

Automated Real-Time Inspection in Indoor and Outdoor 3D Environments with Cooperative Aerial Robots

Andreas Anastasiou, Angelos Zacharia, Savvas Papaioannou,
Panayiotis Kolios, Christos G. Panayiotou, and Marios M. Polycarpou

Abstract—This work introduces a cooperative inspection system designed to efficiently control and coordinate a team of distributed heterogeneous UAV agents for the inspection of 3D structures in cluttered, unknown spaces. Our proposed approach employs a two-stage innovative methodology. Initially, it leverages the complementary sensing capabilities of the robots to cooperatively map the unknown environment. It then generates optimized, collision-free inspection paths, thereby ensuring comprehensive coverage of the structure’s surface area. The effectiveness of our system is demonstrated through qualitative and quantitative results from extensive Gazebo-based simulations that closely replicate real-world inspection scenarios, highlighting its ability to thoroughly inspect real-world-like 3D structures.

Index Terms—Cooperative robots, 3D inspection, intelligent systems, Planning and control.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs), have nowadays, revolutionized numerous fields by providing a versatile platform for a plethora of applications. Such applications include search and rescue [1]–[4], search and track [5]–[7], security operations [8]–[10], and infrastructure inspection [11]–[13]. In this work we will focus on 3D infrastructure inspection which involves the examination of structures such as buildings, bridges, power lines, and wind turbines to detect any anomalies or damages. Traditional methods of inspection often involve manual labor, which can be time-consuming, costly, and potentially dangerous. Quadrotor UAVs, with their high maneuverability, can easily navigate and capture high-resolution images or videos, providing a safer and more efficient alternative. The use of UAVs for 3D infrastructure inspection involves equipping the drones with various sensors like cameras to capture detailed information about the structures. Advanced algorithms and techniques are then used to process and analyze this data, identifying potential issues that may require further investigation or immediate attention. The recent advancements in aerial robotics, particularly UAVs, have significantly broadened the horizon of possibilities for using multiple drones in a coordinated manner for large-scale and complex 3D infrastructure inspections. This not only increases the efficiency of the inspection process but also reduces the time taken to successfully accomplish the task

The authors are with the KIOS Research and Innovation Center of Excellence (KIOS CoE) and the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, 1678, Cyprus.{anastasiou.andreas, zacharia.angelos, papaioannou.savvas, pkolios, christosp, mpolycar}@ucy.ac.cy, (Andreas Anastasiou and Angelos Zacharia contributed equally to this work.)

at hand. However, despite the promising benefits, there are still challenges to be addressed, such as navigating unknown indoor and outdoor environments with obstacles, inspecting complex structures, and the complexity of devising mechanisms for the real time coordination and efficient utilization of robots with heterogeneous capabilities.

This work was motivated by the Cooperative Aerial Robots Inspection Challenge (CARIC) competition, held during the 2023 IEEE Conference on Decision and Control (CDC), whose purpose was to simulate and benchmark multi-UAV infrastructure inspection methodologies in real-world-like inspection scenarios. As such, in this paper, we propose a novel Cooperative Aerial Robots Inspection (CARI) methodology for inspecting complex 3D infrastructures in both indoor and outdoor environments. Our approach considers a system of multiple heterogeneous quadrotor UAVs each equipped with different sensors, such as gimbal cameras and LiDAR, each capable of inspecting the surface area of the infrastructure of interest using only limited knowledge about the environment, under communication and time constraints. To evaluate the quality of the inspection carried by the fleet of UAVs, both metrics for the level of bluriness and the level of resolution are employed assessing the observations of the camera sensors. The key contributions of the proposed scheme are as follows:

- We propose a cooperative inspection system designed for the efficient coordination and control of a distributed team of heterogeneous UAV agents assigned to inspect 3D structures in complex, cluttered indoor and outdoor environments. Our approach employs a two-stage strategy. Initially, it leverages the heterogeneous sensing capabilities of the UAVs to cooperatively map the unknown environment. Subsequently, it generates cooperative, collision-free inspection paths in real time, thereby ensuring full coverage of the structure’s surface area.
- The effectiveness of our system is demonstrated through extensive qualitative and quantitative results from Gazebo-based simulations. These simulations closely replicate real-world inspection scenarios, underscoring the system’s capability and practicality to thoroughly inspect 3D structures that closely resemble those found in the real-world settings.

The rest of this paper is structured as follows. Section II delves into related work on infrastructure inspection. In Section III, we introduce the system model. Following

this, the infrastructure inspection problem is introduced in Section IV. Our proposed approach for online cooperative 3D inspection planning is presented in Section V. Section VI focuses on the evaluation of our approach, and finally Section VII concludes the paper.

II. RELATED WORK

Infrastructure inspection is a critical part of ensuring the safety and functionality of essential facilities such as spanning bridges, pipelines, and other civil structures. Traditional inspection practices often pose challenges in terms of cost, safety, and efficiency. As technology advances, there is a growing exploration of robotic systems as a viable alternative for infrastructure inspection. These systems have the potential to mitigate risks associated with human inspectors, enhance data collection through advanced sensing technologies, and provide a cost-effective and efficient means of evaluating the structural integrity of vital infrastructure components. This shift towards robotic inspection solutions is underpinned by the need for innovations in mobility, autonomy, and sensing, as well as a growing emphasis on purpose-driven designs and multidisciplinary collaborations [14], [15].

In the literature, numerous works explore the use of a single UAV for efficiently generating optimal paths to facilitate infrastructure inspection. More specifically, the study outlined in [16] introduces a sampling-based approach for visual path planning. Designed for a camera-equipped UAV, this method visually covers a specific target of interest. Recasting the problem of coverage path planning (CPP) as a mixed-integer programming (MIP) problem has also been considered in the literature. The authors in [17] proposed a CPP algorithm for visually covering a 2D object of interest by formulating a MIP that utilizes a set of visibility constraints (i.e. camera-rays). Moreover, the authors in [18] address a 3D case of the CPP by recasting it as a MIP problem, enabling a single-UAV system to efficiently cover cuboid-like objects of interest. Our previous work, [19], introduces a UAV-based inspection planning approach utilizing a novel receding horizon MIP control strategy. The resulting paths achieve optimal inspection of feature points scattered on the surface of the target object. Furthermore, the authors in [20] propose a path planning methodology for information gathering utilizing a single-UAV system. The proposed approach involves conducting a discrete search by generating optimal paths, based on a predetermined set of viewpoints, that maximize the information gathered based on recent observations. In [21], a framework has been introduced to address 3D coverage by directly sampling and planning paths within intricate 3D environments. The approach uses voxel dilation and subtraction based sampling to extract via-points and path primitives around the target. Subsequently, a primitive coverage graph is proposed that encodes the extracted information and finally graph search is used to calculate the inspection paths. In [22] an algorithm for 3D inspection path planning for multiple structures using time-constrained aerial robots is presented. It optimizes paths

by considering spatial properties and importance of infrastructure facilities, adhering to robot constraints, employing multi-step optimization.

Improving the path for a single UAV does not lead to a substantial decrease in inspection time when contrasted with employing multiple UAVs. The latter not only accelerates the inspection process but also strengthens the system resilience, by eliminating the single point of failure. To address the limitations of single-UAV systems, researchers have explored the use of multi-UAV approaches. This allows for the development of systems that can assign tasks to UAVs for cooperative infrastructure inspection, enhancing the reliability, accuracy, and speed of the inspection process. In [23], a framework for multi-UAV Coverage Path Planning (CPP) is presented for inspecting large, complex 3D structures. This framework combines the Set Covering Problem and the Vehicle Routing Problem into a Set-Covering Vehicle Routing Problem, which is solved using a modified Biased Random Key Genetic Algorithm. [24] proposes a novel scheme to solve the CPP problem by slicing the infrastructure into horizontal planes and assigning areas to each UAV. [25] introduces a path planning algorithm for multiple UAVs, featuring a Next-Best-View planner for surface reconstruction, using a greedy allocation of configurations and the Traveling Salesman Problem. It also employs a volumetric representation of surfaces, called Truncated Signed Distance Function, for cautious navigation and evaluation of 3D reconstruction. [26] presents a unique path planning algorithm based on a potential field approach. This algorithm uses a refined version of the Heat Equation Driven Area Coverage technique to minimize the difference between desired and actual coverage. It also incorporates a camera direction control feature that ensures comprehensive coverage of the structure within the sensor's field of view. Lastly, [12] introduces an online trajectory planning algorithm for a multi-UAV system to thoroughly inspect large-scale complex structures in a 3D environment. This is achieved by utilizing a distributed control scheme for each UAV that operates based on local measurements.

III. PRELIMINARIES

In this work, we consider a fleet of heterogeneous quadrotor UAVs employed to inspect the surface of an infrastructure of interest, while avoiding potential static and dynamic obstacles in an unknown cluttered environment, under communication constraints. The fleet is composed of two types of UAVs: (a) explorer UAVs, equipped with rotating LiDAR and gimbal camera sensors, and (b) photographer UAVs, which are smaller and only carry a gimbal camera sensor.

A. Operational Environment

The operational environment, in which the UAVs are deployed, is an agnostic cluttered volume that needs to be mapped in real time to facilitate the identification of static obstacles and the infrastructure to be inspected. The infrastructure targeted for inspection can be represented by a triangular mesh. We assume the presence of a finite

set $\mathcal{Z} \subset \mathbb{R}^3$ consisting of points of interest uniformly distributed across the surface of the infrastructure. These points must be discovered and observed, in real-time, by the UAV agents using their payload sensors. The quality of the visual inspection (i.e., how well a particular point $\zeta \in \mathcal{Z}$ is observed) is determined by a scoring scheme further described in III-C. In this work, since we consider no prior map for the environment, the system operator restricts the operational volume by defining a set $\mathcal{B}$ of bounding boxes, thereby focusing the exploration to areas of high importance.

B. UAV Dynamical Model

We consider N quadrotor UAVs, of which $N_e \in \{1, 2\}$ are explorer UAVs and $N_p \in \{0, \dots, N - N_e\}$ are photographer UAVs, each free to move in the 3D Cartesian space. The inertial coordinate frame is defined in this space as $\mathbf{I} = \{x_I, y_I, z_I\}$. The discretized dynamics of the quadrotor UAV agent can be expressed as:

$$\mathbf{x}_k = \begin{bmatrix} \mathbb{I}_{4 \times 4} & \delta t \cdot \mathbb{I}_{4 \times 4} \\ \mathbb{O}_{4 \times 4} & \mathbb{I}_{4 \times 4} \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} \mathbb{O}_{4 \times 4} \\ \delta t \cdot \mathbb{I}_{4 \times 4} \end{bmatrix} \mathbf{u}_{k-1} \quad (1)$$

where $\mathbf{x}_k = [\mathbf{p}_k, \psi_k, \mathbf{v}_k, \dot{\psi}_k]^T \in \mathbb{R}^8$ represents the state of the UAV agent at timestep k , comprising the position $\mathbf{p}_k \in \mathbb{R}^3$ and velocity $\mathbf{v}_k \in \mathbb{R}^3$, both in the inertial coordinate frame $\mathbf{I}$. The yaw angle and its angular velocity are denoted by ψ_k and $\dot{\psi}_k$, respectively. Note that $\mathbb{I}_{4 \times 4}$ and $\mathbb{O}_{4 \times 4}$ represent the 4×4 identity and zero matrices, respectively. Additionally, δt stands for the sampling interval. Moreover, $\mathbf{u}_k = [u_k^x, u_k^y, u_k^z, u_k^\psi] \in \mathbb{R}^4$ denotes the control input applied to the UAV, consisting of the linear and yaw acceleration in the inertial coordinate frame $\mathbf{I}$.

C. Ranging and Visual Sensor Models

For efficiently reconstructing the unknown environment in real-time, the explorer UAVs are equipped with a LiDAR sensor, capable at providing dense measurements that represent the environment. More specifically, the measurement provided by the LiDAR sensor consists of a point-cloud drawn from the surrounding 3D space. Thus, the returned points lay on the surface of the nearest obstacles in the environment that are within the sensor's limited sensing range. This range is defined as a sphere-like shape around the sensor of radius r_L . The utilization of these measurements is crucial for obstacle avoidance and path planning purposes.

Furthermore, all UAV agents are equipped with gimballed camera sensors that possess rotation capabilities and are utilized for observing the interest points in $\mathcal{Z}$ with a certain quality. In this work, the camera view is represented as a rectangular pyramid, with its apex located at the camera's optical center. The pyramid is defined by the horizontal ϑ_h and vertical ϑ_v Field of View (FoV) angles, and the effective range r_C , with its volume is denoted as $\mathcal{C}$. On command, the FoV can be oriented in a specific direction by rotating the gimbal to the desired inclination angle $\theta \in [\theta_{\min}, \theta_{\max}]$ and azimuthal angle $\phi \in [\phi_{\min}, \phi_{\max}]$, adhering to the gimbal's rotation constraints. The agent observes an interest point ζ within its camera FOV at a certain quality level

(i.e., observation quality) based on the following observation metrics: (a) $q_{\zeta, \text{blur}} \in [0, 1]$, describing the level of blurriness with which the point was captured, and (b) $q_{\zeta, \text{res}} \in [0, 1]$, indicating the level of resolution. The level of blurriness is interpreted as the number of pixels that an interest point moves across during the exposure time τ and is derived from:

$$q_{\zeta, \text{blur}} = \min \left(\frac{c}{\max(\|u_1 - u_0\|, \|v_1 - v_0\|)}, 1.0 \right) \quad (2a)$$

$$u_0 = f \cdot \frac{x_0}{z_0}, u_1 = f \cdot \frac{x_1}{z_1}, v_0 = f \cdot \frac{y_0}{z_0}, v_1 = f \cdot \frac{y_1}{z_1} \quad (2b)$$

$$[x_1, y_1, z_1]^T = [x_0, y_0, z_0]^T + \mathbf{v} \cdot \tau \quad (2c)$$

where c is the pixel width, f is the focal length, $[x_0, y_0, z_0]^T$ is the position of the captured point of interest, and $\mathbf{v} \in \mathbb{R}^2$ is the velocity of the interest point in the camera frame at the time of capture [27]. Moreover, the resolution level is ensured to be satisfactory by assessing the estimated real-world size captured within a single image pixel (expressed in millimeters-per-pixel) and assigning a score based on:

$$q_{\zeta, \text{res}} = \min \left(\frac{r_{\text{des}}}{\max(r_{\text{horz}}, r_{\text{vert}})}, 1.0 \right) \quad (3a)$$

$$r_{\text{horz}} = \frac{c}{\|u_4 - u_3\|}, r_{\text{vert}} = \frac{c}{\|v_4 - v_3\|} \quad (3b)$$

where r_{des} is the desired resolution, and (u_3, u_4) and (v_3, v_4) represent the shifted positions of the interest point in x -axis and y -axis, respectively, in the camera frame.

The point of interest ζ is observed at time-step k when:

$$\zeta \in \bigcup_{i \in \{1, \dots, N\}} \mathcal{C}_{i, k}, \quad q_{\zeta, k} > \bar{q} \quad (4)$$

where, $q_{\zeta, k} = q_{\zeta, \text{blur}, k} \cdot q_{\zeta, \text{res}, k}$ is the observation quality of point ζ , and $\mathcal{C}_{i, k}$ is the FoV of the i_{th} UAV agent, both at time-step k . Parameter $\bar{q}$ is a tuning parameter that describes the lower admissible value of observation quality for each point of interest.

D. Communication Model

The exchange of information between UAV agents occurs in a peer-to-peer fashion only when two agents are within Line of Sight (LoS) with each other. In other words, LoS means that there are no obstructions, such as buildings or other obstacles, blocking the direct view between two UAVs. If an obstruction is present, communication is not possible. The group of agents that agent i can communicate with at a specific time-step k is represented as $\mathcal{N}_{i, k}$, with $i \in \{1, \dots, N\}$. The network topology can change over time due to agent movement, which may affect LoS. Therefore, in this work the agents opportunistically communicate whenever they are within LoS conditions and exchange information such as their current positions, and knowledge of the environment (i.e., their maps) to cooperatively map the environments, and coordinate the inspection task.

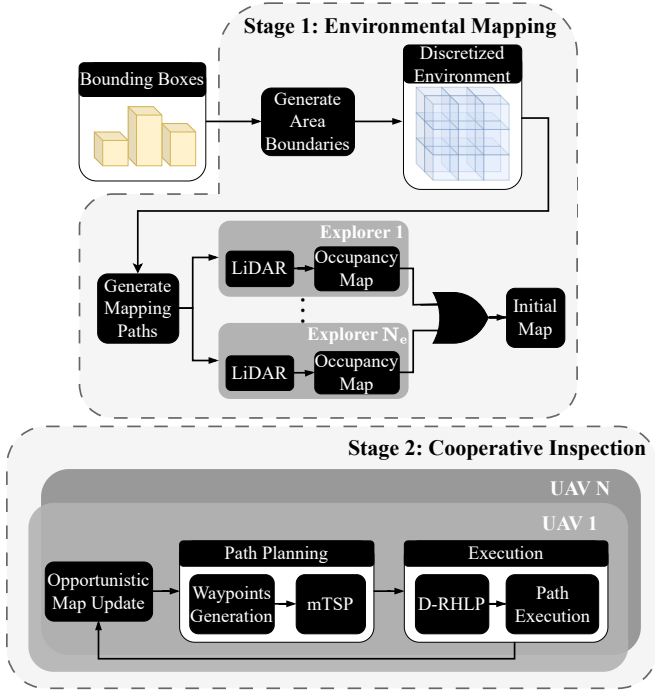


Fig. 1: Overview of the proposed approach for 3D infrastructure inspection using multi-UAV system.

IV. PROBLEM STATEMENT

An infrastructure within an unknown cluttered environment has to be inspected with the maximum observation quality while adhering to a time limit constrain. A heterogeneous UAV fleet, governed by Eq. (1), is employed to observe interest points across the infrastructure's surface using several payload sensors under communication limitations. The observation quality is assessed by the scoring metrics described in Sec. III-C. The 3D infrastructure inspection problem tackled in this work can be stated as follows: *Given a set of bounding boxes $\mathcal{B}$, a heterogeneous team of N UAV agents ($i \in \{1, \dots, N\}$), and mission time $\mathcal{T}$, find a set of cooperative inspection paths (one for every agent) that maximizes the inspection score $\mathcal{Q}$ given by:*

$$\mathcal{Q} = \sum_{\zeta \in \mathcal{Z}} \max_{i \in \{1, \dots, N\}} \left(\max_{k \in \{1, \dots, \mathcal{T}\}} q_{\zeta, i, k} \right). \quad (5)$$

This entails finding the joint kinematic and sensing control inputs of the agents that enable efficient and safe mapping of the environment while avoiding obstacles, discovering points of interest on the structure's surface, and observing them in a manner that maximizes observation quality and consequently the inspection score within the mission time.

V. PROPOSED APPROACH

This section outlines our proposed methodology for inspecting 3D infrastructures using multiple UAVs. This work considers N quadrotor UAVs, of which $N_e \in \{1, 2\}$ are explorer UAVs and $N_p \in \{0, \dots, N - N_e\}$ are photographer UAVs, each one capable of inspecting the surface area $\mathcal{S}$ of

Algorithm 1 Proposed CARI Scheme

Require: A set of bounding boxes $\mathcal{B}$, enclosing the interest points in $\mathcal{Z}$, and the mission time $\mathcal{T}$.
Ensure: Maximization of $\mathcal{Q}$.

Stage 1 – Environmental Mapping

- 1: Retrieves $\mathbf{p}_{i,0} \in \mathcal{P}$, $i = \{1, \dots, N\}$
- 2: Calculates the boundaries of $\mathcal{O}$, as in (6)
- 3: Forms $\mathcal{G}$ by dividing $\mathcal{O}$ by the voxel size V .
- 4: Generates N_e mapping paths that traverse $\mathcal{O}$
- 5: **for each** explorer $i \in \{1, \dots, N_e\}$ **do**
- 6: **while** Executing mapping path **do**
- 7: Gathers LiDAR measurements
- 8: Constructs and updates an occupancy map $\mathcal{M}_{i,k}$
- 9: **end while**
- 10: Shares $\mathcal{M}_{i,k}$ with j UAV, $j \in \mathcal{N}_{i,k}$
- 11: Receives $\mathcal{M}_{j,k}$, $\forall j \in \mathcal{N}_{i,k}$
- 12: Creates an initial map $\mathcal{M}_{i,k}$ by merging the maps
- 13: **end for**

Stage 2 – Cooperative Inspection

- 14: **for each** UAV $i \in \{1, \dots, N\}$ **do**
- 15: **while** true **do**
- 16: Shares, receives and updates $\mathcal{M}_{i,k}$
- 17: Generates w_{l,i,κ_i} and $\hat{n}_{l,i,\kappa_i}$, $l = \{1, \dots, |\mathcal{W}_{i,\kappa_i}|\}$
- 18: Retrieves $\sigma_{i,\kappa_i} \subseteq \mathcal{W}_{i,\kappa_i}$ from Algorithm 2
- 19: **for each** $w_{l,i,\kappa_i} \in \sigma_{i,\kappa_i}$ **do**
- 20: Applies D-RHLP to reach w_{l,i,κ_i}
- 21: Rotates the camera based on $\hat{n}_{l,i,\kappa_i}$
- 22: Retrieves $q_{\zeta,i,k}$ for each $\zeta \in \mathcal{C}_{i,k}$
- 23: **end for**
- 24: $\kappa_i \leftarrow \kappa_i + 1$
- 25: **end while**
- 26: **end for**

the infrastructure of interest. The proposed CARI approach is divided into two key stages: (a) operational volume determination and environmental mapping, and (b) path generation for cooperative inspection. Figure 1 depicts the overview of the two-stage approach. More specifically, at the initial stage an operational volume $\mathcal{O}$ is constructed with its boundaries calculated based on the set of bounding boxes $\mathcal{B}$ that enclose the set of interest points $\mathcal{Z}$ to be inspected. Stage one is also responsible for the environment mapping by the explorer UAVs generating an initial map of the environment. Finally, at stage two, the fleet of UAVs cooperatively generate and execute inspection paths in a manner that maximizes $\mathcal{Q}$. This process is performed in real-time in a closed-loop fashion until the end of the mission time $\mathcal{T}$. In the following sections, we will dive into the specifics of each component and discuss how these components interact and complement each other to create a comprehensive, robust, and scalable CARI system. This work operates on the assumption that each UAV possesses the capability to self-localize within the

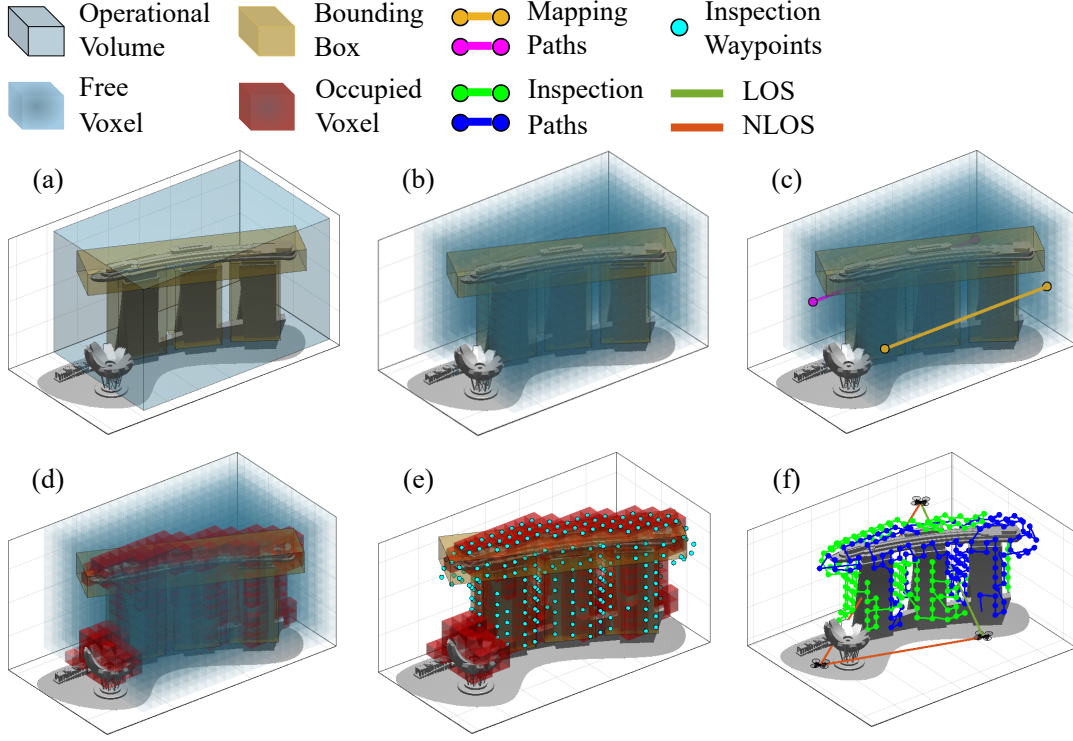


Fig. 2: (a) Derivation of the operational volume, (b) Discretization of the operational volume, (c) Mapping path generation and execution, (d) Initial occupancy map generation, (e) Inspection waypoint generation, (f) Inspection path generation.

environment. The overall process is outlined in Algorithm 1.

A. Operational Volume Determination

The methodology proposed in this work starts with the determination of the area boundaries and the discretization resolution of the operational volume $\mathcal{O}$. As mentioned above, the only available information about the infrastructure to be inspected is a set of bounding boxes $\mathcal{B}$ that enclose the set of interest points $\mathcal{Z}$. The operational volume boundaries are thus calculated based on $\mathcal{B}$ and the initial positions of the UAV agents $\mathcal{P} = \{\mathbf{p}_{i,0}\}_{i=1}^N, \mathbf{p}_{i,0} \in \mathbb{R}^3$. Thus, Eq. (6), calculates the minimum and maximum values of each axis resulting in the creation of the smallest cuboid containing both the infrastructure to be inspected and the UAV fleet. A visual representation of this cuboid is shown in Fig. 2a.

$$\begin{aligned} x\text{-axis: } & \begin{cases} x_{\min} = \min(\min_x \mathcal{B}, \min_x \mathcal{P}) \\ x_{\max} = \max(\max_x \mathcal{B}, \max_x \mathcal{P}) \end{cases} \\ y\text{-axis: } & \begin{cases} y_{\min} = \min(\min_y \mathcal{B}, \min_y \mathcal{P}) \\ y_{\max} = \max(\max_y \mathcal{B}, \max_y \mathcal{P}) \end{cases} \\ z\text{-axis: } & \begin{cases} z_{\min} = \min(\min_z \mathcal{B}, \min_z \mathcal{P}) \\ z_{\max} = \max(\max_z \mathcal{B}, \max_z \mathcal{P}) \end{cases} \end{aligned} \quad (6)$$

As the area boundaries are defined, a 3D grid is formed by dividing $\mathcal{O}$ into voxels of size $V \times V \times V$ meters, as illustrated in Fig. 2b. The resulting grid is treated as a connected mesh graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} = \{(u, v) \mid \|u - v\| < V, \forall u, v \in \mathcal{V}, u \neq v\}$, with each voxel representing a vertex. Subsequently, each UAV agent in the fleet is aware of the adjacency matrix that describes $\mathcal{G}$.

B. Environmental Mapping

The second step in Stage 1 is crucial component in our proposed methodology. Its responsibility is to construct an initial map of $\mathcal{O}$. This stage is solely reliant on the explorer UAVs, which are equipped with LiDAR sensors. These sensors are capable of capturing detailed and accurate spatial data of the surrounding environment, providing a rich dataset for our mapping process. Upon the successful construction of $\mathcal{G}$ from the previous step, the mapping process is initiated with the generation of N_e paths in a way that the explorer UAVs traverse $\mathcal{O}$ along its longest axis (see Fig. 2c). While executing the generated paths, each explorer utilizes measurements of the environment gathered by its LiDAR sensor. It independently constructs an occupancy map $\mathcal{M}_{i,k}$ by identifying the voxels in the 3D grid containing the 3D coordinates of the measurements (representing points of obstacles) and marking them as occupied. Once the paths are executed by the agents, they cooperatively integrate their individual occupancy maps by merging them, thereby creating an initial map of $\mathcal{O}$ shown in Fig. 2d. The initial map is then shared to every photographer UAV in the fleet opportunistically (i.e., when there is LoS with an explorer UAV). This initial map serves as the foundation for the subsequent stage of the infrastructure inspection process. It provides a preliminary layout of the operational area, allowing planning and executing the steps further needed for a successful and effective inspection. This is achieved by eliminating the inward edges of the corresponding occupied

vertices in $\mathcal{G}$, thus preventing any path from traversing through an occupied voxel.

C. Cooperative Inspection

In the final stage of the inspection mission, each UAV takes advantage of any opportunity to engage in the exchange and update of its occupancy map $\mathcal{M}_{i,k}$ with every other UAV within its LoS. This process, is a dynamic one, allowing for real time updates and adjustments to the map based on the most recent data that the explorer UAVs continuously collect. Once the occupancy map is updated, each UAV i then strategically generates a set of inspection waypoints $\mathcal{W}_{i,\kappa_i}$. These inspection waypoints are positioned around the occupied voxels that lie within the bounding boxes in $\mathcal{B}$, as depicted in Fig. 2e. Each waypoint $w_{l,i,\kappa_i} \in \mathcal{W}_{i,\kappa_i}$ is linked with a direction vector $\hat{n}_{l,i,\kappa_i}$, indicating the direction towards the occupied voxel that prompted the creation of the respective waypoint. This implies that the direction vector points towards the occupied voxel responsible for triggering the waypoint's generation. The generation of these inspection waypoints and direction vectors is a crucial step, setting the course for the UAV's path planning within the operational area and facilitating the calculation of the desired θ and ϕ gimbal angles to effectively inspect the infrastructure.

Following the generation of inspection waypoints, each UAV then addresses the multi-Traveling Salesman Problem (mTSP). This problem is solved cooperatively with other UAVs within LoS. The mTSP is a complex optimization problem that aims to determine the most efficient route for each UAV, given multiple destinations. The heuristic algorithm used to solve the mTSP in this work, is inspired by the proposed algorithm in [28]. It's crucial to understand that due to the communication constraints the mTSP is solved in a distributed manner. To elaborate, each UAV i generates inspection paths σ_{κ_i} considering the positions $\mathbf{p}_{j,\kappa_i}$, $j \in \mathcal{N}_{i,\kappa_i}$ and the generated inspection waypoints $\mathcal{W}_{i,\kappa_i}$. Consequently, UAV i selects the inspection path σ_{i,κ_i} specifically tailored to it, ensuring efficient and coordinated movement of the UAVs. A breakdown of the mTSP algorithm is given in Algorithm 2 and a visualization of its output is shown in Fig. 2f. It is important to note that, the inspection waypoints $\mathcal{W}_{i,\kappa_i}$, direction vectors $\hat{n}_{l,i,\kappa_i}$ and inspection paths σ_{κ_i} of each UAV agent i , are regenerated every time the UAV agent successfully visits every $w_{l,i,\kappa_i} \in \sigma_{i,\kappa_i}$ (i.e., $\kappa_i = \kappa_i + 1$).

To execute the inspection path σ_{i,κ_i} generated by the cooperative mTSP solution, each UAV i employs a strategy we referred to as Dijkstra-Receding Horizon Local Planning (D-RHLP). D-RHLP generates at each time-step k an optimal local collision-free path, based on the current state of $\mathcal{M}_{i,k}$, starting from $\mathbf{p}_{i,k}$ to the receding $w_{l,i,\kappa_i} \in \sigma_{i,\kappa_i}$ by exploiting Dijkstra's algorithm [29]. It executes a segment of the generated local path and then goes through a cycle of replanning and execution until the UAV reaches the desired waypoint. More specifically, the receding horizon technique is utilized due to the dynamic nature of the environment. Addressing inter-agent collision avoidance, all the voxels

Algorithm 2 Multiple Traveling Salesman Problem

Require: A set $\mathcal{W}_{i,\kappa_i}$ containing the inspection waypoints and positions $\mathbf{p}_{j,k}$, $j \in \{i \cup \mathcal{N}_{i,k}\}$ of the UAVs that are within LoS with UAV i , at step κ_i .

Ensure: σ_{κ_i} , s.t.: $\cup_{j=0}^{\{i \cup \mathcal{N}_{i,k}\}} \sigma_{j,\kappa_i} = \mathcal{W}_{i,\kappa_i}$.

```

1:  $v_l \leftarrow 0 \forall l \in \{1, \dots, |\mathcal{W}_{i,\kappa_i}|\}$ 
2:  $\sigma_{j,\kappa_i} \leftarrow \{\mathbf{p}_{j,k}\} \forall j \in \{i \cup \mathcal{N}_{i,k}\}$ 
3: while  $\sum_{l=1}^{|\mathcal{W}_{i,\kappa_i}|} v_l < |\mathcal{W}_{i,\kappa_i}|$  do
4:   for  $j \in \{i \cup \mathcal{N}_{i,k}\}$  do
5:      $s \leftarrow$  last waypoint in path  $\sigma_{j,\kappa_i}$ 
6:      $t \leftarrow$  closest  $w_{l,i,\kappa_i} \in \mathcal{W}_{i,\kappa_i}$  from  $s$  s.t.:  $v_l = 0$ 
7:      $\sigma_{j,\kappa_i} \leftarrow \sigma_{j,\kappa_i} \cup t$ 
8:      $v_l \leftarrow 1$ 
9:   end for
10: end while
```

within which the UAVs are currently located are marked as occupied, ensuring that no two UAVs coexist in the same voxel. The adjacency matrix, given as input to Dijkstra's algorithm, represents the graph $\mathcal{G}$. In this graph, each vertex communicates exclusively with its immediate voxel-sized hop neighbor vertices. Note that each voxel previously marked as occupied does not communicate with any other voxel. While UAV i executes its local plan, provided by D-RHLP, it rotates its camera's FoV to point towards the direction described by $\hat{n}_{l,i,\kappa_i}$, which is associated with the last waypoint in the local plan, i.e. the receding waypoint w_{l,i,κ_i} .

This comprehensive process is iteratively repeated until the conclusion of the mission time. The iterative nature of the procedure is further emphasized by the dynamic exchange of occupancy maps whenever two UAVs come within LoS. This continuous exchange of information allows for real time updates and refinements to the mission strategy. Based on the updated occupancy map resulting from this exchange, each UAV regenerates and refines its set of inspection waypoints $\mathcal{W}_{i,\kappa_i}$. It then cooperatively solves the mTSP problem with neighboring UAVs once again. The refined plan, which represents the most efficient path given the current information, is then executed. This cyclic process of exchange, update, regeneration, and execution persists until the mission's conclusion. It ensures that the UAVs are always operating based on the most recent and accurate information, thereby increasing the efficiency and success rate of the mission.

VI. EVALUATION

Extensive simulations have been conducted to evaluate the effectiveness of the proposed approach in several real-world-like infrastructure inspection scenarios, in a Gazebo-based environment.

A. Simulation Setup

The evaluation setup is an open-source scheme based on the Gazebo Simulator, RotorS, and the Robot Operating

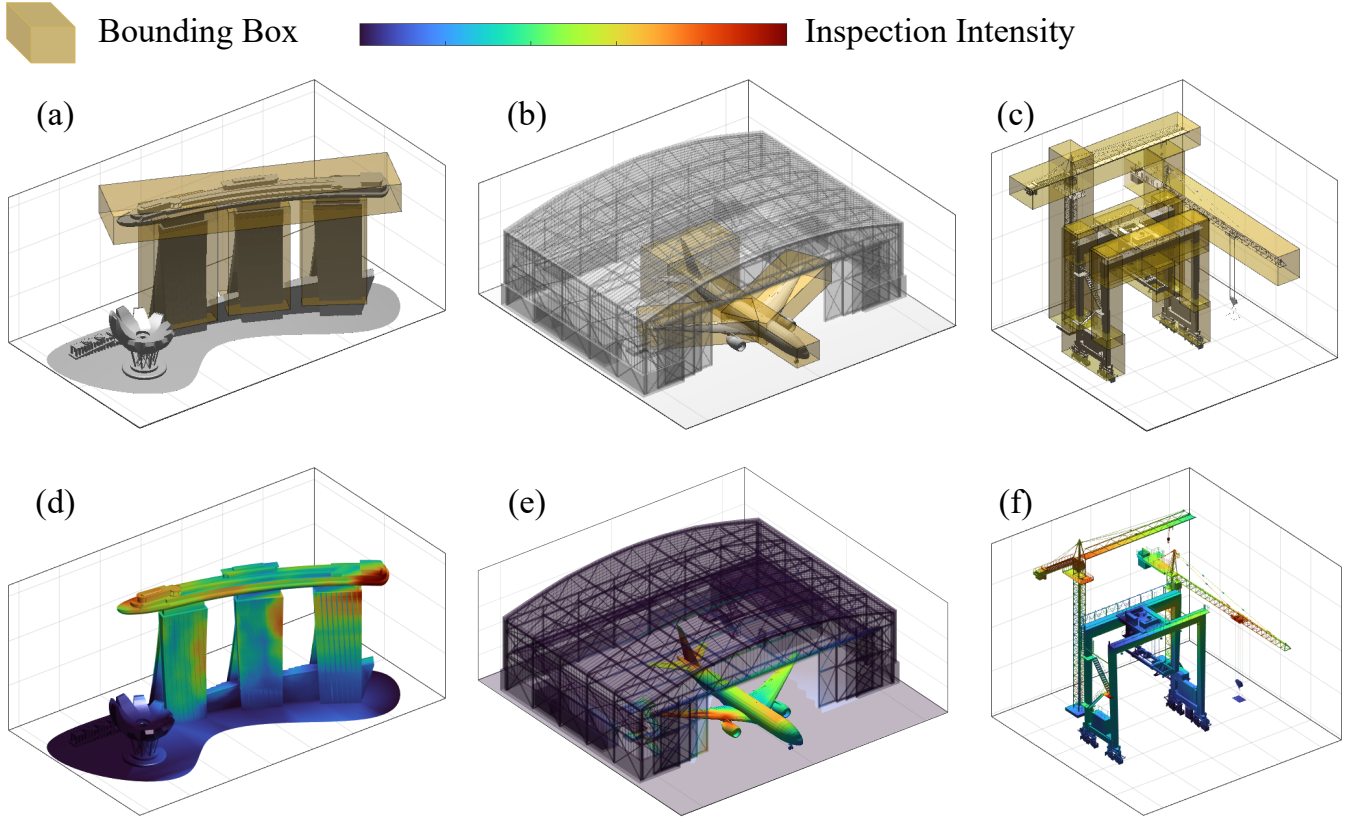


Fig. 3: (a)-(c) Illustrative examples of the 3 inspection scenarios conducted for the evaluation of the proposed approach. The bounded boxes indicate the selected area for inspection. (d)-(f) Visualization of inspection intensity achieved by the UAV fleet at each scenario, respectively, by employing the proposed CARI scheme.

TABLE I: Scenario Setups

Scenario	Infrastructure Size [m]	Interest Points $ \mathcal{Z} $	Mission Time $\mathcal{T}$ [s]	Explorer UAVs N_e	Photographer UAVs N_p
1	$140 \times 60 \times 60$	3973	500	2	3
2	$120 \times 80 \times 40$	1615	300	1	2
3	$100 \times 80 \times 80$	4935	500	2	3

System (ROS). The Gazebo simulator is responsible for rendering the three given real-world-like infrastructure scenarios including, a large building structure (i.e., a reconstruction of the Marina Bay Sands hotel in Singapore), a 3D reconstructed airplane residing inside its hangar, and a crane inspection scenario, as shown in Fig. 3a,b and c, respectively. Each 3D object in the three infrastructure scenarios is represented by a triangular mesh surface $\mathcal{S}$. In each scenario, the set of interest points $\mathcal{Z}$ to be inspected were scattered along the surface of the infrastructure. More details about each scenario including size and number of interest points are shown in Table I. As mentioned before, the environment is represented as an occupancy map which is generated by utilizing the OctoMap, a 3D mapping framework [30]. The voxel size of the occupancy map is selected as $V = 6$ [m]. The models of the two types of quadrotor UAVs are provided by the RotorS simulator which is responsible for simulating the flight dynamics and inertial sensors, ensur-

ing accurate localization. Both types are equipped with a gimbal camera sensor capable of capturing information within its limited FoV, described by $\vartheta_v = 60^\circ$ and $\vartheta_h = 80^\circ$. Moreover, the gimbal to which the camera sensor is attached, is constraint by $\theta \in [-90^\circ, 80^\circ]$ and $\phi \in [-90^\circ, 90^\circ]$. Despite their similarity in regards to the camera sensor, the explorer UAVs are double both in size and weight. Taking advantage of their greater payload capacity, explorer UAVs also carry a Velodyne VLP-16 LiDAR sensor capable of gathering environmental information up to 50 meters away with zero noise. In addition, the LiDAR is attached on a rotating servo mechanism that shifts the LiDAR's beams around the body-frame x -axis in the interval of $[-90^\circ, 90^\circ]$ completing a full cycle every 8 seconds. Note that the number of deployed explorer and photographer UAVs, inspecting the infrastructure of interest within the mission time, differs in each scenario, as shown in Table I. The exchange of information between the various components is achieved by the utilization of the ROS framework, acting as a middleware. The simulation setup as described above was provided as part of the CARIC competition by the organizing committee.

To ensure that our approach is practical and can be readily used in real-world applications, we have implemented it as a package for ROS. The implementation of our methodology was done using the Python programming language. Our

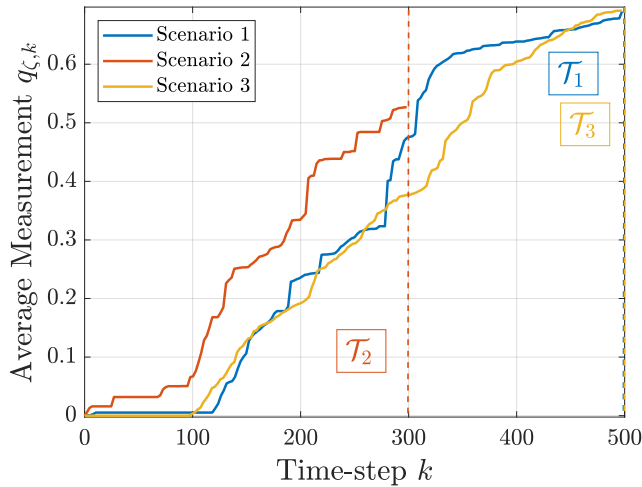


Fig. 4: Average observation quality score $q_{\zeta,k}$ achieved up to time-step k for each scenario for inspection mission durations of $T_1 = 300$, $T_2 = 500$, and $T_3 = 500$, respectively.

implementation was developed and tested on the Nvidia Jetson AGX Orin Developer Kit. The Orin System-on-Chip integrates 12 ARM Cortex-A78AE CPU cores running at 2.2 GHz, and an Ampere GPU with 2048 CUDA cores and 64 Tensor cores running at 1.3 GHz as well as 64GB of LPDDR5 RAM. This makes it an ideal platform for compute-intensive applications like ours.

B. Simulation Results

To evaluate the effectiveness and efficiency of the proposed CARI methodology, we employ two Key Performance Indicators (KPIs). The primary KPI in this study is the total inspection intensity of the infrastructure achieved by the UAV fleet. This KPI offers a more comprehensive method for assessing the distribution of inspection over the surface of the triangular mesh $\mathcal{S}$. The secondary KPI is the average score of observation q_{ζ} . This latter KPI offers insights into the quality of observations of each interest point.

For the inspection task of all three scenarios, we employ the proposed CARI scheme given in Algorithm 1. The given sets of bounding boxes $\mathcal{B}_m$, as shown in Fig. 3a-c, serve as input for each scenario $m = \{1, 2, 3\}$. This results in a set of cooperative, collision-free inspection paths for all UAV agents, as well as direction vectors for the gimballed cameras carried by the UAV agents. Based on the algorithm's output, a heat-map is generated that describes the inspection coverage intensity. In this heat-map, the most inspected area is represented with a red color, whereas the least inspected area is represented by blue. Thus, the inspection intensity achieved by employing the proposed CARI scheme for each scenario is illustrated in Fig. 3d-f. It can be observed that the surfaces encapsulated within the given bounding boxes have been successfully inspected by the fleet, without wasting resources inspecting parts of the infrastructure that lie outside of the bounding boxes. This results into a relatively uniform inspection intensity which is desired since

TABLE II: CARIC Competition Results

Trial	1	2	3	4	5
Score	4245.35	4559.80	4241.84	3933.50	3724.61

the interest points lay uniformly across the surface of the infrastructure. However, some parts exhibit higher intensity. This is mainly observed at transitional areas where the UAVs reduced speed, either for drastically changing course or for collision avoidance purposes.

Beyond the inspection intensity, the quality of the observation is another crucial factor. For instance, covering the surface of an infrastructure at a higher speed, or lower resolution, might result in complete and uniform coverage, but lead to non-informative inspection. Therefore, the average observation quality for each scenario at each time-step was also recorded and is depicted in Fig. 4. It can be observed that all three scenarios show an increasing trend in the average observation over time. This suggests that the average observation q_{ζ} improves as the time-step k progresses, demonstrating that the proposed scheme achieves high levels of inspection quality regardless of the operating environment.

The proposed CARI system, has participated in the Co-operative Aerial Robots Inspection Challenge of the 62nd IEEE Conference on Decision and Control (CDC), held in 2023 in Singapore. The effectiveness and practicality of the proposed approach was proven by taking the First place (i.e., ranking 1st), among 19 international teams. The total score for all three scenarios of the top five out of ten trials from the CARIC evaluation are provided in Table II. A video demonstration of our proposed approach can be accessed at <https://youtu.be/pJtwYsDdd2U>.

VII. CONCLUSIONS

In conclusion, this work has presented a novel CARI scheme that leverages the advanced mobility of aerial robots for efficient infrastructure inspection. The proposed scheme addressed the challenge of 3D path planning in diverse environments by employing multiple heterogeneous autonomous UAV agents for cooperative mapping and inspection under limited sensing capabilities and communication constraints. The effectiveness of our two-stage approach has been substantiated through extensive real-world-like simulations, showcasing its potential for practical 3D infrastructure inspection.

ACKNOWLEDGMENT

This work was undertaken as part of the GLIMPSE project EXCELLENCE/0421/0586 which is co-financed by the European Regional Development Fund and the Republic of Cyprus through the Research and Innovation Foundation's RESTART 2016-2020 Programme for Research, Technological Development and Innovation and 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] M. Terzi, A. Anastasiou, P. Kolios, C. Panayiotou, and T. Theodorides, "SWIFTERS: A multi-UAV platform for disaster management," in *2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, 2019, pp. 1–7.
- [2] S. Papaioannou, P. Kolios, T. Theodorides, C. G. Panayiotou, and M. M. Polycarpou, "Towards automated 3D search planning for emergency response missions," *Journal of Intelligent & Robotic Systems*, vol. 103, no. 1, p. 2, 2021.
- [3] A. Farooq, A. Anastasiou, N. Souli, C. Laoudias, P. S. Kolios, and T. Theodorides, "Uav autonomous indoor exploration and mapping for SAR missions: Reflections from the ICUAS 2022 competition," in *2022 19th International Conference on Ubiquitous Robots (UR)*, 2022, pp. 621–626.
- [4] S. Papaioannou, S. Kim, C. Laoudias, P. Kolios, S. Kim, T. Theodorides, C. Panayiotou, and M. Polycarpou, "Coordinated crlb-based control for tracking multiple first responders in 3d environments," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 1475–1484.
- [5] A. Anastasiou, R. Makrigiorgis, P. Kolios, and C. Panayiotou, "Hyperion: A robust drone-based target tracking system," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 927–933.
- [6] A. Anastasiou, S. Papaioannou, P. Kolios, and C. G. Panayiotou, "Model predictive control for multiple castaway tracking with an autonomous aerial agent," in *2023 European Control Conference (ECC)*, 2023, pp. 1–8.
- [7] S. Papaioannou, P. Kolios, T. Theodorides, 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)*, 2019, pp. 909–915.
- [8] N. Souli, R. Makrigiorgis, A. Anastasiou, A. Zacharia, P. Petrides, A. Lazanas, P. Valianti, P. Kolios, and G. Ellinas, "HorizonBlock: Implementation of an autonomous counter-drone system," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2020, pp. 398–404.
- [9] 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, 2020, pp. 7919–7926.
- [10] 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)*, 2021, pp. 2555–2562.
- [11] A. Savva, A. Zacharia, R. Makrigiorgis, A. Anastasiou, C. Kyrkou, P. Kolios, C. Panayiotou, and T. Theodorides, "ICARUS: automatic autonomous power infrastructure inspection with UAVs," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2021, pp. 918–926.
- [12] A. Zacharia, S. Papaioannou, P. Kolios, and C. Panayiotou, "Distributed control for 3D inspection using multi-UAV systems," in *2023 31st Mediterranean Conference on Control and Automation (MED)*, IEEE, 2023, pp. 164–169.
- [13] S. Papaioannou, P. Kolios, T. Theodorides, 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.
- [14] D. Lattanzi and G. Miller, "Review of robotic infrastructure inspection systems," *Journal of Infrastructure Systems*, vol. 23, no. 3, p. 04017004, 2017.
- [15] A. J. Lee, W. Song, B. Yu, D. Choi, C. Tirtawardhana, and H. Myung, "Survey of robotics technologies for civil infrastructure inspection," *Journal of Infrastructure Intelligence and Resilience*, vol. 2, no. 1, p. 100018, 2023.
- [16] W. Jing, J. Polden, W. Lin, and K. Shimada, "Sampling-based view planning for 3D visual coverage task with unmanned aerial vehicle," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 1808–1815.
- [17] S. Papaioannou, P. Kolios, T. Theodorides, C. G. Panayiotou, and M. M. Polycarpou, "Integrated guidance and gimbal control for coverage planning with visibility constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 2, pp. 1276–1291, 2023.
- [18] —, "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.
- [19] —, "UAV-based receding horizon control for 3D inspection planning," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 1121–1130.
- [20] H. Zhu, J. J. Chung, N. R. Lawrance, R. Siegwart, and J. Alonso-Mora, "Online informative path planning for active information gathering of a 3D surface," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 1488–1494.
- [21] W. Jing, D. Deng, Z. Xiao, Y. Liu, and K. Shimada, "Coverage path planning using path primitive sampling and primitive coverage graph for visual inspection," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 1472–1479.
- [22] C. Papachristos, K. Alexis, L. R. G. Carrillo, and A. Tzes, "Distributed infrastructure inspection path planning for aerial robotics subject to time constraints," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 406–412.
- [23] W. Jing, D. Deng, Y. Wu, and K. Shimada, "Multi-UAV coverage path planning using the inspection of large and complex structures," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 1480–1486.
- [24] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative coverage path planning for visual inspection," *Control Engineering Practice*, vol. 74, pp. 118–131, 2018.
- [25] G. Hardouin, J. Moras, F. Morbidi, J. Marzat, and E. M. Mouaddib, "Next-Best-View planning for surface reconstruction of large-scale 3D environments with multiple UAVs," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 1567–1574.
- [26] 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.
- [27] T.-M. Nguyen, "Determining the Velocity of a World Point in a Moving Camera Frame," https://ntu-aris.github.io/caric/docs/CARIC_motion_blur.pdf, 2023, [Accessed 07-02-2024].
- [28] A. Anastasiou, P. Kolios, C. Panayiotou, and K. Papadaki, "Swarm path planning for the deployment of drones in emergency response missions," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2020, pp. 456–465.
- [29] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [30] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, pp. 189–206, 2013.