_t \in \mathcal{X}, \quad u_t \in \mathcal{U}, \quad \theta \in \Theta, \quad \xi \in \Xi$$

$$m \in [1, \dots, |\mathcal{M}|], p \in [1, \dots, |\mathcal{P}|], c \in [1, \dots, |\mathcal{G}|]$$

$$\mathcal{F}_{m,t}^{\text{sel}}, b_{n,p,m,t} \in \{0, 1\}$$

$$b_{p,m,t}^{S_t}, \tilde{b}_{k,c,t}, b_{c,t}^{x_t^{pos}}, b_{p,m,t}^{S_t'}, b_{c,p}^{\mathcal{K}} \in \{0, 1\}$$

the equation of the line which passes through these two vertices. In total 3 line equations are constructed which fully specify the convex hull of the triangular FoV i.e., a point $x \in \mathbb{R}^2$ belongs to the convex hull of the triangular FoV iff $A_{n,m,t}^\mathcal{V} \times x \leq B_{n,m,t}^\mathcal{V}, \forall n = [1, \dots, 3]$. More specifically, $A_{m,t}^\mathcal{V}$ and $B_{m,t}^\mathcal{V}$ contain the coefficients of the lines which form the triangular FoV, where in particular the matrix $A_{n,m,t}^\mathcal{V}$ contains the outward normal to the n_{th} line. We can write the line equation which passes from two FoV vertices i.e., (v_x^1, v_y^1) and (v_x^2, v_y^2) as $ax + by = c$, where the coefficients $a = v_y^1 - v_y^2$ and $b = v_x^2 - v_x^1$ define the normal on the line i.e., $\vec{n} = (a, b)$ and $c = v_x^2 v_y^1 - v_x^1 v_y^2$, thus $A_{1,m,t}^\mathcal{V} = [a, b]$ and $B_{1,m,t}^\mathcal{V} = c$. To summarize, we can determine whether an arbitrary point $x \in \mathbb{R}^2$ is included inside the sensor's FoV by checking if the system of linear inequalities in Eqn. (21) is satisfied.

The constraint in Eqn. (21) is implemented as shown in (P2) with the constraints shown in Eqn. (20f)-(20g) i.e.,:

$$A_{n,m,t}^\mathcal{V} \times p + b_{n,p,m,t}(M - B_{n,m,t}^\mathcal{V}) \leq M, \quad \forall n, p, m, t,$$

$$3b_{p,m,t}^{S_t} - \sum_{n=1}^3 b_{n,p,m,t} \leq 0, \quad \forall p, m, t.$$

In essence, these constraints allows us to check whether some point $p \in \mathcal{P}$ is covered by the agent's sensor i.e., $p \in \mathcal{S}_t(m, x_t^{\text{pos}})$, when the agent is at position x_t^{pos} at time-step t , and the sensor's FoV is at the m_{th} configuration. As we discussed earlier the matrices $A_{m,t}^\mathcal{V}$ and $B_{m,t}^\mathcal{V}$ contain the coefficients of the sensor's FoV for every possible configuration $m \in \mathcal{M}$ and time-step $t = [1, \dots, T]$. With this in mind, we use the binary variable $b_{n,p,m,t}$ to decide whether some point $p \in \mathcal{P}$, resides inside the negative half-plane which is created by the n_{th} line, of the m_{th} FoV configuration at time t . When this happens the $b_{n,p,m,t}$ is activated and the inequality in Eqn. (20f) is satisfied i.e., $A_{n,m,t}^\mathcal{V} \times p \leq B_{n,m,t}^\mathcal{V}$. On the other hand when $A_{n,m,t}^\mathcal{V} \times p > B_{n,m,t}^\mathcal{V}$, the constraint in Eqn. (20f) is satisfied by setting $b_{n,p,m,t} = 0$ and using the large positive constant M . Subsequently, the constraint in Eqn. (20g) uses the binary variable $b_{p,m,t}^{S_t}$ to determine whether the point p resides at time t inside the m_{th} configuration of the FoV. Thus $b_{p,m,t}^{S_t}$ is activated only when $\sum_{n=1}^3 b_{n,p,m,t} = 3$, which signifies that the point p is covered by the sensor's FoV.

Similarly, the next 3 constraints shown in Eqn. (20h) - (20j) (also shown below) use the same principles discussed above, to determine whether the agent with position x_t^{pos} resides inside cell $c \in \mathcal{G}$ at time t .

$$A_{k,c}^\mathcal{G}, B_{k,c}^\mathcal{G} = \mathcal{L}(\mathcal{G}_c), \quad \forall k = [1, \dots, 4], c,$$

$$A_{k,c}^\mathcal{G} \times x_t^{\text{pos}} + \tilde{b}_{k,c,t}(M - B_{k,c}^\mathcal{G}) \leq M, \quad \forall k, c, t,$$

$$4b_{c,t}^{x_t^{\text{pos}}} - \sum_{k=1}^4 \tilde{b}_{k,c,t} \leq 0, \quad \forall c, t.$$

Specifically, the constraint Eqn. (20h) uses the function $\mathcal{L}(\mathcal{G}_c)$ on the grid cells and returns in the matrices $A_{k,c}^\mathcal{G}$ and $B_{k,c}^\mathcal{G}$, the coefficients of the linear inequalities which define the convex hull of every cell $c \in \mathcal{G}$ in the grid. A point $x \in \mathbb{R}^2$ resides inside a rectangular cell c iff $A_c^\mathcal{G} \times x \leq B_c^\mathcal{G}$, where $A_c^\mathcal{G}$ is a 4 by 2 matrix and $B_c^\mathcal{G}$ is a 4 by 1 column vector. Therefore, the constraint in Eqn. (20i) uses the binary variable

$\tilde{b}_{k,c,t}$ to determine whether the agent's position satisfies the k_{th} inequality i.e., $A_{k,c}^\mathcal{G} \times x_t^{\text{pos}} \leq B_{k,c}^\mathcal{G}, \forall k = [1, \dots, 4]$. Subsequently, the binary variable $b_{c,t}^{x_t^{\text{pos}}}$ in Eqn. (20j) is activated when x_t^{pos} resides inside cell c at time t . Next, we make use of the constraint in Eqn. (20k) i.e.,

$$\sum_{m=1}^{|\mathcal{M}|} \mathcal{F}_{m,t}^{\text{sel}} = 1, \quad \forall t,$$

to account for the fact that at any point in time t , only one FoV configuration is active. In other words we would like to prevent the scenario where more than one sets of sensor input controls are applied and executed at some particular time-step t . To do this we define the binary variable $\mathcal{F}^{\text{sel}}$ consisting of $|\mathcal{M}|$ rows and T columns, such that $\mathcal{F}_{m,t}^{\text{sel}} \in \{0, 1\}, \forall m \in \mathcal{M}, t \in \mathcal{T}$, and we require that at each time-step t only one FoV configuration is active by enforcing the sum of each column to be equal to one, as shown in Eqn. (20k).

We can now determine whether some point p belongs to the visible FoV as:

$$b_{p,m,t}^{S_t'} = \mathcal{F}_{m,t}^{\text{sel}} \wedge b_{p,m,t}^{S_t} \wedge \left[\bigvee_{c=1}^{|\mathcal{G}|} (b_{c,t}^{x_t^{\text{pos}}} \wedge b_{c,p}^\mathcal{K}) \right], \quad \forall p, m, t,$$

where the binary variable $b_{p,m,t}^{S_t'}$ is activated when the point $p \in \mathcal{P}$ is visible at time t , and specifically resides inside the m_{th} FoV configuration i.e., $p \in \mathcal{S}_t'(m, x_t^{\text{pos}})$. As it is shown, in the conjunction above, the binary variable $b_{p,m,t}^{S_t}$ captures the sensor's pose, which is determined by the agent's position x_t^{pos} at time t and sensor orientation given by the m_{th} FoV configuration, and determines whether the point p resides inside the sensor's FoV i.e., constraints in Eqn. (20f)-(20g). Then the binary variable $b_{c,t}^{x_t^{\text{pos}}}$ checks if the agent at time t resides inside a particular cell c i.e., Eqn. (20h) - (20j) and together with the learned variable $b_{c,p}^\mathcal{K}$ (as discussed in Sec. V-B), determine whether the point p is visible given that the agent is at cell c . To summarize, a point p belongs to the visible FoV when there exists some cell c from which the point p is visible and at time-step t the agent's position x_t^{pos} resides inside that cell c and exists a FoV configuration m such that point p is covered by the agent's sensor at time t . Lastly, in order to make sure that only one FoV configuration is active at each time-step (i.e., one set of sensor controls is applied), we use the FoV selector $\mathcal{F}_{m,t}^{\text{sel}}$ as shown in Eqn. (20k).

Finally, we can ensure that during the planning horizon, every point $p \in \mathcal{P}$ will be covered at least once by the agent's sensor via the constraint in Eqn. (20m):

$$\sum_{t=1}^T \sum_{m=1}^{|\mathcal{M}|} b_{p,m,t}^{S_t'} \geq 1, \quad \forall p,$$

where we require that there exists at least one FoV configuration (i.e., one set of sensor input controls) $m \in \mathcal{M}$ at some time-step $t \in \mathcal{T}$, such that the binary variable $b_{p,m,t}^{S_t'}$ is activated i.e., $b_{p,m,t}^{S_t'} = 1$ for every point $p \in \mathcal{P}$. Therefore, determining the agent's mobility and sensor control inputs such that all points are included inside the agent's visible FoV at some time-step during the planning horizon.

The last constraint shown in Eqn. (20n), implements the obstacle avoidance constraints as discussed in Sec. V-A. To summarize, we have derived a set of constraints i.e., Eqn.(20b)-(20n) which jointly account for the agent's kinematic and sensing model, integrate visibility constraints into the coverage planning problem and guarantee full coverage inside the planning horizon, given that a feasible solution exists. Finally, we should note that the formulation of the coverage controller shown in problem (P2) can be easily extended to tackle the problem in 3D environments, however this has been left for future works. Next, we discuss in detail the design of the objective function, which can be used in order to optimize a set of mission-related performance criteria.

2) *Objectives*: The problem of integrated UAV guidance and sensor control which is studied in this work is a core component for many applications and tasks including surveillance, emergency response, and search-and-rescue missions. Take for instance a UAV-based search-and-rescue mission where the objective is to search an area of interest and locate as quickly as possible people in need. In such scenarios, the mission's completion time is of the highest importance for saving lives. In other cases the UAV's efficient battery utilization is imperative for the success of the mission. Motivated by the objectives and requirements discussed above we design a multi-objective cost function $\mathcal{J}_{\text{coverage}}$ to allow for the characterization of several mission-related optimality criteria, and trade-offs amongst these. In this work, $\mathcal{J}_{\text{coverage}}$ is composed of a set of sub-objectives, which sometimes might be competing. More specifically, we define the overall coverage objective $\mathcal{J}_{\text{coverage}}$ as:

$$\mathcal{J}_{\text{coverage}} = (w_1 J_1 + w_2 J_2 + \dots + w_n J_n), \quad (22)$$

where J_i represents the i_{th} sub-objective and w_i is the tuning weight associated with the i_{th} sub-objective. Therefore, the weights are used to emphasize or deemphasize the importance of each sub-objective according to the mission goals. Next we design several possible sub-objectives which can be used to drive an efficient coverage planning mission.

Mission completion time (J_1): As discussed earlier one of the most important objectives in a coverage planning scenario is the minimization of the mission's completion time. In other words, we are interested in finding the optimal UAV control inputs (i.e., mobility and sensor controls), which when executed allow the agent to cover all points of interest $p \in \mathcal{P}$ as quickly as possible, thus minimizing the time needed to conduct a full coverage of the region of interest. This can be defined as follows:

$$J_1 = \sum_{p=1}^{|\mathcal{P}|} \sum_{m=1}^{|\mathcal{M}|} \sum_{t=1}^T \left(b_{p,m,t}^{S'_t} \times \frac{t}{T} \right). \quad (23)$$

In essence by minimizing J_1 , we are minimizing the product of the binary variable $b_{p,m,t}^{S'_t}$ with the factor (t/T) , over the planning horizon of length T , for all points $p \in \mathcal{P}$ and FoV configurations $m \in \mathcal{M}$. Effectively, (t/T) in Eqn. (23) acts as a weight to $b_{p,m,t}^{S'_t}$ which increases over time. This drives the optimizer to find the optimal mobility and sensor control inputs which allow the agent to cover all points $p \in \mathcal{P}$ as quickly as possible or equivalently $b_{p,m,t}^{S'_t}$ is activated for each point at the earliest possible time-step. Finally, we should note here

that the agent's control inputs are directly linked with $b_{p,m,t}^{S'_t}$, since the agent's pose is jointly determined by its mobility and sensor controls i.e., Eqn. (20c)-(20d) and also $b_{p,m,t}^{S'_t}$ is only activated when point p is visible.

Energy Efficiency (J_2): Energy-aware operation is another essential objective for various applications. In essence we are interested in prolonging the UAV's operation time (i.e., minimizing the battery drain), by optimizing the UAV's mobility control inputs (i.e., the amount of force applied), thus generating energy-efficient coverage trajectories. Although, the proposed coverage planning formulation, does not directly uses a battery model for the UAV, it is assumed that the UAV mobility control inputs are directly linked with the battery usage. Therefore, energy efficient coverage planning can be achieved by appropriately selecting the UAV's mobility control inputs. Specifically, it is assumed that the generation of smooth UAV trajectories with reduced abrupt changes in direction and speed can lead to improved battery usage, thus we define the energy-aware coverage planning sub-objective as:

$$J_2 = \sum_{t=1}^{T-1} \|u_t - u_{t-1}\|_2^2 + \sum_{t=0}^{T-1} |u_t|, \quad (24)$$

where we minimize a) the sum of deviations between consecutive control inputs and b) the cumulative magnitude of the absolute value of individual controls, thus leading to energy optimized coverage planning.

Sensor Control Effort (J_3): The last objective aims to minimize the sensor deterioration due to excessive and/or improper usage i.e., by reducing the utilization of the gimbaled sensor during the mission. This allows us to maintain the sensor's healthy status and prolong its lifespan. For this reason, we define as sensor control effort the deviation between successive FoV configurations and thus J_3 is defined by:

$$J_3 = \sum_{t=1}^{T-1} \sum_{m=1}^{|\mathcal{M}|} \|\mathcal{F}_{m,t+1}^{\text{sel}} - \mathcal{F}_{m,t}^{\text{sel}}\|_2^2, \quad (25)$$

which favors the generation of coverage trajectories which achieve full coverage with minimum gimbal utilization.

To summarize, in this section we have described a set (not exhaustive) of sub-objectives, which can be used to compose the overall multi-objective cost function $\mathcal{J}_{\text{coverage}}$ for the coverage path planning problem we examine in this work. These sub-objectives can be prioritized depending on the problem requirements, while new ones can be added according to the mission specifications. We should mention here that the objective J_3 described above can also be incorporated into the objective J_2 i.e., energy efficiency, to account for the overall energy expenditure (i.e., energy expenditure from motion control and from gimbal control) of the system.

VI. EVALUATION

A. Simulation Setup

In order to evaluate the proposed integrated guidance and gimbal control coverage approach we have conducted a thorough simulation analysis. More specifically, the evaluation is divided into three parts. In the first part we investigate the effect of the visibility constraints on the coverage planning behavior. In the second part of the evaluation we showcase

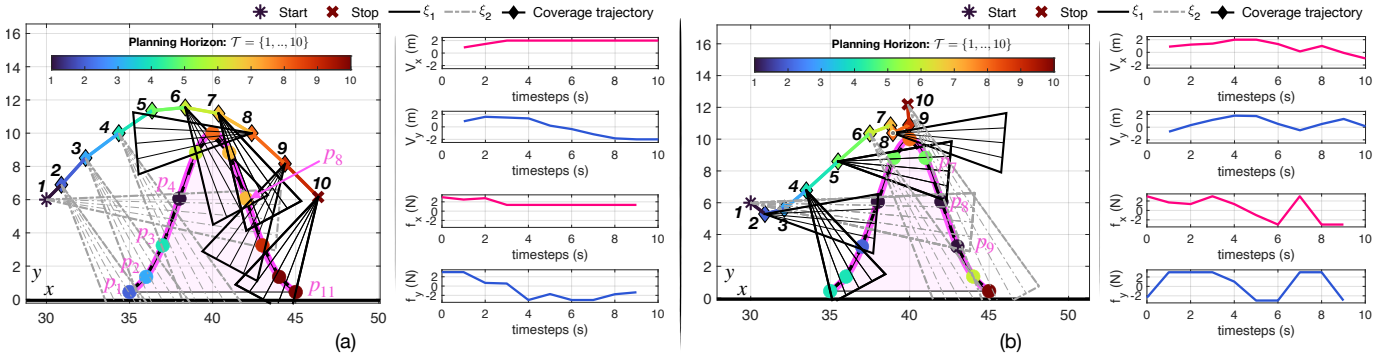


Fig. 4. The figure illustrates the impact of the visibility constraints on the generated coverage trajectories. (a) Visibility constraints are enabled, (b) Visibility constraints disabled. The visibility constraints simulate the physical behavior of camera-rays, therefore allowing the agent to distinguish between visible and occluded points.

the proposed approach for various mission related optimality criteria, and finally in the third part, we analyze the generated coverage trajectories for various parameters of the inputs.

The simulation setup used for the evaluation of the proposed approach is as follows: The agent kinematics are expressed by Eqn. (1) with $\delta t = 1s$, agent mass $m = 3.35kg$ and drag coefficient $\eta = 0.2$. For demonstration purposes, the control input (i.e., input force) $u_t = [f_t(x), f_t(y)]$ is bounded in each dimension according to $|f_t(x|y)| \leq 3N$, and the agent velocity is bounded according to $|\nu_t(x|y)| \leq 2m/s$. The agent's FoV angle is set to $\phi = 30deg$, and the sensing range $h = 7m$. The camera zoom-levels are set to $\Xi = \{1, 2\}$, thus the camera characteristics for zoom-level $\xi = 1$ and $\xi = 2$ are given by $(\phi = 30, h = 7)$ and $(\phi = 15, h = 14)$ respectively. In total we consider 4 camera rotation angles i.e., $\Theta = \{-85, -28, 28, 85\}$ which are used to rotate the camera FoV according to Eqn. (7), leading to a total of 8 possible FoV configurations i.e., $|\mathcal{M}| = 8$. We have used the ray-tracing procedure with $|\mathcal{K}| = 5$ camera-rays in a surveillance region $\mathcal{W}$ that has a total area of $60 \times 20m^2$. The region/object of interest $\mathcal{C}$ is represented by a bell-shaped curve (as illustrated in Fig. 2), given by $f(x) = a \times \exp\left(\frac{-(x-b)^2}{2c^2}\right)$ with a, b and c set to 10, 40 and 2 respectively. The region of interest is assumed to be non traversable, and thus we are interested in generating coverage trajectories to cover a total of 11 points $\mathcal{P} = \{p_1, \dots, p_{11}\}$ sampled from the region's boundary $\partial\mathcal{C}$. Finally, we should mention that the visibility constraints have been learned on a discretized representation $\mathcal{G}$ of the surveillance area, where $\mathcal{G}$ contains 16 square cells (as illustrated in Fig. 3) of size $10m \times 5m$. To summarize, the visibility constraints have been learned according to Sec. V-B, with $|\mathcal{M}| = 8, |\mathcal{P}| = 11, |\mathcal{G}| = 16, n_s = 15$ and $|\Delta\mathcal{C}| = 11$. The results have been obtained with Gurobi v9 solver, running on a 2.5GHz laptop computer.

B. Results

1) **Visibility Constraints:** With the first experiment, shown in Fig. 4, we aim to investigate the impact of the visibility constraints on the trajectory generation process, and gain insights on the coverage planning behavior of the proposed approach. Specifically, Fig. 4(a) shows the coverage trajectory, agent velocity, and mobility control inputs, within a planning horizon of $T = 10$ time-steps when the visibility constraints

are enabled, whereas Fig. 4(b) shows the exact same scenario with the visibility constraints disabled. The region of interest is shaded in pink (i.e., the bell-shaped curve), the agent's trajectory is marked with $-\diamond-$, the agent's start and stop positions are marked with $\star$ and $\times$ respectively, and the points on the boundary to be covered are marked with $\bullet$. The figure also shows the FoV configuration at each time-step, indicated by the isosceles triangles, where the black solid lines and the gray dashed lines correspond to the first (ξ_1) and second (ξ_2) zoom-levels respectively. Finally, the agent's trajectory and the points to be covered are color-coded according to the time-step which are observed, as shown in the figure legend. Therefore, according to Fig. 4(a), point p_1 (colored dark blue) is covered at time-step $t = 2$, point p_2 (colored light blue) is covered at $t = 3$, point p_3 (colored light green) is included inside the agent's FoV at $t = 4$, point p_4 (colored black) is the first point to be viewed by the agent at time-step $t = 1$, and so on and so forth. We should mention that for this experiment we have set $\mathcal{J}_{coverage} = 1$ (i.e., we are minimizing a constant), and thus the solely goal of the optimization in this experiment is to satisfy the coverage constraints (i.e., cover all points).

As we can observe from Fig. 4(a), the agent starts from the left side of the bell-shaped curve, and appropriately selects its mobility and sensor control inputs which achieve full coverage. More importantly, we can observe that although the FoV can extend all the way through the object of interest (e.g., at $t = 1$ points p_4 and p_8 are inside the FoV), the use of visibility constraints, which simulate the physical behavior of camera-rays, allow the identification of occlusions (e.g., at $t = 1$ point p_8 is occluded, and becomes visible at $t = 7$ as shown). Therefore, the agent can identify at each time-step which points are visible through its camera and plan its coverage trajectory as needed. For this reason, in this test the agent goes over the bell-shaped curve, towards the other side of the curve, in order to cover the occluded points i.e., at $t = 7$ points p_7 and p_8 are covered, at $t = 8$ point p_6 is covered, at $t = 9$ the point p_9 is covered and the remaining points (p_{10} and p_{11}) are covered at time-step t_{10} . In addition, it is shown that the obstacle avoidance constraints restrict the agent from passing through the object of interest. On the other hand, observe from Fig. 4(b) that when the visibility constraints are disabled, the agent cannot distinguish between visible and occluded parts of the scene i.e., at $t = 1$ the points p_8 and p_9 (colored black) are occluded but observed, similarly point p_7 (colored green) is

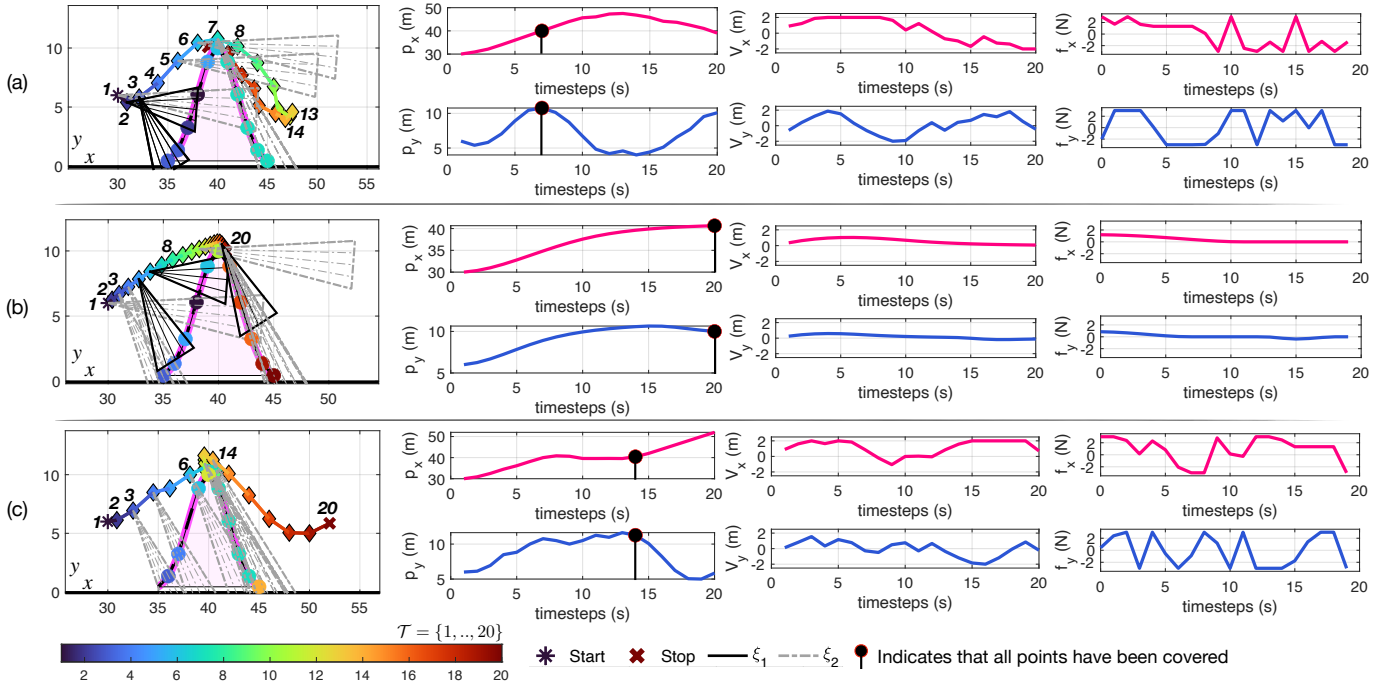


Fig. 5. The figure illustrates the generated coverage planning trajectories over a planning horizon of length 20 time-steps, for 3 different sub-objectives. (a) Mission completion time (J_1), (b) Energy efficiency (J_2), and (c) Sensor control effort (J_3).

occluded at $t = 5$ but it is observed as shown in the figure. This is because the sensor's visible FoV is not modeled adequately without the use of the visibility constraints, and as a result the generated trajectory does not resembles a realistic coverage path.

2) Coverage Objectives: The purpose of the next experiment is to investigate in more detail different coverage planning strategies by optimizing the sub-objectives discussed in Sec. V-C. More specifically, we will show how the coverage plan changes when optimizing the mission completion time (J_1), the energy efficiency (J_2), the sensor control effort (J_3), and a weighted combination of those. Figure 5 shows the coverage planning trajectories along with the agent position, velocity, and input force over time for the same scenario, when optimizing the individual sub-objectives J_1 , J_2 , and J_3 , within a planning horizon of length 20 time-steps. As it can be seen from Fig. 5(a), when optimizing the mission completion time (J_1), the set of 11 points $\mathcal{P}$ is fully covered at time-step 7. Note that the agent's trajectory and the points to be covered are color-coded according to the time-step at which the coverage occurs. In this sense the last point in Fig. 5(a) which is color-coded light green is covered at time-step 7. The time-step at which all points are covered is also shown in the agent position plot, and marked with a black circle. We should point out here that for visual clarity the graphs show the FoV configurations only for the time-instances for which a point is included inside the sensor's FoV. Next, Fig. 5(b) shows the coverage trajectory for the sub-objective which minimizes the agent's energy expenditure. As it is shown, in this scenario the applied input force which is used for guidance is driven to zero over time. In addition we can observe that the agent moves in small increments (i.e., consecutive positions are close to each other) as opposed to the previous scenario, also indicated by the velocity plot. Also, observe how the agent utilizes its

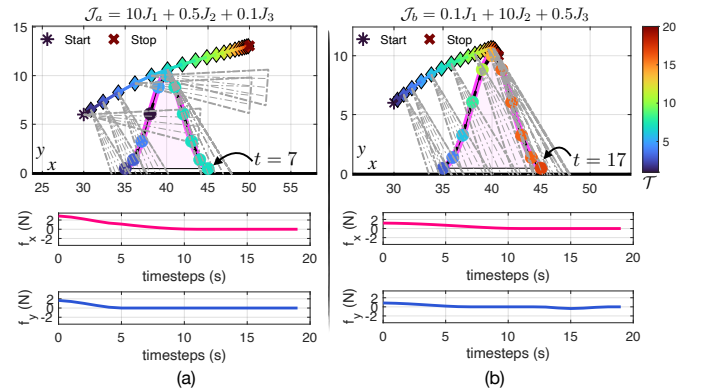


Fig. 6. The figure illustrates the coverage planning trajectories over a planning horizon of 20 time-steps, for two different multi-objective cost functions. (a) $\mathcal{J}_a = 10J_1 + 0.5J_2 + 0.1J_3$, and (b) $\mathcal{J}_b = 0.1J_1 + 10J_2 + 0.5J_3$.

sensor to achieve full coverage (which occurs at time-step 20), while optimizing for energy efficiency. In this scenario, 6 out of the 8 camera FoV configurations are used over the planning horizon, until full coverage is achieved. Finally, Fig. 5(c), shows that the minimization of the sensor control effort (J_3), forces the agent to complete the mission by utilizing just 1 out of the 8 possible FoV configurations. Essentially, in this scenario the camera remains fixed as shown in the figure. In this scenario full coverage was achieved at time-step 14 as shown in the graph. Also observe that in this scenario the mobility controls fluctuate significantly as opposed to the previous scenario.

These three different sub-objectives can be incorporated into a multi-objective cost function, as discussed in Sec. V-C, and by adjusting the emphasis given to each sub-objective the desired coverage behavior can be obtained as shown

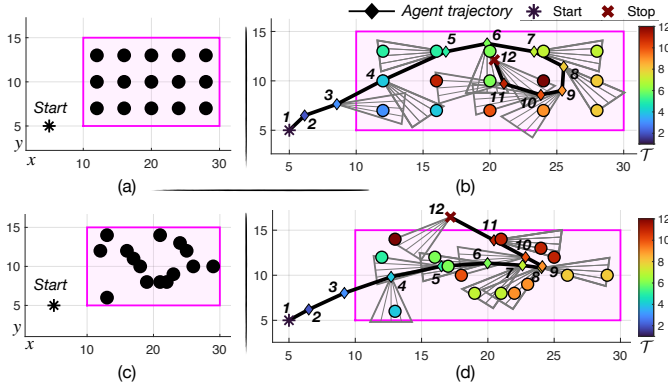


Fig. 7. The figure illustrates the coverage plan for two different scenarios which exhibit a traversable area of interest, indicated by the pink shaded rectangle. (a)(b) The agent's coverage plan when the area of interest is approximated by $|\mathcal{P}| = 15$ equally spaced points shown as $\bullet$, (c)(d) The agent's trajectory used to cover $|\mathcal{P}| = 15$ points uniformly sampled from the area of interest. The numbers on the trajectory indicate time-steps.

in the next experiment i.e., Fig. 6. More specifically, we have run the coverage planning with two different multi-objective cost functions i.e., $\mathcal{J}_a = 10J_1 + 0.5J_2 + 0.1J_3$ and $\mathcal{J}_b = 0.1J_1 + 10J_2 + 0.5J_3$ shown in Fig. 6(a) and Fig. 6(b) respectively. In both scenarios we aim to optimize for energy efficiency by including, and appropriately weighting sub-objective J_2 in the multi-objective cost function. As it is illustrated the applied force is minimized and driven to zero for both scenarios. However, due to the higher emphasis given to J_2 in the second scenario, $\mathcal{J}_b$ optimizes energy savings more aggressively also indicated by the generated coverage trajectory. Observe that in objective $\mathcal{J}_a$ greater emphasis is given to mission completion time (i.e., J_1), which results in faster coverage (at time-step 7). On the other hand in $\mathcal{J}_b$ full coverage is achieved at time-step 17. Finally, we observe that by weighting more on the sub-objective J_3 in $\mathcal{J}_b$, allows the agent minimize the gimbal rotations by maintaining the camera fixed and at the same time optimize for energy efficiency as shown in Fig. 6(b).

Our next experiment shows the behavior of the proposed coverage planning approach for a traversable region of interest, indicated by the pink shaded rectangle shown in Fig. 7. In this scenario, without loss of generality, we assume an obstacle-free region of interest. More specifically, in Fig. 7(a) the region of interest is approximated with a total of $|\mathcal{P}| = 15$ equally spaced points (shown as $\bullet$), which need to be covered by the agent, initially located at $(x, y) = (5, 5)$ (shown as $\star$). Figure 7(b) shows the agent's coverage trajectory when optimizing the coverage objective $\mathcal{J} = 0.1J_1 + J_2$ inside a planning horizon of length $T = 12$ time-steps. In this scenario, the agent's FoV angle ϕ and sensing range h are set to 35deg, and 5m respectively, and the camera can be rotated in 5 ways i.e., $\Theta = \{-85, -42.5, 0, 42.5, 85\}$. In order to make the illustrations easier to read, in this scenario we do not make use of the zoom functionality i.e., $\Xi = \{1\}$ which leads to a total of 5 possible FoV configurations i.e., $|\mathcal{M}| = 5$. As shown in Fig. 7(b) the agent's mobility control inputs and camera rotations are appropriately selected and optimized according to the coverage objective $\mathcal{J}$ to achieve full coverage of the region of interest, i.e., as shown in the figure the points are

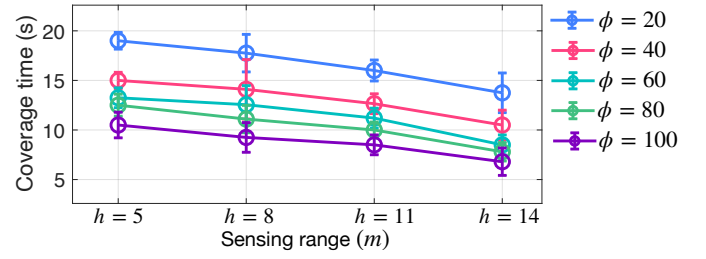


Fig. 8. The figure shows the mission completion time for various parameter configurations of the FoV, i.e., $\phi \in \{20, 40, 60, 80, 100\}$ and $h \in \{5, 8, 11, 14\}$.

colored-coded according to the time-step they are observed by the agent. Next, Fig. 7(c)(d) shows the same setup for 15 points which are sampled uniformly from the region of interest. As shown in Fig. 7(d) the generated coverage plan enables the agent to cover all 15 points over the planning horizon.

Our last experiment investigates how various configurations of the FoV parameters ϕ and h (i.e., opening angle and sensing range) affect the coverage performance and more specifically the mission completion time. For this experiment, we use the simulation setup discussed in the beginning of this section, with $\Xi = \{1\}$ and optimizing J_1 . We perform 50 Monte Carlo trials, where we sample the agent position randomly within a disk of radius 5m centered at $(x, y) = (30, 6)$ for each combination of the parameters $(\phi, h) \in \Phi \times H$, where $\Phi \in \{20, 40, 60, 80, 100\}$ deg and $H = \{5, 8, 11, 14\}$ m. In particular, Fig. 8 shows the average coverage completion time for all configurations of the parameters. As we can observe the time needed for full coverage drops from 20sec to below 15sec when we increase the sensing range from $h = 5$ m to $h = 14$ m for the scenario where the angle opening is set to $\phi = 20$ deg. Similarly, for fixed sensing range set at $h = 5$ m, the coverage time reduces approximately by 50% as the angle opening increases from 20deg to 100deg. Overall, as we can observe from Fig. 8 the mission completion time improves as the FoV increases both in terms of ϕ and h . Finally, observe that the extremities of the bell shaped object of interest are 10m tall (at the very top) and 10m wide (at the base), which means that without the use of the proposed visibility constraints and with a camera configuration of $\phi = 100$ deg and $h = 14$ m, the UAV agent could have observed all points and finish the mission in a couple of seconds, as the entire object of interest would have been included inside its sensor's footprint. However, this erroneous behavior is prevented in this work by the integration of ray-casting into the proposed coverage controller.

VII. CONCLUSION

In this work we have proposed an integrated guidance and gimbal control approach for coverage path planning. In the proposed approach the UAV's mobility and sensor control inputs are jointly optimized to achieve full coverage of a given region of interest, according to a specified set of optimality criteria including mission completion time, energy efficiency and sensor control effort. We have devised a set of visibility constraints in order to integrate ray-casting to the proposed coverage controller, thus allowing the generation of optimized coverage trajectories according to the sensor's visible field-of-view. Finally, we have demonstrated how the constrained

optimal control problem tackled in this work can be formulated as a mixed integer quadratic program (MIQP), and solved using off-the-shelf tools. Extensive numerical experiments have demonstrated the effectiveness of the proposed approach. Future directions include the extension of the proposed approach in 3D environments, the evaluation of the proposed approach in real-world settings, and extensions to multiple agents.

ACKNOWLEDGMENTS

This work is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE), from the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development, and from the Cyprus Research and Innovation Foundation under grant agreement EXCELLENCE/0421/0586 (GLIMPSE).

REFERENCES

- [1] J. Moon, S. Papaioannou, C. Laoudias, P. Kolios, and S. Kim, "Deep reinforcement learning multi-uav trajectory control for target tracking," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15 441–15 455, 2021.
- [2] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "Decentralized search and track with multiple autonomous agents," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 909–915.
- [3] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Computer Communications*, vol. 156, pp. 1–10, 2020.
- [4] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "A Cooperative Multi-Agent Probabilistic Framework for Search and Track Missions," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 2, pp. 847–858, 2021.
- [5] —, "Towards Automated 3D Search Planning for Emergency Response Missions," *Journal of Intelligent & Robotic Systems*, vol. 103, no. 1, pp. 1–19, 2021.
- [6] —, "3D Trajectory Planning for UAV-based Search Missions: An Integrated Assessment and Search Planning Approach," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2021, pp. 517–526.
- [7] U. R. Mogili and B. Deepak, "Review on application of drone systems in precision agriculture," *Procedia Computer Science*, vol. 133, pp. 502–509, 2018.
- [8] D. N. Das, R. Sewani, J. Wang, and M. K. Tiwari, "Synchronized truck and drone routing in package delivery logistics," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [9] F. Afghah, A. Razi, J. Chakareski, and J. Ashdown, "Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops*, 2019, pp. 835–840.
- [10] S. Papaioannou, P. Kolios, C. G. Panayiotou, and M. M. Polycarpou, "Cooperative simultaneous tracking and jamming for disabling a rogue drone," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 7919–7926.
- [11] S. Papaioannou, P. Kolios, and G. Ellinas, "Downing a rogue drone with a team of aerial radio signal jammers," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2555–2562.
- [12] P. Valianti, S. Papaioannou, P. Kolios, and G. Ellinas, "Multi-agent coordinated close-in jamming for disabling a rogue drone," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.
- [13] A. Gasparetto, P. Boscaroli, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," *Motion and operation planning of robotic systems*, pp. 3–27, 2015.
- [14] T. M. Cabreira, L. B. Brisolara, and P. R. Ferreira Jr, "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, no. 1, p. 4, 2019.
- [15] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Efficient complete coverage of a known arbitrary environment with applications to aerial operations," *Autonomous Robots*, vol. 36, no. 4, pp. 365–381, 2014.
- [16] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and Service Robotics*. Springer, 1998, pp. 203–209.
- [17] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, 2002.
- [18] M. N. A. Wahab, S. Nefti-Meziani, and A. Atiyabi, "A comparative review on mobile robot path planning: Classical or meta-heuristic methods?" *Annual Reviews in Control*, vol. 50, pp. 233–252, 2020.
- [19] E. Acar, H. Choset, and J. Y. Lee, "Sensor-based coverage with extended range detectors," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 189–198, 2006.
- [20] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 2001, pp. 27–32.
- [21] R. Mannadiar and I. Rekleitis, "Optimal coverage of a known arbitrary environment," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 5525–5530.
- [22] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, vol. 2, 2001, pp. 1927–1933 vol.2.
- [23] —, "Spiral-STC: an on-line coverage algorithm of grid environments by a mobile robot," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, vol. 1, 2002, pp. 954–960 vol.1.
- [24] N. Agmon, N. Hazon, and G. A. Kaminka, "The giving tree: constructing trees for efficient offline and online multi-robot coverage," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2, pp. 143–168, 2008.
- [25] N. Karapetyan, K. Benson, C. McKinney, P. Taslakian, and I. Rekleitis, "Efficient multi-robot coverage of a known environment," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1846–1852.
- [26] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.
- [27] A. Janchiv, D. Batsaikhan, B. Kim, W. G. Lee, and S.-G. Lee, "Time-efficient and complete coverage path planning based on flow networks for multi-robots," *International Journal of Control, Automation and Systems*, vol. 11, no. 2, pp. 369–376, 2013.
- [28] G. Cannata and A. Sgorbissa, "A minimalist algorithm for multirobot continuous coverage," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 297–312, 2011.
- [29] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.
- [30] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [31] L. C. Pimenta, V. Kumar, R. C. Mesquita, and G. A. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 3947–3952.
- [32] A. V. Savkin and H. Huang, "Navigation of a network of aerial drones for monitoring a frontier of a moving environmental disaster area," *IEEE Systems Journal*, vol. 14, no. 4, pp. 4746–4749, 2020.
- [33] M. D. Phung, T. H. Dinh, Q. P. Ha et al., "System architecture for real-time surface inspection using multiple UAVs," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2925–2936, 2019.
- [34] A. V. Savkin and H. Huang, "A method for optimized deployment of a network of surveillance aerial drones," *IEEE Systems Journal*, vol. 13, no. 4, pp. 4474–4477, 2019.
- [35] Y. Li, H. Chen, M. J. Er, and X. Wang, "Coverage path planning for UAVs based on enhanced exact cellular decomposition method," *Mechatronics*, vol. 21, no. 5, pp. 876–885, 2011.
- [36] L. Paull, C. Thibault, A. Nagaty, M. Seto, and H. Li, "Sensor-driven area coverage for an autonomous fixed-wing unmanned aerial vehicle," *IEEE Transactions on Cybernetics*, vol. 44, no. 9, pp. 1605–1618, 2014.
- [37] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, "A clustering-based coverage path planning method for autonomous heterogeneous UAVs," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [38] W. Bentz, T. Hoang, E. Bayasgalan, and D. Panagou, "Complete 3-D dynamic coverage in energy-constrained multi-UAV sensor networks," *Autonomous Robots*, vol. 42, no. 4, pp. 825–851, 2018.
- [39] T. M. Cabreira, P. R. Ferreira, C. Di Franco, and G. C. Buttazzo, "Grid-based coverage path planning with minimum energy over irregular-shaped areas with UAVs," in *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2019, pp. 758–767.
- [40] Y. Choi, Y. Choi, S. Briceño, and D. N. Mavris, "Energy-constrained multi-UAV coverage path planning for an aerial imagery mission using column generation," *Journal of Intelligent & Robotic Systems*, vol. 97, no. 1, pp. 125–139, 2020.
- [41] T. Elmokadem, "Distributed coverage control of quadrotor multi-UAV systems for precision agriculture," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 251–256, 2019.

- [42] M. Theile, H. Bayerlein, R. Nai, D. Gesbert, and M. Caccamo, "UAV coverage path planning under varying power constraints using deep reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1444–1449.
- [43] G. Sanna, S. Godio, and G. Guglieri, "Neural Network Based Algorithm for Multi-UAV Coverage Path Planning," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2021, pp. 1210–1217.
- [44] P. Maini, P. Tokekar, and P. Sujit, "Visual monitoring of points of interest on a 2.5 d terrain using a uav with limited field-of-view constraint," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 3661–3672, 2021.
- [45] P. Wang, R. Krishnamurti, and K. Gupta, "View planning problem with combined view and traveling cost," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 711–716.
- [46] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [47] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Computing Surveys (CSUR)*, vol. 35, no. 1, pp. 64–96, 2003.
- [48] R. Anand, D. Aggarwal, and V. Kumar, "A comparative analysis of optimization solvers," *Journal of Statistics and Management Systems*, vol. 20, no. 4, pp. 623–635, 2017.
- [49] C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 604–611, 2020.
- [50] G. Garcia and S. Keshmiri, "Nonlinear model predictive controller for navigation, guidance and control of a fixed-wing uav," in *AIAA guidance, navigation, and control conference*, 2011, p. 6310.
- [51] F. Gavilan, R. Vazquez, and E. F. Camacho, "An iterative model predictive control algorithm for uav guidance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 2406–2419, 2015.
- [52] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "A prototype of an autonomous controller for a quadrotor uav," in *2007 European Control Conference (ECC)*. IEEE, 2007, pp. 4001–4008.
- [53] Q. Wang, Y. Gao, J. Ji, C. Xu, and F. Gao, "Visibility-aware trajectory optimization with application to aerial tracking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5249–5256.
- [54] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3725–3732, 2018.
- [55] Z. Zhang and D. Scaramuzza, "Fisher information field: an efficient and differentiable map for perception-aware planning," *arXiv preprint arXiv:2008.03324*, 2020.
- [56] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 12, pp. 4338–4364, 2020.
- [57] R. A. Newcombe and A. J. Davison, "Live dense reconstruction with a single moving camera," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1498–1505.



Savvas Papaioannou received the B.S. degree in Electronic and Computer Engineering from the Technical University of Crete, Chania, Greece in 2011, the M.S. degree in Electrical Engineering from Yale University, New Haven, CT, USA, in 2013, and the Ph.D. degree in Computer Science from the University of Oxford, Oxford, U.K. in 2017. He is currently a Research Associate with the KIOS Research and Innovation Center of Excellence, University of Cyprus, Nicosia, Cyprus. His research interests include multi-agent and autonomous systems, state estimation and control, multi-target tracking, trajectory planning, and intelligent unmanned aerial vehicle (UAV) systems and applications. Dr. Papaioannou is a reviewer for various conferences and journals of the IEEE and ACM Associations, and he has served in the organizing committees of various international conferences.



Panayiotis Kolios is currently a Research Assistant Professor at the KIOS Research and Innovation Centre of Excellence of the University of Cyprus. He received his BEng and PhD degree in Telecommunications Engineering from King's College London in 2008 and 2011, respectively. Before joining the KIOS CoE, he worked at the Department of Communications and Internet Studies at the Cyprus University of Technology and the Department of Computer Science of the University of Cyprus (UCY). His work focuses on both basic and applied research on networked intelligent systems. Some examples of systems that fall into the latter category include intelligent transportation systems, autonomous unmanned aerial systems and the plethora of cyber-physical systems that arise within the Internet of Things. Particular emphasis is given to emergency management in which natural disasters, technological faults and man-made attacks could cause disruptions that need to be effectively handled. Tools used include graph theoretic approaches, algorithmic development, mathematical and dynamic programming, as well as combinatorial optimization.



Theocharis Theocharides is an Associate Professor in the Department of Electrical and Computer Engineering, at the University of Cyprus and a faculty member of the KIOS Research and Innovation Center of Excellence where he serves as the Research Director. Theocharis received his Ph.D. in Computer Engineering from Penn State University, working in the areas of low-power computer architectures and reliable system design, where he was honored with the Robert M. Owens Memorial Scholarship, in May 2005. He has been with the Electrical and

Computer Engineering department at the University of Cyprus since 2006, where he directs the Embedded and Application-Specific Systems-on-Chip Laboratory. His research focuses on the design, development, implementation, and deployment of low-power and reliable on-chip application-specific architectures, low-power VLSI design, real-time embedded systems design and exploration of energy-reliability trade-offs for Systems on Chip and Embedded Systems. His focus lies on acceleration of computer vision and artificial intelligence algorithms in hardware, geared towards edge computing, and in utilizing reconfigurable hardware towards self-aware, evolvable edge computing systems. His research has been funded by several National and European agencies and the industry, and he is currently involved in over ten funded ongoing research projects. He serves on several organizing and technical program committees of various conferences (currently serving as the Application Track Chair for the DATE Conference), is a Senior Member of the IEEE and a member of the ACM. He is currently an Associate Editor for the ACM Transactions on Emerging Technologies in Computer Systems, IEEE Consumer Electronics magazine, IET's Computers and Digital Techniques, the ETRI journal and Springer Nature Computer Science. He also serves on the Editorial Board of IEEE Design & Test magazine.



Christos Panayiotou is a Professor with the Electrical and Computer Engineering (ECE) Department at the University of Cyprus (UCY). He is also the Deputy Director of the KIOS Research and Innovation Center of Excellence for which he is also a founding member. Christos has received a B.Sc. and a Ph.D. degree in Electrical and Computer Engineering from the University of Massachusetts at Amherst, in 1994 and 1999 respectively. He also received an MBA from the Isenberg School of Management, at the aforementioned university in

1999. Before joining the University of Cyprus in 2002, he was a Research Associate at the Center for Information and System Engineering (CISE) and the Manufacturing Engineering Department at Boston University (1999 - 2002). His research interests include modeling, control, optimization and performance evaluation of discrete event and hybrid systems, intelligent transportation networks, cyber-physical systems, event detection and localization, fault diagnosis, wireless, ad hoc and sensor networks, smart camera networks, resource allocation, and intelligent buildings.

Christos has published more than 270 papers in international refereed journals and conferences and is the recipient of the 2014 Best Paper Award for the journal *Building and Environment* (Elsevier). He is an Associate Editor for the *IEEE Transactions on Intelligent Transportation Systems*, the *Conference Editorial Board of the IEEE Control Systems Society*, the *Journal of Discrete Event Dynamical Systems* and the *European Journal of Control*. During 2016-2020 he was Associate Editor of the *IEEE Transactions on Control Systems Technology*. He held several positions in organizing committees and technical program committees of numerous international conferences, including General Chair of the 23rd European Working Group on Transportation (EWGT2020), and General Co-Chair of the 2018 European Control Conference (ECC2018). He has also served as Chair of various subcommittees of the Education Committee of the IEEE Computational Intelligence Society.



Marios M. Polycarpou is a Professor of Electrical and Computer Engineering and the Director of the KIOS Research and Innovation Center of Excellence at the University of Cyprus. He is also a Member of the Cyprus Academy of Sciences, Letters, and Arts, and an Honorary Professor of Imperial College London. He received the B.A. degree in Computer Science and the B.Sc. in Electrical Engineering, both from Rice University, USA in 1987, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Southern California, in 1989

and 1992 respectively. His teaching and research interests are in intelligent systems and networks, adaptive and learning control systems, fault diagnosis, machine learning, and critical infrastructure systems. Dr. Polycarpou has published more than 350 articles in refereed journals, edited books and refereed conference proceedings, and co-authored 7 books. He is also the holder of 6 patents.

Prof. Polycarpou received the 2016 IEEE Neural Networks Pioneer Award. He is a Fellow of IEEE and IFAC and the recipient of the 2014 Best Paper Award for the journal *Building and Environment* (Elsevier). He served as the President of the IEEE Computational Intelligence Society (2012-2013), as the President of the European Control Association (2017-2019), and as the Editor-in-Chief of the *IEEE Transactions on Neural Networks and Learning Systems* (2004-2010). Prof. Polycarpou serves on the Editorial Boards of the *Proceedings of the IEEE*, the *Annual Reviews in Control*, and the *Foundations and Trends in Systems and Control*. His research work has been funded by several agencies and industry in Europe and the United States, including the prestigious European Research Council (ERC) Advanced Grant, the ERC Synergy Grant and the EU Teaming project.