An Online Path Planner for Bandwidth-aware Aerial Camera Networks

Mohammad Amin Rezaei, Gioacchino Manfredi, Vito Andrea Racanelli, Luca De Cicco, Saverio Mascolo
Department of Electric and Information Engineering, Politecnico di Bari, Italy
Emails: name.surname@poliba.it

***Abstract*—Unmanned Aerial Vehicle (UAV) swarms equipped with cameras are of growing interest for applications such as surveillance and object detection, transmitting real-time videos to a Ground Control Station (GCS) over the Internet. The quality of the videos received depends on both the available network bandwidth and the altitude of the drones. In this paper, a swarm of UAVs is given the task of scanning an area of interest while guaranteeing an overlap between the videos captured by each UAV for post-processing reasons, e.g. video stitching. Drones are enabled to adjust their altitude based on the available bandwidth to enhance visual quality. This work proposes a bandwidth-aware online path planner to maximize the coverage of a given area by extending the classic lawnmower path planner. The UAVs use Nonlinear Model Predictive Control (NMPC) to maximize both the total coverage area and the quality of the videos sent to the GCS. Simulations demonstrate the effectiveness of the proposed approach with different swarm configurations.**

I. Introduction and background

Unmanned Aerial Vehicles (UAVs), such as multi-rotor drones, are employed to carry out several tasks such as photogrammetry [1], search and rescue missions [2], surveillance [3], etc. UAVs gather data through sensors, such as RGB or RGB-D cameras, that are stored on the UAVs or sent in real-time through a wireless network connection to a GCS.

A *UAV swarm* refers to a coordinated group of UAVs working collaboratively to achieve a shared objective. The cooperative nature of UAV swarms allows them to handle more complex tasks, such as collaborative mapping, multi-sensor data fusion, and distributed sensing. However, a swarm requires UAVs to possess autonomous capabilities such as path planning and obstacle avoidance.

Moreover, specific control strategies must be developed to efficiently coordinate the UAVs during an autonomous task execution. Such control strategies are the main focus of the *Multi-Agent Systems* (MAS) research community and may regard swarm intelligence [4], formation control [5], consensus algorithms [6].

Some tasks such as surveillance, object detection, and patrolling often require streaming in real-time the videos captured by the onboard cameras to a GCS. In this case, swarms of drones may improve mission efficiency, both in terms of mission duration and situational awareness.

This work has been funded by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”, CUP: D93C22000910001).

This paper addresses *Aerial Camera Networks*, which consist of swarms of drones equipped with cameras tasked with covering a designated area and capturing videos [7]. The video streams are transmitted in real-time to a GCS over a communication network, typically the public Internet. The primary objective is to maximize the area covered collectively by the drones along their paths. In addition, the system must ensure that the *Field of View* (FoV) of each drone overlaps by a specified amount with its neighboring drones. This overlap is required for dynamic stitching of the videos received at the GCS [3]. However, the time-varying available bandwidth of the communication network affects the rate at which each streamed video can be compressed and transmitted to the GCS. Consequently, to improve video quality under limited bandwidth conditions, the pixel density of the captured images can be increased by lowering a drone’s altitude.

However, maintaining all drones at low altitudes would conflict with the primary objective of maximizing the covered area. Thus, the proposed approach enables drones to dynamically adjust their altitudes in response to changes in the available network bandwidth.

The proposed control strategy combines a partially distributed Nonlinear Model Predictive Control (NMPC) framework with an online path planning technique. The NMPC framework has been previously introduced in [8] for a different scenario. In this paper, we extend that framework to maximize the coverage over a desired area without a predefined path. To achieve this, we introduce a new online path planning algorithm inspired by the classic lawnmower path planner [9]. To the best of our knowledge, no existing study addresses the problem of a drone swarm maximizing coverage with dynamically varying scanned areas while transmitting overlapping video streams at the highest possible quality, given fluctuating network bandwidth.

The Swarm-based UAV (SUAV) [10], for instance, is an algorithm implemented to detect mobile objects in natural disasters. Quality metrics such as SSIM and VQM are employed to determine the visual quality of the acquired images. In this case, the impact of the variable network bandwidth is not taken into account. In [11], the authors tackle the problem of providing maximum coverage through Particle Swarm Optimization (PSO). However, the proposed technique is not focused on swarms of drones but on a certain number of drones simultaneously scanning disjoint areas.

Decentralized monitoring of an area using multiple drones is the scenario considered in [7]. One of the main novelties introduced is the ability to add or remove agents from the team while monitoring the area but without considering the network's available bandwidth.

Regarding coverage path planning techniques, the literature offers works such as [12], [9] and [13] that leverage the classic back-and-forth path and do not consider a collaborative multi-agent system framework with variable altitude of the UAVs.

In this paper, simulations show that the proposed path planner is able to provide a high percentage of coverage of a given rectangular area and a tunable overlap between successive areas scanned by the swarm.

II. System design

This section presents the system designed to achieve the goals described in Section I.

A. Preliminaries

The p norm of a vector is denoted as $\|\cdot\|_p$. A *directed graph* (digraph) is denoted with $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \dots, v_N\}$ is the set of nodes, also identified simply by its indices $i = 1, \dots, N$, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges such that an edge (v_i, v_j) denotes the information flow from node i to node j. Moreover, the set of neighbours of node v_i is defined as $\mathcal{N}_i = \{v_j \in \mathcal{V} : (v_j, v_i) \in \mathcal{E}\}$. A tree with n nodes is a connected graph having the minimum number of edges $(n-1)$. A complete graph has the maximum number of connections, i.e., $n(n-1)$ for n nodes.

B. Scenario definition

Consider a swarm of drones equipped with cameras using gimbals to accomplish tasks such as firefighting, agriculture, surveillance, etc. The cameras are pointed downwards to the ground. The area that each drone covers, and the field of view (FoV), is directly related to its altitude. To obtain a unified picture, which may be useful for improving situational awareness, the videos of the drones are stitched together by the GCS. Video stitching needs a sufficient amount of overlap between the videos, i.e. the drones fly close to each other, and a high quality of the images [3]. The overall problem is addressed by considering a leader-follower multi-agent system.

The raw video captured by the camera is compressed at an encoding bitrate that matches the available bandwidth b estimated by the *congestion control* algorithm [14] and sent to the GCS using a mobile Internet connection. It is important to stress that the visual quality is a bounded, monotonically increasing function of the encoding bitrate [15]. Thus, if the available bandwidth is low, the agents should fly closer to the ground to capture more detailed frames to compensate for the decreased visual quality. Conversely, when a large network bandwidth is available, the drone can fly higher to increase coverage. To cover the entire area of interest and to actively explore the uncovered area, online path planning at the leader has been employed. The *Path Planner* in Figure 1a, based on the leader's location s_1 and on the other agents' coordinates s_i, $i = 2, \dots, N$, where $s_i = (x_i, y_i, z_i)^T$ and N is the number of agents, computes the reference point $(\bar{x}_1, \bar{y}_1)$ that the leader has to reach at the next time-step. At this point, based on $(\bar{x}_1, \bar{y}_1)$ and $\bar{z}_1$, namely the coordinates to be tracked on the (x, y) plane and along the $z-$axis, respectively, the NMPC computes the velocity input u_1 that the leader has to set to get to the reference point. Notice that the path planner, which is inside the leader's architecture, is aware of the other agents' dynamics and location.

Therefore, each drone has two contrasting goals: i) maximize the video quality based on the available bandwidth with a good amount of overlap, and ii) maximize the coverage of the area.

Although the followers employ the same procedure to transmit the captured video to the GCS (see Figure 1b), their control inputs are computed differently, since they do not run the path planning algorithm. Based on the reference altitude $\bar{z}_i = \mathcal{H}(b_i)$ and on the planar coordinates (x_i, y_i) of all the swarm, the NMPC computes the control input u_i.

C. Problem definition

To keep the drones close to each other for a large overlap between the videos scanned by their cameras, a two-state algorithm has been implemented. In each state, a specific cost function for the NMPC has been designed.

As a first step, let us assume that the discrete-time dynamics of the evolution of the state for each drone is a simple integrator.

$$s_i(k+1) = s_i(k) + T_s u_i(k), \qquad i = 1, \dots, N \tag{1}$$

where N is the number of agents, $s_i = (x_i, y_i, z_i)^T$ is the vector of the spatial coordinates of agent i, T_s is the sampling time, and $u_i = (u_{x,i}, u_{y,i}, u_{z,i})$ is the control input for agent i. Modeling the 3-D orientation of each drone can be avoided since the camera points to the ground with a common yaw angle.

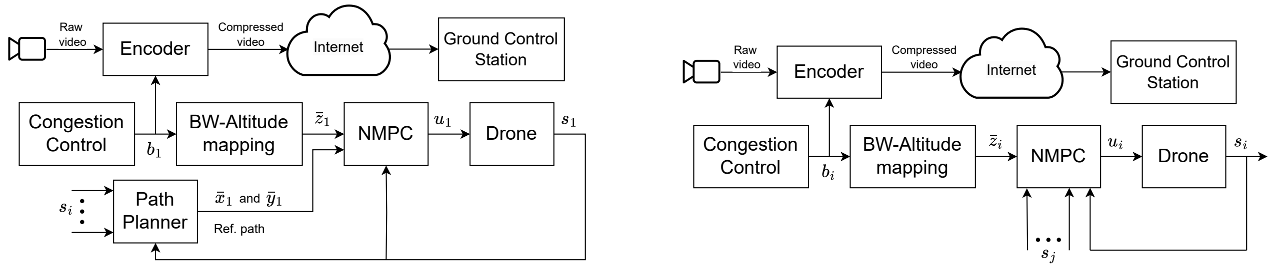
We can now provide a formulation for the NMPC:

$$\begin{aligned} \underset{u_i}{\text{Minimize}} \quad & \mathcal{J}_i && (2) \\ \text{subject to} \quad & s_i(k+1) = s_i(k) + T_s u_i(k) && (3) \\ & z_{\min} \leq z_i \leq z_{\max} && (4) \\ & \|X_i(t) - X_j(t)\|_2 \geq 2R_{\text{safe}} && (5) \\ & |u_i| \leq u_{\max} && (6) \end{aligned}$$

where $\mathcal{J}_i$ is the cost function associated with agent i whose structure depends on the state the swarm is in, as shown further on. $X_i = (x_i, y_i)^T$ is the 2-D position of the i-th agent, z_i is the drone's altitude, and R_{safe} defines the safety distance that the drones must always keep with the surroundings. Equations (3)-(6) are constraints associated with the system dynamics, altitude limits, collision avoidance, and input bounds.

Unlike the other agents, the leader, denoted by $i = 1$, tracks the reference path provided by the path planner and disregards constraint (5), leaving the task of keeping a safe distance from the leader to the other agents.

Without loss of generality, we have assumed that the area scanned by each drone is a square, whose size depends on the



(a) The leader

(b) The followers ($i = 2, \ldots, N$)

Fig. 1: Proposed control architecture

drone's altitude. The relationship between the altitude of an agent z_i and the associated side of the square L_i delimiting the scanned area is given by $L_i = 2z_i \tan(\phi/2)$, where ϕ is the angular aperture of the video camera [16].

As previously explained, to avoid possible detachments, the swarm can assume two possible states: the *Tracking State*, denoted as T, and the *Recovery State*, denoted as S.

State T - Tracking:
In the Tracking state there is a sufficient amount of overlap between the agents. The leader strives to track the planned path while the other agents try to follow the leader. Also, the leader and followers track their reference altitude based on the available bandwidth. The leader solves the optimization problem (2) with the cost function $\mathcal{J}_{1,T}$ in (7) and the constraints (3)-(6), except for the safety constraint (5). Notice that the leader's index is always assumed to be $i = 1$.

$$\mathcal{J}_{1,T} = w_p J_p + w_z J_z + w_u J_u + w_{\Delta u} J_{\Delta u}, \tag{7}$$

$$J_p = \sum_{k=1}^{N_p} \left\| X_i(t+k|t) - \bar{X}(t+k) \right\|_\infty \tag{8}$$

$$J_z = \sum_{k=1}^{N_p} \| z_i(t+k|t) - \bar{z}_i(t+k) \|_2 \tag{9}$$

$$J_u = \sum_{k=1}^{N_c} \| u_i(t+k-1) \|_2 \tag{10}$$

$$J_{\Delta u} = \sum_{k=1}^{N_c} \| \Delta u_i(t+k-1) \|_2 \tag{11}$$

where J_p is the cost associated with the path tracking, $X_1(t)$ is the 2-D position of the leader at time t, N_p is the prediction horizon, N_c is the control horizon for the NMPC controller, and $\bar{X}(t)$ denotes the 2-D coordinates of the reference point provided by the path planner at time t. J_z is responsible for altitude tracking, with $z_1(t)$ as the leader's altitude and $\bar{z}_1(t)$ as the leader's reference altitude at time t. J_u is the cost for the control effort, and $J_{\Delta u}$ is the cost for the variation in the control input.

Concerning the followers, they solve the optimization problem (2)-(6) where the cost function $\mathcal{J}_{i,T}, i = 2, \ldots, N$ for the state T is defined as follows.

$$\mathcal{J}_{i,T} = w_l J_l + w_z J_z + w_u J_u + w_{\Delta u} J_{\Delta u}, \tag{12}$$

$$J_l = \sum_{k=1}^{N_p} \left(\| X_i(t+k|t) - X_1(t+k) \|_\infty - r(R_i + R_1) \right)^2 \tag{13}$$

where J_l encourages agent i to follow the leader with $R_i = L_i/2$. To obtain an overlap, $0 < r < 1$ makes the agents strive to get to a distance less than $R_i + R_1$ from the leader.

State S - Recovery:
If the risk of detachment exceeds a certain threshold, the agents switch to the Recovery state, where the leader stops tracking the reference path and waits for the followers to gather up.

As a consequence, the cost function for the leader loses the term J_p related to the path tracking, thus changing as follows.

$$\mathcal{J}_{1,S} = w_z J_z + w_u J_u + w_{\Delta u} J_{\Delta u} \tag{14}$$

In this state, the cost function for the followers does not penalize the control input to allow the agents to use maximum control effort to recover as soon as possible. The cost function for the agents, $i = 2, \ldots, N$, is defined as follows:

$$\mathcal{J}_{i,S} = w_l J_l + w_z J_z + w_f J_f \tag{15}$$

$$\begin{aligned} J_f = \sum_{j \in \mathcal{N}_i} \Big(\sum_{k=1}^{N_p} \big(\| X_i(t+k|t) - X_j(t+k) \|_\infty \\ - r(R_i + R_j)^2 \big) \end{aligned} \tag{16}$$

where J_f is the term that holds the agents together to retrieve a certain amount of overlap from the scanned areas. For J_f, considering a complete directed graph would cause issues with the altitude tracking, or the leader following. For this reason, we consider a tree with N nodes and $N-1$ edges where agent 2 receives from the leader and sends and receives from agent 3, agent 3 sends and receives from agents 2 and 4, and the final agent sends and receives from its previous neighbor. For this new graph, and therefore for J_f, let $\mathcal{N}_i = \{ j \in \mathcal{V} : (j, i) \in \mathcal{E} \}$.

The overlap of the scanned areas between the agents is achieved through the term $r(R_i + R_j)$ in J_f, where $0 < r < 1$ determines the desired amount of overlap. Therefore, r close to 1 implies a relatively small percentage of overlap while decreasing r pushes the agents closer to each other.

Notice that w_l,w_p,w_z,w_u,$w_{\Delta u}$, and w_f are tunable optimization weights. A more detailed formulation and parameter explanation are provided in [8] due to space constraints.

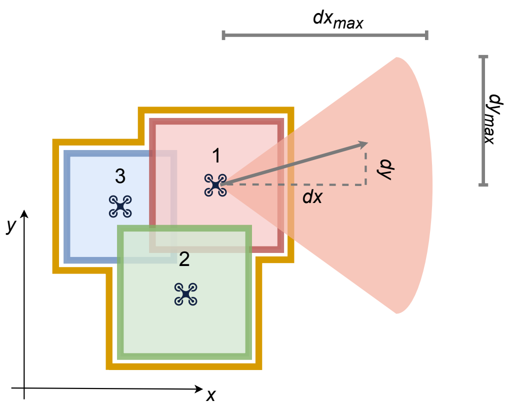


Fig. 2: Formation scheme

1) Bandwidth-altitude mapping $\mathcal{H}$: According to the rationale described in Section II-B, any monotonically increasing function $\mathcal{H} : b \mapsto \bar{z}$ mapping the available bandwidth b to $[z_{\min}, z_{\max}]$ can be used to compute the reference height $\bar{z}_i$ for agent i. In this paper, without loss of generality, we use the following monotonically increasing function.

$$\bar{z}_i(b) = \mathcal{H}(b) = \begin{cases} z_{\min} & b < b_{\min} \\ S(b - b_{\max}) + z_{\max} & b_{\min} \le b \le b_{\max} \\ z_{\max} & b > b_{\max} \end{cases} \tag{17}$$

where S is defined as $S = (z_{\max} - z_{\min})/(b_{\max} - b_{\min})$.

III. Path planning technique

This section presents the proposed online path-planning technique used to achieve coverage of a fixed rectangular area known to the leader of the swarm. Figure 2 illustrates a scenario with three UAVs, where UAV 1 acts as the leader. The total scanning area of the UAVs at each time-step is not constant because it changes in size and shape.

The proposed path planner is run by the leader and works as follows. Given the initial configuration and scanned area, each UAV communicates its position to the leader. At this point, the leader computes the union of all the scanned areas, denoted *union shape* (orange shape in Figure 2), and runs the path planner, which, based on such an area and the map of the environment, computes the next reference point of the path. The leader then runs the NMPC to compute the reference velocities required to reach that point while the other UAVs run their own NMPC to follow the leader and stay close together. At the next time-step, all the UAVs communicate their new position to the leader and the procedure iterates.

It is important to point out that, in a realistic scenario, if a UAV moves from one point to another close point and such a movement is performed in a reasonably short amount of time, the available bandwidth does not change abruptly.

A widespread approach to achieve coverage when the scanned area is constant is to follow a lawnmower path. Therefore, inspired by this approach, we have designed an adaptive path planner, i.e. a path planner dependent on the variable scanned area. At this point, locating without loss of generality the initial formation in a top left area of the rectangular map, the general approach is to proceed along the $x-$axis on the right while also adjusting the path along the $y-$axis to improve coverage and avoid blind spots, i.e. uncovered areas, when the union shape changes. Once the right border is reached, the swarm moves downwards by a certain number of steps. Then, the swarm goes from right to left, and so on.

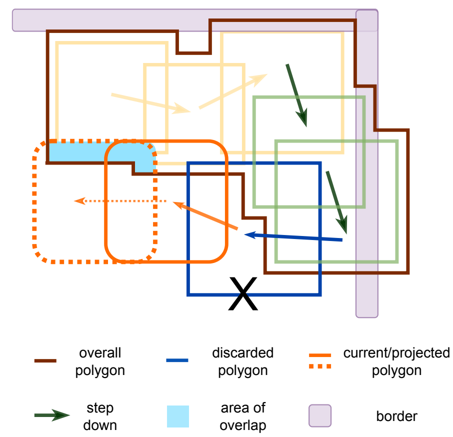


Fig. 3: Polygon computation and turning phase

Concerning the coordinate on the $x-$axis of the next reference point of the path when going from left to right (right to left), the general lawnmower approach is first evaluated. In other words, the current union shape is projected straight onward along the $x-$axis on the right (on the left) with zero displacements on the $y-$axis. The step d_x along the $x-$axis is computed as shown in Algorithm 1 (Line 1).

Algorithm 1 Computation of d_x and d_y

1: $d_x \leftarrow (1 - 2 \cdot \mathrm{mod}(flip, 2)) \cdot dx_{\max}$
2: Compute p
3: $d_y \leftarrow 0$
4: **if** $p < L_{\text{thr}}$ **then**
5: $\quad d_y \leftarrow dy_{\max} \cdot (1 - \frac{p}{L_{\text{thr}}})$
6: **else if** $p \geq H_{\text{thr}}$ **then**
7: $\quad d_y \leftarrow -dy_{\max} \cdot \frac{p}{100}$
8: **end if**

$flip$ is an indicator of the direction and is used as follows: even if the swarm goes from left to right, otherwise the UAVs move from right to left. $dx_{\max}$ is the maximum reference distance that the path planner can impose in a time-step (see Figure 2). Notice that $dx_{\max}$ has to be chosen according to the maximum power of the drones.

The step in the y direction is computed on the basis of the percentage of overlap with the upper or lower previously scanned area or with the border of the map. We want to keep such a percentage within a tunable interval for two reasons: the lower bound L_{thr} guarantees full coverage of the area whereas the upper bound H_{thr} prevents the swarm from going

TABLE I: Values of the weights in the cost functions of the NMPCs

	w_l	w_p	w_f	w_z	w_u	$w_{\Delta u}$
$\mathcal{J}_{1,T}$	-	15	-	2	0.001	0.001
$\mathcal{J}_{i,T}$	1.5	-	-	3	0.001	0.001
$\mathcal{J}_{1,S}$	-	-	-	2	0.001	0.001
$\mathcal{J}_{i,S}$	3	-	1	3	-	-

too much out of the unexplored area. In the meantime, the overall polygon given by the union of all the union shapes at each time-step is computed and stored. When the swarm moves on in the map, the path planner evaluates the percentage of overlap of the current union shape with the overall polygon. It is important to point out that, in order to compute the overlap with the already explored area, the recently computed union shapes that overlap on the right or left with the current one have to be discarded from the computation of the overall polygon.

Figure 3 clarifies this step of the proposed path-planning algorithm. For the sake of simplicity, in the example shown, we assume that the union shape at each time-step is a square but the algorithm works for polygons with general shape.

The overall polygon (brown shape) is computed by discarding the recently computed union shapes (blue square) that overlap with the current union shape (continuous orange square). This way, the area of overlap (light blue area in the figure) between the projection of the current polygon (dashed orange square) and the overall polygon is not affected by a possible lateral overlap with recent union shapes (blue square). Notice that this is not the case illustrated in the figure, but since the step forward can be small, it is likely to occur.

The step along the y direction has to depend on the percentage of overlap related to the overall polygon (light blue area in Figure 3) or to the border of the map. Such a step is computed according to Algorithm 1 (Lines 2-8) where p is the percentage of overlap. dy_{max} is the maximum reference distance that the path planner can impose (see Figure 2).

When the swarm reaches one of the two vertical borders, it goes vertically down for a number of steps equal to $\lfloor \gamma \cdot V \rfloor$, where V is the maximum length of the current union shape along the $y-$axis and $\gamma \in \mathbb{N}$ is a tunable parameter. It has been observed that a good choice for such a parameter is a value in the interval $[10, 20]$. Finally, $d_y = -dy_{\text{max}}$ and d_x is proportional to the overlap with the lateral boundary.

IV. Simulations

In this section, simulation results are presented and discussed for the case of $N \in \{3, 4, 6, 8, 10\}$, where N is the number of agents. As previously stated, the given area to cover is assumed to be an obstacle-free rectangle. Finally, we have set the values for the weights in the cost functions previously introduced as in Table I. A study on hyperparameter tuning for this problem will be object of future work.

Concerning the NMPC, $N_c = 2$, $N_p = 5$, $R_{\text{safe}} = 2$, $r = 0.4$ and $T_s = 0.1s$. Additionally, we have imposed an upper and lower bound for the drones' altitude, in particular $z_{\text{min}} = 6\ m$ and $z_{\text{max}} = 15\ m$. The velocity of the leader has been intentionally set lower than that of the follower agents to make sure that they can keep up with the leader. Therefore, $u_{x,\text{max}} = u_{y,\text{max}} = 0.7\,m/s$ in (6) for the leader whereas $u_{x,\text{max}} = u_{y,\text{max}} = 1\,m/s$ for the followers. Finally, $u_{z,\text{max}}$ has been set to $0.1\ m/s$ for all agents to avoid abrupt changes in altitude. Regarding the path planner, we have imposed a desired overlap in the range 20-60% and a maximum reference distance as $dx_{\text{max}} = 0.1\ m$ and $dy_{\text{max}} = 0.1\ m$. γ has been set to 15.

The available network bandwidth has been modeled as a uniform random distribution over the area to be scanned and has been assumed to be constant in time and altitude. Since, to the best of our knowledge, no dataset providing bandwidth measurements as a function of the geographical position of a drone is available, we have chosen a minimum value of 500 kbit/s and a maximum value of 5000 kbit/s.

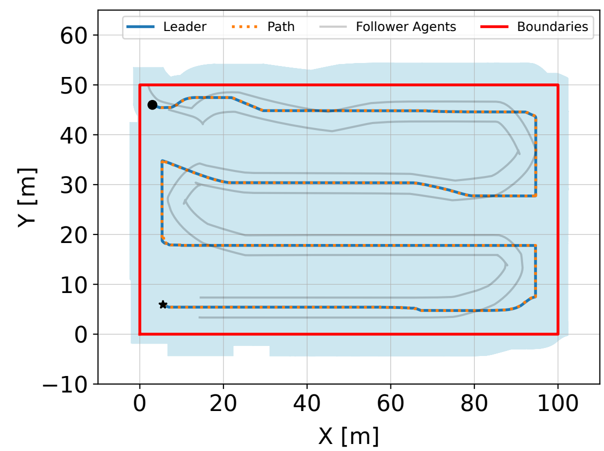


Fig. 4: Drones' trajectories and area covered for $N = 3$

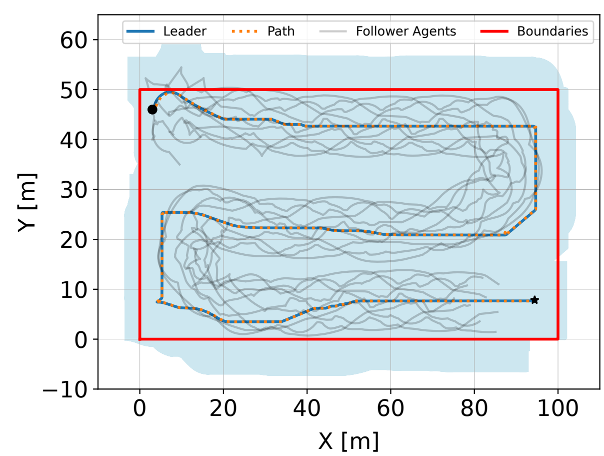


Fig. 5: Drones' trajectories and area covered for $N = 10$

Let us start by considering the case of $N = 3$ agents. Figure 4 shows the reference path provided by the path planner (dashed orange path in the figure), the drones' trajectories projected on the (x, y) plane (blue and grey paths), and the overall area covered by their cameras (light blue area). The initial position of the leader (round marker) and of the other agents is in the top left corner of the given area. Looking at the drones' trajectories, one can observe that while the leader

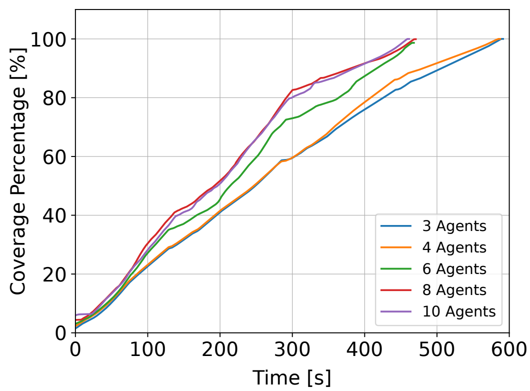


Fig. 6: Percentage trend of the area covered over time

tracks the reference path until the end of the mission (star marker), the followers strive to catch up with the leader and stick together. It is also possible to inspect the adaptive nature of the proposed path planner, whose output trajectory tries to comply with the variable area scanned by each drone.

Figure 5 reports the case of $N = 10$ agents, where it is clear that, since the swarm can cover a considerably wider area around the reference path, it takes fewer turns to the path planner to complete the task. Intermediate results have been observed for the other numbers of agents.

An interesting observation can be drawn from Figure 6, which shows how the percentage of the area covered by the swarm grows over time for each number of agents N considered for the simulations. The general trend coming out is a decreasing time needed to complete the mission and a steeper curve as N increases.

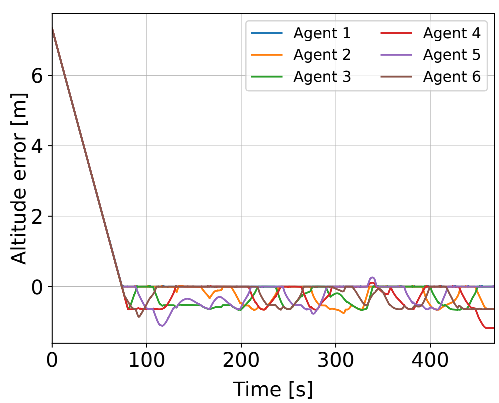


Fig. 7: Tracking error of the altitude for $N = 6$

Along the z-axis, the drones have to track a reference altitude based on their location and available network bandwidth as computed in (17). Figure 7 shows the dynamics of the tracking error of the altitude for the case of $N = 6$ agents. All the agents start from the same initial altitude and then track their own reference altitude with satisfactory precision after a transient. At the beginning there is a substantial error because the initial altitude of the agents is lower that the reference. Similar considerations can be made for the other cases.

Overall, the proposed path planner strives to provide full coverage and to adapt to the changing area scanned by drones. In addition, it is scalable with respect to the number of drones since it runs only on the leader.

V. Conclusions

We have considered a swarm of UAVs equipped with cameras that has to cover a given area while keeping close to each other to guarantee a certain overlap between the single videos captured and sent to a GCS. Moreover, the altitude of each UAV changes while moving according to the available network bandwidth. To this end, a leader-follower approach leveraging NMPC controllers at each UAV and an adaptive path planner have been designed. Future research is focused on improving the path planner in terms of obstacle avoidance and path optimization, thus considering different profiles other than the back-and-forth motion on which it is currently based.

References

[1] T. M. Cabreira, C. D. Franco, P. R. Ferreira, and G. C. Buttazzo, "Energy-Aware Spiral Coverage Path Planning for UAV Photogrammetric Applications," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3662–3668, 2018.

[2] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Computer Communications*, vol. 156, pp. 1–10, 2020.

[3] X. Meng, W. Wang, and B. Leong, "Skystitch: A cooperative multi-uav-based real-time video surveillance system with stitching," in *Proc. ACM Multimedia*, 2015, pp. 261–270.

[4] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 10, pp. 1627–1643, 2021.

[5] H.-S. Ahn, *Formation control*. Springer, 2020.

[6] M. A. Rezaei, P. Bagheri, and F. Hashemzadeh, "Predictive consensus tracking of multi-agent systems in the presence of byzantine agents and connection loss of reference signals," *Optimal Control Applications and Methods*, 2023.

[7] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.

[8] M. A. Rezaei, G. Manfredi, V. A. Racanelli, L. De Cicco, and S. Mascolo, "Decentralized control of uav swarms for bandwidth-aware video surveillance using nmpc," in *Proc. International Conference on Unmanned Aircraft Systems (ICUAS)*, 2024, pp. 947–954.

[9] M. A. Luna, M. S. Ale Isaac, A. R. Ragab, P. Campoy, P. Flores Peña, and M. Molina, "Fast multi-uav path planning for optimal area coverage in aerial sensing applications," *Sensors*, vol. 22, no. 6, p. 2297, 2022.

[10] I. Medeiros, A. Boukerche, and E. Cerqueira, "Swarm-based and energy-aware unmanned aerial vehicle system for video delivery of mobile objects," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 766–779, 2021.

[11] H. S. Munawar, A. W. Hammad, and S. T. Waller, "Disaster Region Coverage Using Drones: Maximum Area Coverage and Minimum Resource Utilisation," *Drones*, vol. 6, no. 4, p. 96, 2022.

[12] 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. 27 783–27 803, 2015.

[13] U. Choi and S. Lee, "Bandwidth-aware coverage path planning for swarm of uavs with aerial base station," in *Proc. of Int. Conference on Unmanned Aircraft Systems (ICUAS)*, 2023, pp. 360–365.

[14] G. Carlucci, L. De Cicco, S. Holmer, and S. Mascolo, "Congestion control for web real-time communication," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2629–2642, 2017.

[15] G. Manfredi, L. De Cicco, and S. Mascolo, "Optimal QoE-Fair Resource Allocation in Multipath Video Delivery Network," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 3487–3500, 2022.

[16] H. Huang, A. V. Savkin, and C. Huang, "Decentralized autonomous navigation of a uav network for road traffic monitoring," *IEEE Tran. on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2558–2564, 2021.