

Rolling Horizon Coverage Control with Collaborative Autonomous Agents

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

Abstract

This work proposes a coverage controller that enables an aerial team of distributed autonomous agents to collaboratively generate non-myopic coverage plans over a rolling finite horizon, aiming to cover specific points on the surface area of a 3D object of interest. The collaborative coverage problem, formulated, as a distributed model predictive control problem, optimizes the agents' motion and camera control inputs, while considering inter-agent constraints aiming at reducing work redundancy. The proposed coverage controller integrates constraints based on light-path propagation techniques to predict the parts of the object's surface that are visible with regard to the agents' future anticipated states. This work also demonstrates how complex, non-linear visibility assessment constraints can be converted into logical expressions that are embedded as binary constraints into a mixed-integer optimization framework. The proposed approach has been demonstrated through simulations and practical applications for inspecting buildings with unmanned aerial vehicles (UAVs).

Index Terms

Intelligent agents, UAVs, Planning, Coverage, Model Predictive Control

I. INTRODUCTION

The interest in swarm systems such as systems utilizing multiple autonomous unmanned aerial vehicles (UAVs) has skyrocketed over the last few decades. Rapid advancements in robotics, automation and artificial intelligence coupled with the decreasing costs of electronic components have fuelled a remarkable surge in interest towards the technologies and applications of swarming systems. This work addresses the challenge of coverage planning and control using multiple collaborative intelligent autonomous agents, specifically autonomous UAVs. Coverage planning [1] is crucial in several application domains including search and rescue operations and critical infrastructure inspections. It is one of the essential functionalities that can notably enhance the autonomy of existing swarming systems enabling them to execute fully automated missions in the aforementioned scenarios. In coverage planning our objective is to design trajectories that allow a team of autonomous mobile agents to comprehensively cover a designated area or points of interest. Concurrently we aim to optimize a specific mission goal such as minimizing the mission's duration and energy consumption of the agents.

This work introduces a coverage control framework that optimizes both the kinematic and camera control inputs of multiple UAV agents simultaneously. This approach facilitates the creation of collaborative non-myopic coverage plans for 3D objects of interest. Specifically, the contributions of this work are the following: a) We propose a coverage planning framework designed to enable an aerial team of distributed agents to efficiently cover specific points on the surface area of an object of interest in 3D. Specifically, we develop a collaborative model predictive coverage controller that orchestrates the agents guiding them to optimally select their kinematic and camera control inputs. This process facilitates the generation of complementary non-myopic coverage plans that optimize coverage; b) we demonstrate that these collaborative coverage plans can be generated by incorporating light-path propagation constraints into the coverage controller. This integration aids in identifying parts of the scene that will be visible from the agents' projected future positions, a crucial aspect of developing effective non-myopic coverage strategies; and finally c) we demonstrate how the proposed collaborative coverage controller can be realized using mixed-integer programming techniques (MIP), by transforming complex, non-linear coverage constraints into logical expressions. We showcase its performance through both qualitative and quantitative real-world and synthetic experiments.

The rest of the paper is structured as follows: Section II reviews the relevant literature on coverage planning with single and multiple agents. Section III formulates the problem addressed in this study. Section IV details the proposed approach followed by an evaluation in Section V. Section VI discusses open problems and future directions in swarm systems and finally Section VII concludes this work.

II. RELATED WORK

With consumer drones becoming widespread and affordable coverage planning research has focused recently towards UAV-based applications. For instance, [2] uses a traveling-salesman approach for UAV-based coverage of 2D polygonal areas

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

whereas the authors in [3] use exact cellular decomposition to cover polygonal planar regions with a UAV agent equipped with a downward facing camera. Regarding the use of multiple UAVs for this problem, the work in [4] proposes an artificially weighted spanning tree algorithm for distributed coverage in planar regions involving multiple UAV agents. The authors in [5] propose a spatiotemporal clustering-based coverage method for multiple heterogeneous UAV agents also for planar environments. Other works such as [6] have focused on the aspect of energy efficiency in multi-UAV coverage planning. The approach in [7] uses mathematical programming techniques to design coverage paths that guide a team of UAV agents to cover a specified area of interest in the minimum amount of time. A UAV-centric cooperative coverage planning methodology is introduced in [8] utilizing the simulated annealing algorithm. This approach considers the sensing and operational capabilities of each UAV, their starting positions, and the designated no-fly zones. Additionally, [9] outlines a multi-UAV strategy for 2D terrain coverage focusing on reducing the overall completion time by ensuring a balanced workload distribution among the UAVs. Meanwhile, [10] proposes a cell decomposition algorithm for multi-UAV area coverage which employs regular hexagons to optimize the coverage process. The work in [11] investigates the multi-UAV coverage problem for 3D structures of interest and proposes a centralized sampling-based heuristic approach which combines the set-covering problem and the vehicle-routing problem. The resulting set-covering vehicle routing problem is then solved with a genetic algorithm. The work in [12] investigates the problem of 3D coverage using multiple UAV agents framing it as an offline path planning problem. The study proposes a heuristic approach based on potential fields, solved using the finite elements method.

Despite numerous coverage planning approaches proposed in the literature, a dominant solution for enabling autonomous multi-agent coverage planning in realistic 3D environments has yet to emerge. Current state-of-the-art methods primarily focus on 2D terrain coverage [13], neglecting the complexity of 3D objects. Moreover, they often presuppose that UAVs are equipped with fixed, uncontrollable sensors (e.g., downward-facing cameras) [14] disregarding the simultaneous optimization of the UAVs' kinematics and camera control inputs during the planning phase. This simplification notably reduces the complexity of coverage planning essentially transforming it into a path-planning problem [15]. Furthermore, many of the coverage planning techniques rely on simple geometric patterns (e.g., back-and-forth, zig-zag, and spiral motions) and utilize heuristics for area coverage [1], [12] which are not optimal, do not effectively generalize to 3D environments, or require centralized controllers [11], [16]. Finally, while distributed coverage planning methods have been explored to leverage the capabilities of multiple autonomous agents, these tend to yield myopic and greedy paths [17], [18] rather than collaborative, look-ahead coverage trajectories.

III. PROBLEM FORMULATION

A. Agent Dynamical Model

We consider a collaborative team of N autonomous networked aerial agents (i.e., UAVs) represented by $n \in \{1, \dots, N\}$ evolving within a finite 3D environment denoted as $\mathcal{E} \subset \mathbb{R}^3$. These agents, with states $\mathbf{x}_n = [\mathbf{x}_n^p, \mathbf{x}_n^v]^\top, \forall n$ are characterized by discrete-time dynamics composed of position ($\mathbf{x}_n^p \in \mathbb{R}^3$) and velocity ($\mathbf{x}_n^v \in \mathbb{R}^3$) components which are described by the discrete-time state-space model:

$$\mathbf{x}_n(t+1) = \mathbf{A}\mathbf{x}_n(t) + \mathbf{B}\mathbf{u}_n(t), \quad \forall t, n \in \{1, \dots, N\}, \quad (1)$$

where $\mathbf{x}_n(t)$ represents the state of the n_{th} agent at time step t , and $\mathbf{u}_n(t) \in \mathbb{R}^3$ signifies the control input. This input denotes the applied force to the n_{th} agent, enabling movement in a desired direction and at a certain speed. The state transition matrix \mathbf{A} , and the input matrix \mathbf{B} are given by: $\mathbf{A} = \begin{bmatrix} \mathbf{1}_{3 \times 3} & \Delta T \times \mathbf{1}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & (1 - \epsilon) \times \mathbf{1}_{3 \times 3} \end{bmatrix}$, and $\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \frac{\Delta T}{m} \times \mathbf{1}_{3 \times 3} \end{bmatrix}$ respectively, where ΔT is the sampling interval, ϵ is the air drag coefficient, and m denotes the agent's mass which without loss of generality in this work is assumed to be the same for all agents. Moreover, the constant matrices $\mathbf{1}_{3 \times 3}$ and $\mathbf{0}_{3 \times 3}$ denote the 3-by-3 identity and zero matrices respectively. In this work it is assumed that all agents maintain a wireless communication link with all their peers i.e., for exchanging information, and devising collaborative coverage plans. However, this assumption can be relaxed, and the role of communication is further discussed in Sec. V.

B. Agent Sensing Model

We consider that each agent n is outfitted with a rotating camera to monitor its environment as illustrated in Fig. 1(a). The camera's finite field of view (FOV) is represented as a regular right pyramid, featuring four triangular sides and a rectangular base. The optical center of the camera is aligned directly above the centroid of this rectangular base. The parameters defining the camera's FOV in our model are denoted by (c_l, c_w, c_r) , where c_l and c_w correspond to the length and width of the rectangular base respectively and c_r represents the pyramid's height indicating the range of the FOV. Subsequently, the camera's field of view (FOV) can be manipulated in three-dimensional space by instructing the camera controller to perform two sequential elemental rotations i.e., initially it rotates by an angle $\theta \in \Theta \subset [0, \pi]$ about the y -axis and then it undergoes a rotation by $\phi \in \Phi \subset [0, 2\pi]$ around the z -axis where Θ and Φ are finite sets denoting the admissible camera rotation angles. As a

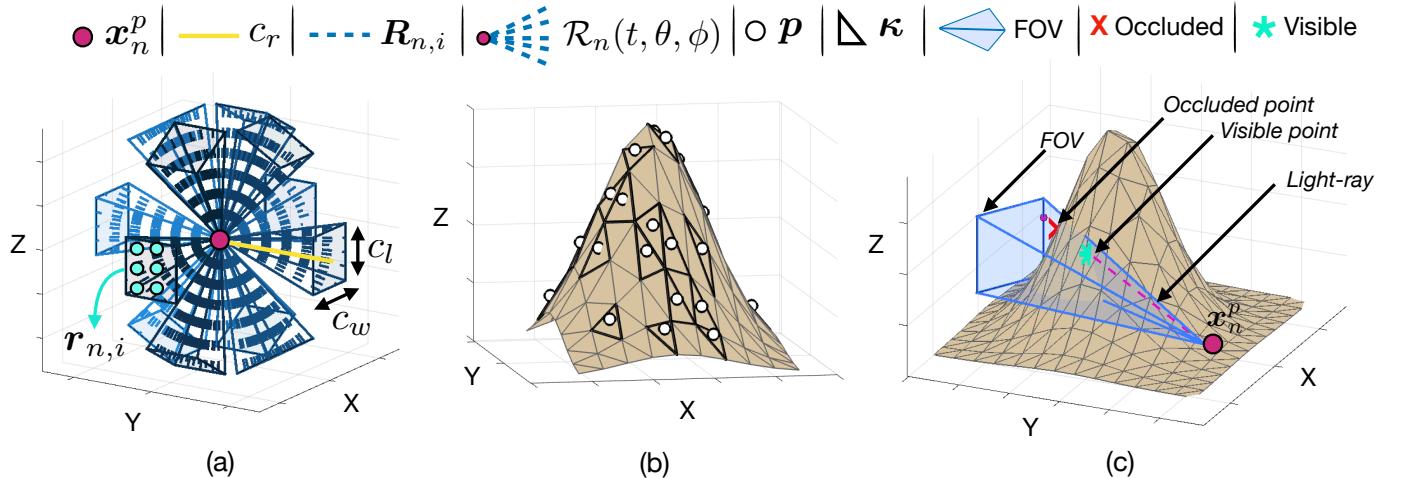


Fig. 1. The figure illustrates: (a) the agent sensing model, (b) the triangulated surface area of the object of interest, and the points p_i that need to be covered, (c) the need for incorporating light-path propagation to assess the coverage of points i.e., although both points (marked with *, and \times) reside inside the agent's FOV, only point * is visible as shown, this aspect is further discussed in Sec. IV-B.

result, at each time step t , the agent n with position $x_n^p(t)$ can rotate its camera's FOV through the subsequent geometric transformation:

$$\mathbf{M}_n(t, \theta, \phi)^i = \text{Rot}_z(\phi)\text{Rot}_y(\theta)\mathbf{M}_0^i + x_n^p(t), \forall i \in \{1, \dots, 5\}, \theta \in \Theta, \phi \in \Phi, \quad (2)$$

where $\mathbf{M}_0 \in \mathbb{R}^{3 \times 5}$ denotes the FOV vertices of a downward facing camera centered at the origin of the 3D cartesian coordinate system and $\text{Rot}_y(\theta)$, $\text{Rot}_z(\phi)$ denote the 3D rotation matrices around the y-axis and z-axis respectively. Here, \mathbf{M}_0^i represents the i th column of \mathbf{M}_0 . Consequently, $\mathbf{M}_n(t, \theta, \phi)^i$ denotes the vertex of the FOV that has been rotated and translated accordingly. The matrices \mathbf{M}_0 , $\text{Rot}_y(\theta)$ and $\text{Rot}_z(\phi)$ are respectively given by:

$$\begin{bmatrix} -\frac{c_l}{2} & \frac{c_l}{2} & \frac{c_l}{2} & -\frac{c_l}{2} & 0 \\ \frac{c_w}{2} & \frac{c_w}{2} & -\frac{c_w}{2} & -\frac{c_w}{2} & 0 \\ -c_r & -c_r & -c_r & -c_r & 0 \end{bmatrix}, \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \text{ and } \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Finally, we assume that at each time-step t a finite number N_R of (straight) light-rays modeling the direction of light propagation enters the camera's optical center and contribute to the imaging process. The set of light-rays captured by the agent's camera FOV $\mathbf{M}_n(t, \theta, \phi)$ is represented in this work by $\mathcal{R}_n(t, \theta, \phi) = \{\mathbf{R}_{n,1}, \dots, \mathbf{R}_{n,N_R}\}$ where $\mathbf{R}_{n,i}$ indicates an individual light-ray within the set. This light-ray is further characterized by the line segment:

$$\mathbf{R}_{n,i} = \mathbf{r}_{n,i} + h[\mathbf{x}_n^p(t) - \mathbf{r}_{n,i}], \quad \forall i \in \{1, \dots, N_R\}, h \in [0, 1], \quad (4)$$

where $\mathbf{x}_n^p(t)$ denotes the endpoint of the light-ray entering the camera's optical center at time-step t (given by the agent's positional state) and $\mathbf{r}_{n,i} \in \mathbb{R}^3$ is a fixed point on the camera's FOV base that serves as the ray's origin. It is important to note that depending on the agent's positional state and camera rotation angles θ and ϕ a different set $\mathcal{R}_n(t, \theta, \phi)$ of light-rays occur as $\mathcal{R}_n(t, \theta, \phi)$ is a function of $\mathbf{x}_n^p(t)$, θ and ϕ .

C. Collaborative Coverage Problem

The goal of the N agents is to collaboratively cover (i.e., observe) with their cameras a given set $\mathcal{P} = \{p_1, \dots, p_{|\mathcal{P}|}\}$, $p_{\tilde{p}} \in \mathbb{R}^3$ of points of interest (\tilde{p} denotes the index of point p) which reside on the surface area $\partial\mathcal{O}$ of an object of interest \mathcal{O} . Specifically, it is assumed that the object of interest \mathcal{O} has been 3D reconstructed as a point-cloud and its surface area $\partial\mathcal{O}$ has been triangulated into a 3D triangle mesh \mathcal{K} consisting of a finite set of non-overlapping triangular facets $\kappa \in \mathcal{K}$ where $\kappa \in \mathbb{R}^{3 \times 3}$ as shown in Fig. 1(b). Consequently, our aim becomes the generation of collision-free collaborative coverage trajectories which cover all points \mathcal{P} (where each point $p_{\tilde{p}} \in \mathcal{P}$ resides on the corresponding facet $\kappa_{\tilde{p}} \in \mathcal{K}$) on the object's surface area. Specifically, the problem tackled in this work can be stated as follows: *Given a team of N agents $n = \{1, \dots, N\}$, at each time-step t design a set of collision-free collaborative look-ahead coverage plans (i.e., determine each agent's kinematic and camera control inputs) inside a rolling finite planning horizon T which maximize the coverage of the points of interest \mathcal{P} on the object's surface area.*

IV. COLLABORATIVE ROLLING HORIZON 3D COVERAGE CONTROL

In order to tackle the coverage problem discussed above we formulate a distributed rolling horizon model predictive control (DMPC) problem, as detailed in Problem (P1). This controller seeks to find each agent's joint control inputs $\{\mathbf{u}_n(t+\tau|t), \theta_n(t+\tau|t)\}$ for $\tau \in [0, T]$ which maximizes the coverage of the points of interest \mathcal{P} on the object's surface area.

$\tau|t)$, $\phi_n(t + \tau|t)\}$, $n \in \{1, \dots, N\}$ over a rolling finite planning horizon $\tau \in \{1, \dots, T\}$ of length T time steps (i.e., predict the agents' states T time steps into the future), which optimize each agent's individual coverage objective function, denoted as \mathcal{J}_n and shown in Eq. (5a), subject to a set of kinematic, sensing, and coverage planning constraints that aim to generate complementary coverage trajectories as detailed in Eq. (5b) through Eq. (5m). In model predictive control, a longer planning horizon improves the optimality of results by considering a more extended future period but increases computational complexity, while a shorter horizon reduces computational demands at the expense of potentially suboptimal control actions.

Problem (P1) - Coverage Controller of Agent n :

$$\arg \min_{\{\mathbf{u}_n(t+\tau|t), \theta_n(t+\tau|t), \phi_n(t+\tau|t)\}} \mathcal{J}_n = \eta J_{\text{guidance}} + J_{\text{coverage}} \quad (5a)$$

subject to: $\tau \in \{1, \dots, T\}$

$$\mathbf{x}_n(t + \tau|t) = A\mathbf{x}_n(t + \tau - 1|t) + B\mathbf{u}_n(t + \tau|t) \quad \forall \tau \quad (5b)$$

$$\mathbf{x}_n(t|t) = \mathbf{x}_n(t|t - 1) \quad \forall \tau \quad (5c)$$

$$\mathbf{M}_{n,m}(t + \tau|t) = \tilde{\mathbf{M}}_{n,m} + \mathbf{x}_n^p(t + \tau|t) \quad \forall \tau, m \quad (5d)$$

$$\xi_{n,m,\tilde{p}}(t + \tau|t) = 1 \iff p \in \text{ConvHull}[\mathbf{M}_{n,m}(t + \tau|t)] \quad \forall \tau, m, \tilde{p} \quad (5e)$$

$$\sum_{m=1}^{|\{\Theta \times \Phi\}|} \mu_{n,m}(t + \tau|t) = 1 \quad \forall \tau, \quad (5f)$$

$$\bar{\xi}_{n,m,\tilde{p}}(t + \tau|t) = \xi_{n,m,\tilde{p}}(t + \tau|t) \wedge \mu_{n,m}(t + \tau|t) \quad \forall \tau, m, \tilde{p} \quad (5g)$$

$$\hat{\xi}_{n,m,\tilde{p}}(t + \tau|t) \leq \bar{\xi}_{n,m,\tilde{p}}(t + \tau|t) + \mathcal{Q}_n(\tilde{p}) + \sum_{k \neq n=1}^N \mathcal{Q}_k(\tilde{p}) \quad \forall \tau, m, \tilde{p} \quad (5h)$$

$$\dot{\xi}_{n,m,\tilde{p}}(t + \tau|t) \leq \hat{\xi}_{n,m,\tilde{p}}(t + \tau|t) + \sum_{k < n} \sum_m \sum_\tau \dot{\xi}_{k,m,\tilde{p}}(t + \tau|t) + \sum_{k > n} \sum_m \sum_\tau \dot{\xi}_{k,m,\tilde{p}}(t - 1 + \tau|t - 1) \quad \forall \tilde{p} \quad (5i)$$

$$\mathbf{x}_n^p(t + \tau|t) \notin \text{ConvHull}[\mathcal{O}] \quad \forall \tau \quad (5j)$$

$$\mathbf{x}_n(t + \tau|t) \in \mathcal{X}, \mathbf{u}_n(t + \tau|t) \in \mathcal{U} \quad (5k)$$

$$\xi_{n,m,\tilde{p}}(t + \tau|t), \bar{\xi}_{n,m,\tilde{p}}(t + \tau|t), \hat{\xi}_{n,m,\tilde{p}}(t + \tau|t), \dot{\xi}_{n,m,\tilde{p}}(t + \tau|t) \in \{0, 1\} \quad (5l)$$

$$\mu_{n,m}(t + \tau|t), \mathcal{Q}_n(\tilde{p}) \in \{0, 1\}, \tilde{p} \in \{1, \dots, |\mathcal{P}|\}, m \in \{1, \dots, |\{\Theta \times \Phi\}|\} \quad (5m)$$

In Problem (P1), at each time-step t agent n plans collaborative finite-length look-ahead coverage trajectories $\mathbf{x}_n(t + \tau|t)$, $\tau \in \{1, \dots, T\}$ which aim at optimizing the coverage of the points of interest \mathcal{P} inside the planning horizon. The notation $\mathbf{x}_n(t'|t)$ denotes the predicted agent state at time-step $t' \geq t$ which was computed at time-step t . Therefore, the coverage planning problem is thus solved iteratively over multiple time-steps t in a rolling horizon fashion where the first set of predicted control inputs in the sequence $\{\mathbf{u}_n(t + 1|t), \theta_n(t + 1|t), \phi_n(t + 1|t)\}$, $n \in \{1, \dots, N\}$ is executed in the next time-step the agents move to their new states and the optimization process shown above repeats for the next time-step over a shifted planning horizon until all points \mathcal{P} are covered.

Instead of addressing a large centralized optimization problem where all required information is sent to a central station that subsequently determines the control inputs for each agent as suggested in [16] we break down the multi-agent coverage problem into smaller sub-problems that each agent can solve autonomously. Problem (P1) ensures that the control actions undertaken by one agent are consistent with those of all other agents in the team taking into account any interlinked coverage constraints between agents during the decision-making process. This is achieved through a coordination procedure [19] where agent n acquires the most recent plans from all preceding agents $i < n$ in the sequence and also gathers the projected plans from subsequent agents $i > n$ who have yet to finalize their latest plans.

In this approach the computational complexity of the proposed distributed coverage controller is decoupled from the number of collaborative agents, as opposed to a centralized formulation which becomes computationally intractable as the number of agents increases. In addition, we should point out that although this approach necessitates constant communication among the agents it enables the creation of complementary predictive look-ahead coverage plans that minimize the duplication of work. It is important to note that in certain application scenarios such as monitoring critical infrastructure structural inspection and area coverage for emergency response constant communication among the team of agents is crucial not only for operational efficiency but also for safety and security purposes. For example, in search-and-rescue operations once a victim is located, the discovering agent must quickly relay this information to the rest of the team which can then coordinate to provide assistance. Nevertheless, it is possible to relax the assumption of constant communication. In such cases, agents can opportunistically exchange plans

whenever they are within communication range or apply the proposed coordination scheme within smaller neighborhoods. However, this flexibility comes at the cost of generating coverage plans that may exhibit overlaps and some duplication of work. In Sec. IV-A we discuss in detail the proposed collaborative controller shown in Problem (P1). Subsequently, in Sec. IV-B we show how we extend this approach to include light-path propagation constraints to determine the visibility of the points of interest with respect to the agents' future predicted states, and we demonstrated how non-linear and non-convex visibility assessment constraints can be converted into logical expression which can easily be embedded into a mixed integer optimization framework.

A. Distributed Coverage Planning

Problem (P1) is executed by each agent n in a coordinated manner to generate joint optimized coverage plans over a rolling planning horizon $\tau \in \{1, \dots, T\}$ with the goal of covering points of interest in \mathcal{P} . As demonstrated, this objective is accomplished through a rolling horizon mixed integer program (MIP), the specifics of which are elaborated next.

1) *Collaborative Coverage Constraints:* The constraints in Eq. (5b)-(5c) are due to the agent dynamics. These constraints are used to compute the predicted trajectory of agent n inside the planning horizon by appropriately selecting the agent's direction and speed through the control input $\mathbf{u}_n(t + \tau|t), \forall \tau$. Subsequently, the constraint in Eq. (5d) computes, all possible realisations of agent's n camera states $\mathbf{M}_{n,m}(t + \tau|t)$ inside the planning horizon with respect to its predicted trajectory. Because the camera rotations involve non-linear transformations, as shown in Eq. (2), which make optimization challenging in this work we precompute all possible camera rotations at the origin. These are subsequently translated during optimization to the desired location based on the predicted trajectory. In other words, in order to embed camera rotation constraints into a linear MIP we precompute all possible camera states $\tilde{\mathbf{M}}_{n,m}$ at the origin as: $\tilde{\mathbf{M}}_{n,m} = \text{Rot}_z(\phi)\text{Rot}_y(\theta)\mathbf{M}_0, \forall m \in \{1, \dots, |\{\Theta \times \Phi\}|\}$, where $|\{\Theta \times \Phi\}|$ is the cardinality of the set of angles which results from the Cartesian product between the finite sets Θ and Φ . Subsequently, Eq. (5d) translates for each agent all these possible camera states to its predicted location $\mathbf{x}_n^p(t + \tau|t)$. Therefore, $\mathbf{M}_{n,m}(t + \tau|t)$ denotes the m th camera pose of agent n at time-step $t + \tau|t$.

Next, for each point of interest $\mathbf{p}_{\tilde{p}} \in \mathcal{P}$ we associate a binary variable $\xi_{n,m,\tilde{p}}(t + \tau|t) \in \{0, 1\}$ which indicates whether point $\mathbf{p}_{\tilde{p}}$ is covered by the m th camera state of agent n at the future time-step $t + \tau|t$. This binary variable is activated (i.e., becomes equal to 1) when point $\mathbf{p}_{\tilde{p}}$ resides inside the convex hull of the m th camera FOV $\mathbf{M}_{n,m}(t + \tau|t)$ at time-step $t + \tau|t$ as shown in Eq. (5e). A point $\mathbf{p}_{\tilde{p}} \in \mathcal{P}$ that belongs to the convex-hull defined by the camera FOV vertices in $\mathbf{M}_{n,m}(t + \tau|t)$ must satisfy the following system of linear inequalities: $\text{dot}(\boldsymbol{\alpha}_{i,n,m}(t + \tau|t), \mathbf{p}_{\tilde{p}}) \leq \beta_{i,n,m}(t + \tau|t), \forall i \in \{1, \dots, 5\}$, where $\text{dot}(\mathbf{a}, \mathbf{b})$ is the dot product between vectors \mathbf{a} and \mathbf{b} , $\text{dot}(\boldsymbol{\alpha}_{i,n,m}(t + \tau|t), \mathbf{p}_{\tilde{p}}) = \beta_{i,n,m}(t + \tau|t)$ is the equation of the plane which at time-step $t + \tau|t$ contains the i th face of the m th camera FOV state, of the n th agent $\boldsymbol{\alpha}_{i,n,m}(t + \tau|t)$ is the unit outward normal vector to this plane and $\beta_{i,n,m}(t + \tau|t)$ is a constant. Any point $\mathbf{p}_{\tilde{p}} \in \mathcal{E}$ which satisfies the aforementioned system of inequalities is contained within the convex-hull of $\mathbf{M}_{n,m}(t + \tau|t)$ and therefore can be potentially observed by the agent (provided it is visible). This functionality is then implemented as follows:

$$\text{dot}(\boldsymbol{\alpha}_{i,n,m}(t + \tau|t), \mathbf{p}_{\tilde{p}}) + \tilde{\xi}_{i,n,m,\tilde{p}}(t + \tau|t) (M - \beta_{i,n,m}(t + \tau|t)) \leq M, \quad \forall i, m, \tau, \tilde{p} \quad (6a)$$

$$5\xi_{n,m,\tilde{p}}(t + \tau|t) - \sum_{i=1}^5 \tilde{\xi}_{i,n,m,\tilde{p}}(t + \tau|t) \leq 0, \quad \forall m, \tau, \tilde{p}. \quad (6b)$$

In Eq. (6a) observe that the auxiliary binary variable $\tilde{\xi}_{i,n,m,\tilde{p}}(t + \tau|t) \in \{0, 1\}$ becomes equal to 0 when the i th inequality is not satisfied (i.e., $\text{dot}(\boldsymbol{\alpha}_{i,n,m}(t + \tau|t), \mathbf{p}_{\tilde{p}}) \not\leq \beta_{i,n,m}(t + \tau|t)$) and the constraints in Eq. (6a) holds with the use of a big positive constant M . Conversely, $\tilde{\xi}_{i,n,m,\tilde{p}}(t + \tau|t) = 1$ when the i th inequality is satisfied. As a result the activation of $\xi_{n,m,\tilde{p}}(t + \tau|t) \in \{0, 1\}$ for a particular configuration of the parameters (m, \tilde{p}, τ) indicates that agent n covers with the m th camera state point $\mathbf{p}_{\tilde{p}}$ at time-step $t + \tau|t$ which is achieved with the constraint shown in Eq. (6b).

Next, the binary variable $\mu_{n,m}(t + \tau|t) \in \{0, 1\}$ indicates which of the $|\Theta \times \Phi|$ camera states is active at time-step $t + \tau|t$. Essentially, $\boldsymbol{\mu}_n(t + \tau|t)$ is a matrix with $|\Theta \times \Phi|$ rows (indexed by m) and T columns. At each time-step $t + \tau|t$ only one camera state should be active (i.e., the sum of each column should be equal to one) which is enforced via the constraint in Eq. (5f). The subsequent logical conjunction, i.e., Eq. (5g) makes sure that only points that have been covered with the active camera state are considered as indicated by the binary variable $\bar{\xi}_{n,m,\tilde{p}}(t + \tau|t) \in \{0, 1\}$. Now, in order to handle the duplication of work (i.e., ensure that points of interest previously covered are not scheduled for future coverage) we need to keep track all points that have been covered. In a centralized formulation where the agents consult a central database $\mathcal{Q}(\tilde{p}) \in \{0, 1\}, \forall \tilde{p} \in \{1, \dots, |\mathcal{P}|\}$ which records the coverage status of each point $\mathbf{p}_{\tilde{p}} \in \mathcal{P}$ this can be accomplished with the following constraint: $\hat{\xi}_{n,m,\tilde{p}}(t + \tau|t) \leq \bar{\xi}_{n,m,\tilde{p}}(t + \tau|t) + \mathcal{Q}(\tilde{p}), \forall n, m, \tilde{p}$. For a point indexed by \tilde{p} which has been already covered this constraint makes sure that the agents are discouraged from generating plans that include for coverage point $\mathbf{p}_{\tilde{p}}$ since the value of $\hat{\xi}_{n,m,\tilde{p}}(t + \tau|t)$ is activated through $\mathcal{Q}(\tilde{p})$ prompting the agents to focus on points yet to be covered. We will show in Sec. IV-A2 how this binary variable can be linked to the coverage objective.

However, in the absence of a central station the database $\mathcal{Q}(\tilde{p})$ is held in a distributed form amongst the agents i.e., $\mathcal{Q}(\tilde{p}) = \sum_{n=1}^N \mathcal{Q}_n(\tilde{p}), \forall \tilde{p}$ and therefore this functionality is implemented by allowing the agents to exchange at each time-step t their individual records \mathcal{Q}_n as shown in Eq. (5h).

To promote the generation of collaborative and complementary coverage plans that minimize work duplication the following constraint must be implemented: $\sum_n \sum_m \sum_{\tau} \dot{\xi}_{n,m,\tilde{p}}(t + \tau|t) \leq 1, \forall \tilde{p}$. This constraint ensures that during the planning horizon each point $p_{\tilde{p}}$ is scheduled for coverage exactly once and by no more than one agent. In order to implement this constraint in a distributed fashion the agents generate their plans in a coordinated fashion and communicate their predicted plans to their peers. Specifically, agent n receives the latest plans denoted as $\dot{\xi}_{k,m,\tilde{p}}(t + \tau|t), \forall k < n$ from all agents earlier in the sequence (i.e., for $k < n$) and the previous predicted plans from all agents (i.e., $k > n$) later in the sequence i.e., $\dot{\xi}_{k,m,\tilde{p}}(t - 1 + \tau|t - 1), \forall k > n$. Consequently, the generation of complimentary coverage plans is achieved with the constraint shown in Eq. (5i). The communication and information exchange protocol is further discussed in Sec. IVIV-A-IV-A3.

Finally, the constraint in Eq. (5j) ensures that agent n avoids collisions with the object of interest (and various obstacles in the environment). The convex-hull of the object of interest \mathcal{O} , is given by the intersection of $|\mathcal{K}|$ half-spaces (where \mathcal{K} is the triangle mesh representing the surface area of \mathcal{O}). Suppose that the i th half-space is associated with the plane equation $\text{dot}(\alpha_i, \mathbf{x}) = \beta_i, i \in \{1, \dots, |\mathcal{K}|\}, \mathbf{x} \in \mathcal{E}$ which divides the 3D space into two parts. A collision with the object of interest can be avoided with the following constraints:

$$\text{dot}(\alpha_i, \mathbf{x}_n^p(t + \tau|t)) + Mo_{n,i}(t + \tau|t) > \beta_i, \forall \tau, i, \quad (7)$$

$$\sum_{i=1}^{|\mathcal{K}|} o_{n,i}(t + \tau|t) \leq (|\mathcal{K}| - 1), \forall \tau, \quad (8)$$

where M is a big positive constant, and $o_{n,i}(t + \tau|t) \in \{0, 1\}$ is a binary variable which on activation (i.e., Eq. (7)) indicates that $\text{dot}(\alpha_i, \mathbf{x}_n^p(t + \tau|t)) \leq \beta_i$ holds for time-step $t + \tau|t$. Because agent n is considered to collide with the object \mathcal{O} at time-step $t + \tau|t$ (i.e., resides within the object's convex hull) when $\text{dot}(\alpha_i, \mathbf{x}_n^p(t + \tau|t)) \leq \beta_i$ for all i the constraint in Eq. (8) ensures that the number of times $o_{n,i}(t + \tau|t)$ is activated at each time-step during the planning horizon is less than or equal to $|\mathcal{K}| - 1$. This condition is set to avoid collision. The constraints in Eq. (7) - (8) can be extended for multiple obstacles in the environment. The same principle can be applied for collision avoidance among the agents by making sure that the positional state of agent n must remain outside the convex hull defined by a certain safety area $\mathcal{A}_k(t + \tau|t)$ of any other agent k (where $k \neq n$ and $k \in \{1, \dots, N\}$). For instance, $\mathcal{A}_k(t + \tau|t)$ could represent a convex area around $\mathbf{x}_k^p(t + \tau|t)$ serving as an approximation of a spherical safety zone with a specific radius around the agent as described in our previous work [20]. Finally, the coverage mission concludes once every point p has been covered a condition met when $\exists n \in \{1, \dots, N\} : \sum_{\tilde{p}} \mathcal{Q}_n(\tilde{p}) = |\mathcal{P}|$.

2) *Agent Guidance and Coverage Objective:* Given the problem constraints discussed above and shown in Eq. (5b) - (5m) each agent n can generate non-myopic coverage plans (i.e., a sequence of camera states $\mathbf{M}_{n,m}(t + \tau|t), m \in \{1, \dots, |\{\Theta \times \Phi\}|\}$ along its predicted trajectory $\mathbf{x}_n(t + \tau|t)$) by minimizing the objective function $J_n = \eta J_{\text{guidance}} + J_{\text{coverage}}$ which is composed of a guidance term as well as a coverage term defined respectively by $J_{\text{guidance}} = \|\mathbf{x}_n(t + 1|t) - \mathbf{p}_n^*\|_2^2$ and $J_{\text{coverage}} = \sum_{\tau=1}^{T-1} \sum_{m=1}^{|\{\Theta \times \Phi\}|} \sum_{\tilde{p}=1}^{|\mathcal{P}|} \dot{\xi}_{n,m,\tilde{p}}(t + \tau + 1|t)$. Therefore, the following objective function is minimized:

$$J_n = \eta \|\mathbf{x}_n^p(t + 1|t) - \mathbf{p}_n^*\|_2^2 - \sum_{\tau=1}^{T-1} \sum_{m=1}^{|\{\Theta \times \Phi\}|} \sum_{\tilde{p}=1}^{|\mathcal{P}|} \dot{\xi}_{n,m,\tilde{p}}(t + \tau + 1|t). \quad (9)$$

where η is a tuning weight. The first term (i.e., J_{guidance}) is a sub-objective responsible for guiding agent n to its nearest unobserved point \mathbf{p}_n^* . This ensures mission progress (avoiding deadlocks) especially in scenarios where no points of interest are reachable within the finite planning horizon. To determine this point at each time-step t agent n receives from all its peers their current positional states $\mathbf{x}_k^p(t|t), \forall k \neq n$, and computes \mathbf{p}_n^* by solving the following assignment problem:

$$\hat{\mathbf{H}} = \min_{\mathbf{H}} \sum_{k=1}^N \sum_{\tilde{p}=1}^{|\tilde{\mathcal{P}}|} \mathbf{C}(k, \tilde{p}) \mathbf{H}(k, \tilde{p}) \quad (10)$$

where $k \in \{1, \dots, N\}$ is the agent index, $|\tilde{\mathcal{P}}|$ represents the total number of unvisited points of interest obtained from the record \mathcal{Q} , \mathbf{C} is a cost matrix containing the pairwise distances between the agents' positional states and the unobserved points of interest, and $\mathbf{H}(k, \tilde{p}) \in \{0, 1\}$ is an assignment matrix with the property that each column and each row sum to one. Therefore, all agents $n \in \{1, \dots, N\}$ solve at each time-step t the same assignment problem shown in Eq. (10) to jointly find their closest unobserved points of interest to target next, i.e., by determining the optimal assignment matrix $\hat{\mathbf{H}}$ and identifying \mathbf{p}_n^* from the n th row of $\hat{\mathbf{H}}$.

The second term (J_{coverage}) maximizes the number of points predicted for coverage within the planning horizon. The binary variable $\dot{\xi}_{n,m,\tilde{p}}(t + \tau|t)$ signals whether agent n 's predicted trajectory covers a point of interest (\tilde{p}) at a future time-step $(t + \tau|t)$ with camera state m . Minimizing the negative of this variable optimizes the agent's movement and camera controls to increase coverage.

3) *Information Exchange:* To summarize, at each time-step t each agent $n \in \{1, \dots, N\}$ solves the distributed Model Predictive Control (MPC) problem outlined in Problem (P1) over a rolling finite planning horizon of length T time-steps. The agents optimize their local objective function as described in Sec. IVIV-A-IV-A2 with respect to their own control inputs and decision (binary) variables as detailed in equations (5b) to (5m) to generate complementary look-ahead coverage plans through coordination and information exchange. This process is facilitated by the following coordinated processing and information exchange protocol: At each time-step t agent n acquires the most recent plans from all preceding agents $k < n$ in the sequence $\xi_{k,m,\tilde{p}}(t + \tau|t), \tau \in \{1, \dots, T\}$ and also collects the previous predicted plans of all subsequent agents $k > n$ who have yet to generate their latest plans, $\xi_{k,m,\tilde{p}}(t - 1 + \tau|t - 1), \tau \in \{1, \dots, T\}$. Because these binary decision variables indicate the points of interest $\tilde{p} \in \{1, \dots, |\mathcal{P}|\}$ that are scheduled for coverage by other agents $k \neq n$ agent n uses this information to generate plans that cover points not scheduled for observation by other agents. This is shown in Eq. (5i) which essentially decentivizes agent n for generating a coverage trajectory for a point of interest which has been scheduled for coverage by preceding agents in the sequence while also accounting for points of interest that have been scheduled for coverage in the previous time-step by agents which have yet to generate their latest plans. Subsequently, the agents that generate plans after agent n will take into account the latest plans of all previous agents $k \in \{1, \dots, n\}$ and in turn they will revise their previous plans accordingly targeting points of interest not already scheduled for coverage.

Additionally, at each time-step t agent n receives from all other agents their record of already visited points, $\mathcal{Q}_k, \forall k \neq n$ which it combines to generate plans for unobserved points as captured by the binary variable $\hat{\xi}_n$ used in Eq. (5i). Finally, agent n also receives the positional state of all other agents $k \neq n$ at each time-step t which is used for computing p_n^* in the guidance objective as discussed in Sec. IV-A2. This information exchange protocol is repeated at each time-step enabling the agents to adapt their decisions on-line while considering the plans of their peers and minimizing duplication of work by generating complementary plans.

B. Integrating Light-path Propagation Constraints

Although Problem (P1), generates non-myopic coverage plans which maximize the number of points of interest which reside inside the convex-hull of the agent's camera FOV inside the planning horizon it does not account for the notion of visibility as illustrated in Fig. 1(c). In its current form Problem (P1) does not provide a way of determining which parts of the object's surface area are visible given the future planned state of the agent.

In the context of visibility determination consider the point $p_{\tilde{p}}$ situated on a specific facet $\kappa_{\tilde{p}}$ within the set \mathcal{K} . For $p_{\tilde{p}}$ to be deemed visible at a future time-step $t + \tau|t$ through the field of view (FOV) of the camera $M_{n,m}(t + \tau|t)$ it must reside within the convex hull of this FOV and subsequently visibility is affirmed if there is at least one light-ray $R_{n,i}$ from the set $\mathcal{R}_{n,m}(t + \tau|t)$ for $i \in \{1, \dots, N_R\}$ that intersects lastly with $\kappa_{\tilde{p}}$ which contains $p_{\tilde{p}}$ before hitting the camera lens center. This set $\mathcal{R}_{n,m}(t + \tau|t)$ encapsulates all potential light-rays dictated by the camera's state m at the given future time. To confirm visibility it must be ensured that the intersection operation symbolized by \oplus between any light-ray $R_{n,i}$ and the facets in \mathcal{K} results in the facet $\kappa_{\tilde{p}}$ which holds $p_{\tilde{p}}$. If this intersection operation returns $\kappa_{\tilde{p}}$ then $p_{\tilde{p}}$ is considered visible from the camera state $M_{n,m}(t + \tau|t)$. Conversely, if no intersections occur the operation results in \emptyset .

Define the equation of the plane containing the facet κ as $\text{dot}(\alpha_\kappa, \mathbf{x}) = \beta_\kappa$ where $\alpha_\kappa \in \mathbb{R}^3$ represents the unit normal vector perpendicular to the plane of κ and $\mathbf{x} \in \mathbb{R}^3$ denotes a point in space. The intersection point of the light-ray $R_{n,i} = \mathbf{r}_{n,i} + h[\mathbf{x}_n^p(t) - \mathbf{r}_{n,i}]$, where $h \in [0, 1]$, with the plane of facet κ is determined by the following set of equations:

$$\text{dot}(\alpha_\kappa, \mathbf{r}_{n,i} + h[\mathbf{x}_n^p(t) - \mathbf{r}_{n,i}]) = \beta_\kappa \implies \quad (11a)$$

$$h = \frac{\beta_\kappa - \text{dot}(\alpha_\kappa, \mathbf{r}_{n,i})}{\text{dot}(\alpha_\kappa, \mathbf{x}_n^p(t) - \mathbf{r}_{n,i})}, \quad (11b)$$

where Eq. (11a) results from inserting the expression for $R_{n,i}$ into the plane's equation and Eq. (11b) solves for h . If the denominator in Eq. (11b) equals zero it implies the light-ray is parallel to the facet leading to no intersection or an undefined one. Thus, for visibility the condition $\text{dot}(\alpha_\kappa, \mathbf{x}_n^p(t) - \mathbf{r}_{n,i}) \neq 0$ is required to ensure a unique intersection. The intersection is valid if $h \in [0, 1]$ and the intersection point $\hat{\mathbf{x}} = \mathbf{r}_{n,i} + \hat{h}[\mathbf{x}_n^p(t) - \mathbf{r}_{n,i}]$ (with \hat{h} as the solution from Eq. (11b)) lies within the convex hull of κ that is, $\hat{\mathbf{x}} \in \text{ConvHull}(\kappa)$.

The process described above must be applied at every time-step within the planning horizon to evaluate all potential realizations of light-rays $R_{n,i} \in \mathcal{R}_{n,m}(t + \tau|t)$ determined by the agent's camera states $M_{n,m}(t + \tau|t)$ and the triangular mesh \mathcal{K} describing the object of interest. This assessment is crucial for determining the visibility of each point of interest $p_{\tilde{p}} \in \mathcal{P}$ in relation to the agent's projected path. However, this method is computationally intensive and in addition requires the integration of non-convex and non-linear constraints (as illustrated in Eq. (11b)) which complicate efficient optimization. To circumvent these difficulties this work adopts an alternative strategy that involves initially learning state-dependent light-path propagation constraints to assess visibility. These constraints are then integrated as logical constraints into the coverage controller outlined in Problem (P1).

This is achieved by first partitioning the environment \mathcal{E} into a 3D grid \mathcal{G} composed of discrete, non-overlapping cells denoted as $\mathcal{G} = \{\mathbf{G}_1, \dots, \mathbf{G}_{|\mathcal{G}|}\}$, with the union of all cells covering the entire grid $\bigcup_{g=1}^{|\mathcal{G}|} \mathbf{G}_g = \mathcal{G}$. Within each cell $\mathbf{G} \in \mathcal{G}$, N_s joint

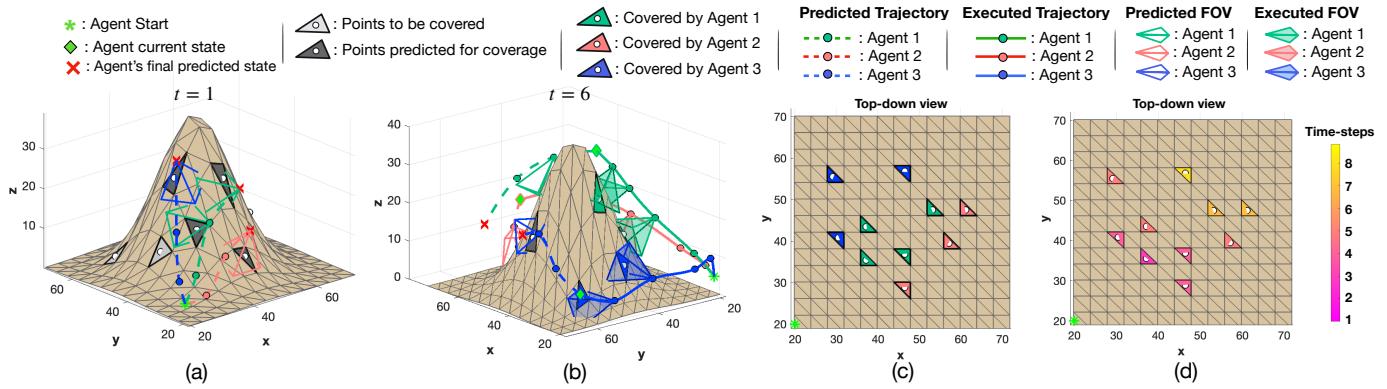


Fig. 2. An illustrative example of the proposed collaborative 3D coverage controller, involving three agents indicated by green, red, and blue colors. (a) and (b) Predicted plans at time steps 1 and 6, respectively. (c) Allocation of points to the agents, and (d) the time step at which each point was covered.

configurations of the agent's position and camera state are randomly sampled, $(\mathbf{x}_{n,i}^p \in G, M_{n,m,i}, m \in \{1, \dots, |\{\Theta \times \Phi\}|\})$ for each $i \in \{1, \dots, N_s\}$. The visibility assessment outlined in Eq. (11b) is then executed for these configurations to determine point visibility within each cell. In essence for each cell $g \in \{1, \dots, |\mathcal{G}|\}$ we obtain a set of light-rays denoted as $\tilde{\mathcal{R}}_g = \bigcup_{i=1}^{N_s} \mathcal{R}_i$, where \mathcal{R}_i corresponds to the set of light-rays obtained from the i th sample of the agent's positional and camera state in cell G_g . Subsequently, we learn off-line the following state-dependant visibility assessment constraints for each point of interest and each cell:

$$v_{g,\tilde{p}} = 1 \iff \exists \mathbf{R} \in \tilde{\mathcal{R}}_g : \mathbf{R} \oplus \mathcal{K} = \kappa_{\tilde{p}}, \forall g \in \{1, \dots, |\mathcal{G}|\}, \tilde{p} \in \{1, \dots, |\mathcal{P}|\} \quad (12)$$

Once the constraints above are learned we can determine the visibility of point $p_{\tilde{p}}$ by utilizing the binary variable $v_{g,\tilde{p}}$ which is activated when there exists a light-ray \mathbf{R} which traces back to facet $\kappa_{\tilde{p}}$ when the agent resides within the cell G_g . Subsequently, we integrate these visibility assessment constraints to the proposed coverage controller in Problem (P1) by extending Eq. (5g) as follows:

$$\bar{\xi}_{n,m,\tilde{p}}(t + \tau|t) = \xi_{n,m,\tilde{p}}(t + \tau|t) \wedge \mu_{n,m}(t + \tau|t) \wedge [v_{g,\tilde{p}} \wedge \varpi_{n,g}(t + \tau|t)], \forall m, \tilde{p}, g, \tau, \quad (13)$$

where the last part in the square brackets is due to the visibility assessment. Specifically, the binary variable $\varpi_{n,g}(t + \tau|t) = 1 \iff \mathbf{x}_n^p(t + \tau|t) \in \text{ConvHull}[\mathcal{G}_g]$ is activated whenever agent n resides within the convex-hull of cell \mathcal{G}_g , and subsequently this result is integrated with the visibility assessment binary variable $v_{g,\tilde{p}}$ which indicates whether the point $p_{\tilde{p}}$ is visible from cell \mathcal{G}_g , further combined with the active FOV (i.e., $\mu_{n,m}(t + \tau|t)$), and the binary variable that indicates the coverage of point $p_{\tilde{p}}$, i.e., $\xi_{n,m,\tilde{p}}(t + \tau|t)$, as shown in Eq. (13).

V. EVALUATION

A. Experimental Setup

To evaluate the proposed approach we assume agents with identical capabilities are operating within a bounded 3D environment \mathcal{E} which is confined in each dimension to the interval $[0, 100]\text{m}$. The UAV agents are modelled based on the performance characteristics and specifications of the DJI Mavic Enterprise UAV platform. Subsequently, the agent's n dynamics are characterised by $\Delta T = 1\text{s}$, $\epsilon = 0.3$, and $m = 1.75\text{kg}$. The agent velocity is bounded within the interval $[-17, 17]\text{m/s}$, whereas the control input \mathbf{u}_n is bounded within the interval $[-10, 10]\text{N}$. The tuning weight η in Eq. (9) is set to 0.8. The agent camera FOV model parameters (c_l, c_w, c_r) are set to $(8.5, 8.5, 10)\text{m}$, and the gimbal rotation angles θ and ϕ take their values from the finite sets $\Theta = \{30, 70, 110, 150\}\text{deg}$, and $\Phi = \{30, 90, 150, 210, 270, 330\}\text{deg}$ respectively. To determine visibility we have used $N_R = 50$ light rays, N_s was set to 100, and the environment was decomposed into $|\mathcal{G}| = 100$ cells. Finally Problem (P1) was solved with the Gurobi solver running on a 3.2GHz desktop computer.

B. Results

An illustrative example of the proposed approach featuring 3 UAV agents is presented in Fig. 2 for the coverage of 10 points on the surface area of the object depicted in the figure. Specifically, Fig. 2(a) presents the predicted coverage plans (i.e., dotted lines) for the first time-step for 3 agents operating within a planning horizon of length $T = 3$ time-steps all starting at the position $(x, y, z) = [20, 20, 5]$. This figure also illustrates the predicted camera Field of View (FOV) that optimizes coverage over the planning horizon. Triangular facets shaded in gray indicate the points of interest on these facets planned for coverage. The implemented coverage plans (i.e., the agents' trajectories and camera FOV states) are depicted with green, red, and blue lines for agents 1, 2, and 3 respectively. These colours also denote the points covered by each agent as shown in Fig. 2(b). As depicted in the figure the agents collaboratively plan non-myopic coverage strategies focusing on maximizing the coverage

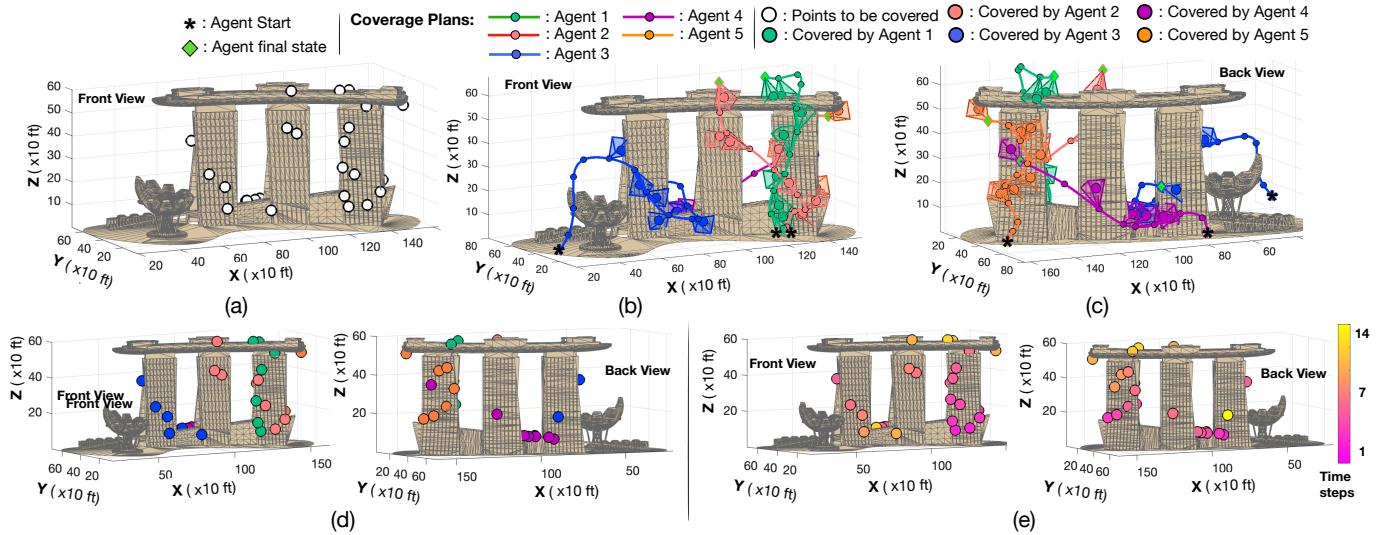


Fig. 3. Coverage of the Marina Bay Sands Hotel in Singapore with 5 collaborative UAV agents. (a) Points of interest for coverage shown as \circ , (b)(c) Final coverage trajectories, (d) Allocation of points to agents, and (e) Timing of point coverage during the mission.

of uncovered points while minimizing the duplication of work. Finally, Fig. 2(c) offers a top-down view of how the points of interest have been allocated among the agents for coverage, and Fig. 2(d) indicates the point in time each point was covered.

Next, Fig. 3 shows the proposed approach applied for the task of 3D structural inspection of a real-world building i.e., to a 3D reconstruction of the Marina Bay Sands (MBS) hotel in Singapore. The MBS, consisting of three towers as seen in Fig. 3(a), stands approximately 200m (656 ft) tall. These towers are linked by a 340m (1120 ft) long skybridge as depicted. In this experiment we randomly selected 35 points (marked as white \circ) from the triangulated surface mesh of the building as illustrated in Fig. 3(a). Five agents are positioned at the coordinates (125, 10, 5), (115, 10, 5), (20, 20, 5), (80, 80, 5), and (160, 60, 5) for agents 1 (green), 2 (pink), 3 (blue), 4 (purple), and 5 (orange) respectively. Maintaining the previously discussed setup, Fig. 3(b)(c) display the final executed coverage plans by the five agents achieving full coverage of all points of interest within 15 time-steps. The proposed approach generated collaborative trajectories for optimized coverage with the allocation of points to agents clearly shown in Fig. 3(d) and the timing of point coverage during the mission shown in Fig. 3(e). We should note here that, in order to handle obstacle avoidance in such complex, non-convex structures, the object of interest is first decomposed into several convex sub-regions. This decomposition allows us to utilize the constraints shown in Eq. (5j) in each of those sub-regions to avoid collisions.

Our next goal is to evaluate the proposed approach's performance concerning computational complexity as shown in Fig. 4(a)(b), and performance i.e., mission completion time, as depicted in Fig. 4(c), comparing it with other methods. For this purpose, we have conducted a Monte Carlo (MC) simulation randomly initializing the states of N agents, where $N \in \{1, 3, 5, 7\}$, within the environment \mathcal{E} as per the setup described in Sec. VV-A for covering 35 points of interest on the object depicted in Fig. 2. The first set of experiments measures the average runtime required to solve one iteration of Problem (P1) and generate a solution, varying the planning horizon's length and the number of agents, and then comparing these findings with the centralized approach from [16]. It is important to mention that the computational complexity of a mixed integer program (MIP), such as Problem (P1), is generally influenced by the number of binary variables. This is because the primary optimization method for solving MIPs, a branch-and-bound variant [21], [22], constructs a search tree that enumerates potential solutions systematically. The size of this tree which directly affects computational complexity is determined by the number of integer and binary variables. Therefore, the primary factor affecting computational complexity, as illustrated in Problem (P1), is the length of the planning horizon. Conversely, the centralized approach mentioned in [16] also depends on the number of agents for generating collaborative coverage plans. Figure 4(a) presents the average runtime over 50 MC trials for generating coverage plans with our distributed approach for a planning horizon of $T = 3$ time-steps (blue bar) compared to the centralized approach (gray hatched bar) for scenarios with 1, 3, 5 and 7 agents. As evident from Fig. 4(a), while both approaches are comparable in runtime for the single agent scenario, the centralized controller's computational complexity increases exponentially with the number of agents indicating poor scalability. Conversely, the proposed distributed approach's performance is not influenced by the number of agents, maintaining its ability to generate non-myopic collaborative plans efficiently. Similar findings are depicted in Fig. 4(b) for a planning horizon of $T = 6$, where the proposed method's performance is influenced only by the planning horizon's length, not by the number of agents.

The subsequent experiment assesses the performance of the proposed approach in terms of mission completion time, i.e., the duration required for agents to cover all points of interest. This analysis involved 50 MC trials with $N = 5$ agents

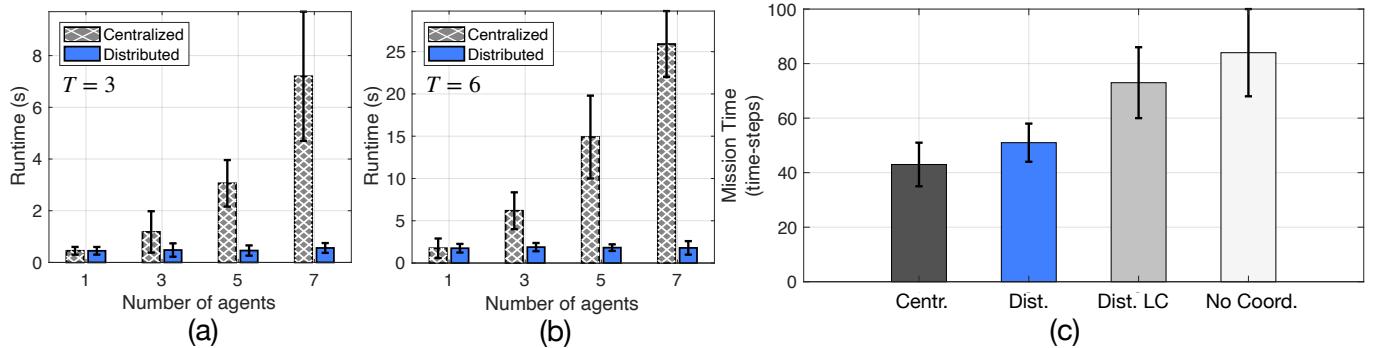


Fig. 4. Performance evaluation of the proposed approach. (a)(b) Computational complexity of centralized ([16]) and distributed (proposed approach) coverage planning with respect to the number of agents and the length of the planning horizon. (c) Performance comparison with competing approaches.

randomly placed within the environment to cover 35 points of interest, using a planning horizon of length $T = 6$. Besides comparing to the centralized approach we also evaluate two variations: distributed with limited communication (Dist. LC) and multi-agent coverage without coordination (No Coord.). In the Dist. LC setup, agents collaborate opportunistically within a 3m communication range with the lowest-index agent initiating the proposed approach. Agents within communication range also exchange their coverage databases Q_n , whereas those outside generate plans independently following Problem (P1). The No Coord. approach involves agents always planning independently without collaborative strategies, though agents within communication range share their coverage databases Q_n .

It is important to mention here that with regard to the approaches Dist. LC and No Coord., in scenarios where the agents are not in communication range, the guidance objective in Eq. (9) cannot be computed due to the lack of information required for solving the assignment problem. For this reason, each agent n computes its nearest unobserved point of interest in a greedy manner by calculating: $\arg \min_{\tilde{p}} \| \mathbf{x}_n(t + \tau|t) - \tilde{\mathcal{P}}_n(\tilde{p}) \|_2^2$, where $\tilde{\mathcal{P}}_n$ is a list of unobserved points of interest based solely on agent n 's record of observed points Q_n .

Figure 4(c) displays the average mission completion time for these four methods. The centralized approach completes the mission in about 40 time-steps, with the proposed distributed method being roughly 15% slower, but 6.5 times more computationally efficient, as shown in Fig. 4(b). The Dist. LC variant highlights the significant role of communication in forming collaborative plans and influencing mission completion times, further emphasized by the No Coord. approach's performance which lacks inter-agent communication for collaborative plan generation. These experiments illustrate the trade-offs between computational complexity, performance, and communication, showcasing the proposed approach's adaptability to diverse scenarios and needs.

Finally, we have conducted experimental tests of the proposed approach for covering the surface area of the University of Cyprus Library building, an oval-like structure with major and minor axis diameters of approximately 170m and 85m, respectively, and a height of about 25m, as depicted in Fig. 5(a). The building's surface was first converted from a 3D point cloud into a mesh of 82 triangular facets as illustrated in Fig. 5(b). Using the proposed approach we generated collaborative coverage trajectories for 3 UAV agents tasked with covering the building's surface area, depicted in Fig. 5(b). This figure specifically presents a top-down view of the building and illustrates the trajectories generated by the 3 agents, ensuring coverage of the entire area. The covered points are colour-coded to match the colour of the agents that covered them i.e., green, red, and blue, for agents 1,2 and 3 respectively.

Implementation details: The UCY library building area was pre-scanned using a single quadrotor UAV equipped with a camera and lidar. This process involved capturing multiple images and raw lidar data, which were subsequently processed to generate a 3D point cloud. The point cloud was triangulated to create the 3D mesh shown in Fig. 5(b). The experimental setup utilized three DJI Mavic Enterprise UAVs, as illustrated in Fig. 5(d). Each UAV was equipped with a 12MP gimbal-stabilized camera and operated at a safety-maximizing velocity of 2 m/s. The UAVs were controlled using the DJI Mobile Software Development Kit (SDK), which enabled the development of a custom Android mobile application running on mobile phones assigned to the UAVs. Each mobile phone running the custom SDK-enabled application wirelessly connected to its respective drone's remote controller, acting as an intermediary to facilitate command transmission to the drone and the reception of telemetry data (i.e., Fig. 5(e)), such as GPS location, altitude, battery status, speed, and sensor information. Due to SDK constraints and hardware limitations (specifically, the lack of direct access to the drones' onboard controllers), the proposed distributed model predictive coverage controller could not be implemented on the UAVs' onboard control systems. Instead, a ground control station (GCS) executed the distributed controller in Matlab and transmitted the resulting trajectories to the UAVs through the mobile applications running on the Android phones connected to each drone's remote controller. The GCS also received telemetry data from the UAVs via this application. Communication between the GCS and the UAVs was established through a VPN server over a wireless network, allowing for data exchange between agents and the GCS. The live mission

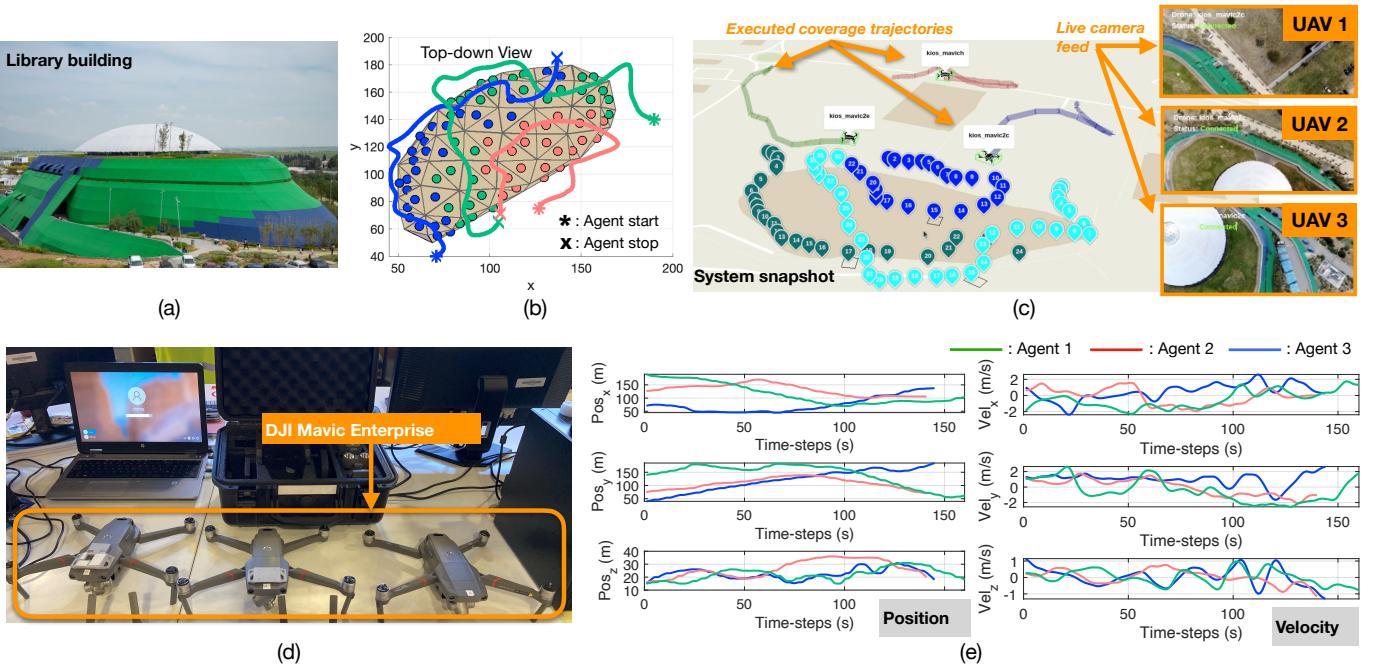


Fig. 5. Real-world experimental evaluation of the proposed collaborative 3D coverage controller: (a) Building for coverage, (b) Generated collaborative coverage plans with 3 UAV agents, (c) Real-time monitoring of mission progress, (d) DJI Mavic drones used in the experiment, (e) Telemetry data.

monitoring was conducted using our dedicated multi-UAV multi-tasking platform [23], as illustrated in Fig. 5(c). While this prototype implementation was not optimal, it successfully demonstrated the proposed approach's effectiveness and revealed limitations and areas for future improvement. Notably, reliance on GPS positioning introduced errors and inconsistencies in UAV localization, resulting in distorted coverage plans in certain cases. Additionally, communication delays due to network latency affected the UAVs' operation, and environmental disturbances caused deviations from the planned trajectories. Finally, with the current prototype implementation, we were unable to evaluate the real-time capabilities of the proposed controller or thoroughly assess its performance in real-world scenarios. For future work, we aim to co-design the software and hardware components of the proposed controller to facilitate its deployment on real-world platforms. Promising avenues for further research include the development of efficient real-time mixed-integer programming (MIP) controllers via adaptive neighborhood search techniques [24], and learning-based approaches [25].

VI. WAY FORWARD IN UAV SWARM SYSTEMS

Swarm systems have the potential to revolutionize society, the economy, technology, and various other fields by enhancing efficiency, scalability, and adaptability across a wide range of applications. In emergency response, swarms of autonomous drones and ground vehicles can cooperatively patrol [26] and search areas not previously encountered, locate victims, gather real-time information, and execute complex missions with minimal human oversight, thereby enhancing situational awareness and operational capabilities. Meanwhile, in military contexts, swarm technologies have the potential to create coordinated defense mechanisms for surveillance and reconnaissance. These fields exemplify the potential of swarm system technologies in tackling complex challenges across diverse sectors through the application of collective intelligence principles. However, despite recent progress in various aspects of swarming systems [27]–[29], numerous challenges remain before these systems reach the required level of maturity. These challenges range from technological issues, such as the development of smarter, more compact machines, to interactive aspects, including the creation of innovative interfaces and modes of communication, and extend to socio-technical issues related to evaluating trustworthiness and addressing the ethical and legal ramifications of deploying swarm systems. In this work, we address several limitations of existing multi-UAV and swarm systems for coverage planning by developing an intelligent distributed model predictive controller. This controller incorporates light-path propagation constraints, allowing for the generation of optimal, complementary look-ahead trajectories for the efficient coverage of 3D objects of interest. However, numerous open challenges and limitations persist, and overcoming these obstacles will be crucial for advancing coverage planning and the field of swarm systems in general, as elaborated next:

a) Scalability and Coordination: A major challenge for UAV swarm systems is their scalability. As the swarm size grows, managing communication, coordination, and control becomes exponentially more complex. Existing algorithms and frameworks, including this work, struggle to scale effectively when the number and interactions among the UAVs increases. Currently this problem is handled by various distributed frameworks which although can handle large number of agents, they only generate myopic and greedy plans [29]. On the other hand, to enable the generation of complimentary look-ahead coverage plans,

this work requires a coordinated optimization sequence to be performed among the agents, which can potentially limit its applicability with large teams. Another limitation of the proposed technique is the dependence of the results on the agents' order of execution. This issue arises from the way the centralized problem was broken down and approximated with a distributed formulation. Essentially, by evaluating all possible permutations of the agents' execution order, the centralized solution can be recovered. However, this is computationally intractable, which is why we proposed this distributed approach. Future research should explore whether, in certain scenarios, there is an opportunity to adopt an intelligent strategy to determine the optimal sequence of agent execution, and prioritize the development of algorithms that are inherently scalable and capable of handling large numbers of UAVs without sacrificing performance. There are several avenues to explore here including learning-based techniques [30], as well as planning without coordination by allowing some degree of work duplication as shown in our previous work [31].

b) Autonomy and Decision Making: A crucial aspect of future UAV swarms is their ability to operate autonomously in complex and unknown environments while making intelligent and optimal decisions. This involves not only navigating through unfamiliar terrain but also determining the most efficient way to accomplish a given task while rapidly adapting to new information and changing conditions. One of the limitations of the proposed approach is the requirement of a known world model, which is essential for designing optimal coverage plans using the current methodology. Specifically, the proposed method requires prior knowledge of the 3D environment and points of interest, as well as pre-computed light-path propagation constraints. However, this information may not be available in many scenarios where UAV-based search is needed, such as search and rescue in natural disaster zones or inside collapsed buildings. Currently, state-of-the-art swarm systems are divided between model-based and model-free planning methodologies [27]. Model-based methodologies [28] rely on predefined environmental models for optimized planning, offering efficiency but lacking adaptability to changes. In contrast, model-free techniques like sampling-based approaches [15] adapt to unknown or dynamic environments, providing flexibility but lacking optimality. The proposed methodology can be extended to handle dynamic and unknown environments by employing a two-stage approach: first, building a world model through environmental mapping and scene reconstruction, and then utilizing this model for planning. This two-stage approach has been explored in our previous work [32] with a swarm of heterogeneous agents, demonstrating promising results. However, to enhance swarm system performance, future research should focus on creating more integrated or hierarchical planning techniques that can synergistically leverage the benefits of both model-based and model-free approaches. Additionally, integrating machine learning techniques [33] and other artificial intelligence methodologies [34], [35] can enhance the swarm's ability to learn from experience, improve its performance over time, and handle complex decision-making.

c) Robustness and Fault Tolerance: Robustness with respect to modelling errors and measurement noise, as well as fault tolerance with respect to hardware malfunctions is paramount for UAV swarm systems.. To operate effectively in real-world environments, swarms must be resilient to unpredictable conditions, varying weather, and hardware malfunctions. Future research should focus on developing advanced robust and fault-tolerant algorithms that enable the swarm to dynamically adapt to UAV losses, malfunctions, and uncertainty. Additionally, incorporating machine learning techniques could allow UAVs to predict potential failures and proactively adjust their behavior to prevent mission-critical consequences. While this work did not study robustness and fault tolerance, our previous research on fault-tolerant coverage planning under non-Gaussian disturbances [36] can provide valuable insights for future directions. Additionally, recent advancements in these fields, as highlighted in [37], offer further potential avenues for exploration.

Swarm systems, with their potential to revolutionize various fields, offer a promising solution to complex challenges that traditional centralized systems struggle to address. While significant strides have been made, numerous challenges remain to fully harness their potential. By addressing the challenges discussed above, swarm systems can become indispensable tools for solving a wide range of real-world problems.

VII. CONCLUSION

This study introduces a coverage control framework that allows a group of distributed autonomous agents to collaboratively plan look-ahead coverage plans over a rolling finite planning horizon, targeting the coverage of designated areas on a 3D object's surface. Formulated as a distributed model predictive control problem, the proposed approach optimizes the movement and visual control inputs of each agent, incorporating constraints to minimize overlapping work among agents. By integrating visibility determination through light-path propagation constraints, the controller can anticipate which areas of the object will be visible based on projected future positions of the agents. This is achieved by transforming non-linear visibility assessments constraints into logical constraints using binary variables within a mixed-integer programming framework. The effectiveness of this approach is validated both in simulation environments and through real-world UAV inspections of architectural structures.

VIII. ACKNOWLEDGEMENTS

This work is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE), and from the Government of the Republic of Cyprus through the Cyprus Deputy Ministry of Research, Innovation and Digital Policy.

REFERENCES

- [1] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms," *IEEE Access*, vol. 9, pp. 119310–119342, 2021.
- [2] J. Xie, L. R. Garcia Carrillo, and L. Jin, "Path planning for uav to cover multiple separated convex polygonal regions," *IEEE Access*, vol. 8, pp. 51770–51785, 2020.
- [3] 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.
- [4] J. Chleboun, T. Amorim, A. M. Nascimento, and T. P. Nascimento, "An improved spanning tree-based algorithm for coverage of large areas using multi-uav systems," *Drones*, vol. 7, no. 1, p. 9, 2022.
- [5] 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.
- [6] Y. Choi, Y. Choi, S. Briceno, 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.
- [7] G. S. Avellar, G. A. Pereira, L. C. Pimenta, and P. Iscold, "Multi-uav routing for area coverage and remote sensing with minimum time," *Sensors*, vol. 15, no. 11, pp. 27783–27803, 2015.
- [8] S. D. Apostolidis, P. C. Kapoutsis, A. C. Kapoutsis, and E. B. Kosmatopoulos, "Cooperative multi-uav coverage mission planning platform for remote sensing applications," *Autonomous Robots*, vol. 46, no. 2, pp. 373–400, 2022.
- [9] L. Collins, P. Ghassemi, E. T. Esfahani, D. Doermann, K. Dantu, and S. Chowdhury, "Scalable coverage path planning of multi-robot teams for monitoring non-convex areas," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7393–7399.
- [10] H. I. Perez-Imaz, P. A. Rezeck, D. G. Macharet, and M. F. Campos, "Multi-robot 3d coverage path planning for first responders teams," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2016, pp. 1374–1379.
- [11] W. Jing, D. Deng, Y. Wu, and K. Shimada, "Multi-uav coverage path planning for the inspection of large and complex structures," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 1480–1486.
- [12] S. Ivić, B. Crnković, L. Grbčić, and L. Matleković, "Multi-uav trajectory planning for 3d visual inspection of complex structures," *Automation in Construction*, vol. 147, p. 104709, 2023.
- [13] X. Kan, H. Teng, and K. Karydis, "Online exploration and coverage planning in unknown obstacle-cluttered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5969–5976, 2020.
- [14] A. Aminzadeh and A. Khoshnood, "Multi-uav cooperative search and coverage control in post-disaster assessment: experimental implementation," *Intelligent Service Robotics*, vol. 16, no. 4, pp. 415–430, 2023.
- [15] M. Jones, S. Djahel, and K. Welsh, "Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey," *ACM Comput. Surv.*, vol. 55, no. 11, feb 2023.
- [16] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "Cooperative receding horizon 3d coverage control with a team of networked aerial agents," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 4399–4404.
- [17] T. Elmokadem, "Distributed coverage control of quadrotor multi-uav systems for precision agriculture," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 251–256, 2019.
- [18] S. Huang, R. S. H. Teo, and W. L. Leong, "Review of coverage control of multi unmanned aerial vehicles," in *2017 11th Asian Control Conference (ASCC)*, 2017, pp. 228–232.
- [19] A. Richards and J. P. How, "Robust distributed model predictive control," *International Journal of control*, vol. 80, no. 9, pp. 1517–1531, 2007.
- [20] S. Papaioannou, P. Kolios, and G. Ellinas, "Distributed estimation and control for jamming an aerial target with multiple agents," *IEEE Transactions on Mobile Computing*, vol. 22, no. 12, pp. 7203–7217, 2023.
- [21] J. P. Vielma, "Mixed integer linear programming formulation techniques," *Siam Review*, vol. 57, no. 1, pp. 3–57, 2015.
- [22] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning," *Discrete Optimization*, vol. 19, pp. 79–102, 2016.
- [23] M. Terzi, P. Kolios, C. Panayiotou, and T. Theocharides, "A unified framework for reliable multi-drone tasking in emergency response missions," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 819–827.
- [24] G. Hendel, "Adaptive large neighborhood search for mixed integer programming," *Mathematical Programming Computation*, pp. 1–37, 2021.
- [25] Y. Wu, W. Song, Z. Cao, and J. Zhang, "Learning large neighborhood search policy for integer programming," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021.
- [26] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Menezes, V. Corruble, and Y. Chevaleyre, "Recent advances on multi-agent patrolling," in *Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29–October 1, 2004. Proceedings 17*. Springer, 2004, pp. 474–483.
- [27] Y. Zhou, B. Rao, and W. Wang, "Uav swarm intelligence: Recent advances and future trends," *IEEE Access*, vol. 8, pp. 183856–183878, 2020.
- [28] B. Zhu, L. Xie, and D. Han, "Recent developments in control and optimization of swarm systems: A brief survey," in *2016 12th IEEE international conference on control and automation (ICCA)*. IEEE, 2016, pp. 19–24.
- [29] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.
- [30] D. Bertsimas and B. Stellato, "Online mixed-integer optimization in milliseconds," *INFORMS Journal on Computing*, vol. 34, no. 4, pp. 2229–2248, 2022.
- [31] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "Distributed search planning in 3-d environments with a dynamically varying number of agents," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 7, pp. 4117–4130, 2023.
- [32] A. Anastasiou, A. Zacharia, S. Papaioannou, P. Kolios, C. G. Panayiotou, and M. M. Polycarpou, "Automated real-time inspection in indoor and outdoor 3d environments with cooperative aerial robots," in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2024, pp. 496–504.
- [33] J. Westheimer, J. Ruckin, and M. Popovic, "Multi-uav adaptive path planning using deep reinforcement learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 649–656.
- [34] G. Booch, F. Fabiano, L. Horesh, K. Kate, J. Lenchner, N. Linck, A. Loreggia, K. Murgesan, N. Mattei, F. Rossi *et al.*, "Thinking fast and slow in AI," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 17, 2021, pp. 15042–15046.
- [35] S. Papaioannou, P. Kolios, C. G. Panayiotou, and M. M. Polycarpou, "Synergising human-like responses and machine intelligence for planning in disaster response," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–8.
- [36] S. Papaioannou, C. Vitale, P. Kolios, C. G. Panayiotou, and M. M. Polycarpou, "Hierarchical fault-tolerant coverage control for an autonomous aerial agent," *IFAC-PapersOnLine*, vol. 58, no. 4, pp. 532–537, 2024, 12th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2024.
- [37] M. Saeid, H. Shraim, and C. Francis, "A review on recent development of multirotor uav fault-tolerant control systems," *IEEE Aerospace and Electronic Systems Magazine*, pp. 1–30, 2023.