# Virtual Leader based Trajectory Generation of UAV Formation for Visual Area Coverage

Hilton Tnunay, Kaouther Moussa, Ahmad Hably, Nicolas Marchand

# Virtual Leader based Trajectory Generation of UAV Formation for Visual Area Coverage

Hilton Tnunay[1], Kaouther Moussa[2], Ahmad Hably[1], Nicolas Marchand[1]

[1]Univ. Grenoble Alpes, Grenoble INP, GIPSA-Lab, CNRS, France
{ishak.tnunay, ahmad.hably, nicolas.marchand}@gipsa-lab.fr

[2]Univ. Polytechnique Hauts-de-France, LAMIH, CNRS, UMR 8201, F-59313 Valenciennes, France
kaouther.moussa@uphf.fr

***Abstract*—This paper proposes a trajectory generation strategy of quadcopter formation for area surveillance and inspection using multiple cameras. The proposed technique exploits a minimal virtual-leader based network topology to generate the executable trajectories where generating trajectories for each agent is no longer necessary. A coverage path planning algorithm is employed to generate a set of planar waypoints, which are then optimised to yield a minimum-jerk time-parametrised trajectory with dynamical constraints of the virtual leader to follow. In order to establish a formation pattern, the desired distance among the agents is then computed based on the operation mode, size of the area to cover, and the number of deployed UAVs. Numerical simulations demonstrate the effectiveness of the proposed method.**

***Index Terms*—Virtual-leader Formation, Visual Coverage Path Planning, Trajectory Generation, Quadcopter**

## I. Introduction

In the few decades, visual area coverage with static or dynamic cameras has found applications in populated and natural environments. They include gathering information of an area, including population activity, agricultural information, area security, vehicle traffic, natural phenomenon, post-disaster victim searching, and terrain information [1], [2].

The planar area coverage with a static camera can be achieved by equipping the camera with actuating mechanism for adjusting its direction [3], [4] and [5]. The work in [3] considered static camera implementation with both circular and rectangular field-of-view (FOV). Coordination of multiple Pan-Tilt-Zoom (PTZ) cameras was addressed by accommodating the sensing quality of the camera and the configuration of the network [5]. However, it is noticeable that static camera-based implementations still have limited flexibility and non-adjustable placement.

Attaching cameras on Unmanned Aerial Vehicles (UAVs) might significantly improve the flexibility and efficiency of the coverage system. In contrast to the static camera, the controllability of this mechanism is achievable by adjusting the position of UAVs. In [6], visual coverage control with downward-facing cameras on multiple quadcopters has been addressed for cameras with both circular and rectangular FOV. In addition to the static camera scenario, the work in [3] has also investigated the visual coverage control via mixed static and dynamic cameras. Collaborative visual area coverage to reduce the overlapping between the sensing regions has been reported in [7], [8], [9] and [1]. Implementation of multi-camera coverage with flying UAVs and PTZ cameras to improve the degree of freedom of the camera was conducted in [10]. However, these existing works have not considered the scenario where the coverage area is significantly larger than the whole-team camera coverage. These existing algorithms might require the UAVs to climb to higher altitudes and consequently deteriorate the image quality of the corresponding area.

In a large area coverage problem, the deployed flying camera should dynamically cover the whole region without sacrificing the expected image quality. Coverage path planning problem can be employed where the camera carrier executes the boustrophedon motion throughout the region [11], [12], [13], [14], [15]. In [11], the coverage problem with a minimal-sum-of-altitudes decomposition approach via multi-line sweeps and dynamic programming was considered to obtain a cheaper coverage path. In [12], a boustrophedon motion of a mobile robot with the battery voltage constraints was studied to generate trajectories with minimal abrupt turns. Energy consumption and resolution constraints in coverage path planning have also been addressed utilising the size of partition [14], [15]. In [13], a multi-UAV minimum-time coverage path planning has been addressed by assigning UAVs to several decomposed regions and employing the vehicle routing problem (VRP) to compute the fastest route. However, the problem of deploying UAV formation executing the coverage path planning has not been investigated.

This paper proposes a strategy for the mission manager of a UAV formation to provide commands to the UAVs to cover a large area visually. This method considers a virtual leader topology in the formation mechanism and considers an interrupting scenario where some UAVs are assigned to acquire higher image quality of specific regions while the rests compensate for the left area gaps. The proposed algorithm generates minimum-jerk trajectories of multiple flying cameras in a large-area coverage problem to reduce energy consumptions and offers adjustable formation distances to accommodate dynamical changes of the expected image quality. In contrast to the previous works, our method considers the UAVs to move in a formation following the boustrophedon motion and adjust their altitudes to satisfy the expected image quality. The outputs of our algorithm are the time-parametrised minimum-jerk trajectories for the virtual leader and the desired distances among the agents.

The remainder of this paper is organised as follows. Section 2 briefly reviews the notions of graph theory and camera settings

This work was supported by the TAMOS (TActical Multi-Objective Swarming UAVs) project.

related to the proposed strategy. Then, section 3 states the main problem addressed in this paper. Afterwards, the main algorithm for trajectory generation of the quadcopter formation is presented in Section 4. Finally, some simulations validating the proposed algorithm is presented in Section 5, followed by concluding remarks in Section 6.

## II. Preliminaries

### A. Graph Theory

A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a collection of vertices $\mathcal{V} = \{p_1, p_2, \ldots, p_n\}$ connected by a collection of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. If there exists an edge $(p_i, p_j) \in \mathcal{E}$, vertex $p_i$ is able to receive information from vertex $j$. If, for $(p_i, p_j) \in \mathcal{E}$, there exists $(p_j, p_i) \in \mathcal{E}$, the graph is called undirected. We refer $p_j \in N_i \subset \mathcal{V}$, for $p_j \neq p_i$, to the neighbour of vertex $p_i$ if $(p_i, p_j) \in \mathcal{E}$.

### B. Camera Setup

For convenience, Fig. 1 is provided to illustrate the camera setup. Camera resolution is the total number of the pixel of the captured image; camera angles $\alpha$ and $\beta$ are the maximum angle a camera can cover in a static position and orientation, where where $\alpha = 2\angle rpp'$ and $\beta = 2\angle qpp'$; while aspect ratio is the ratio between image length and width. The length of a coverage area

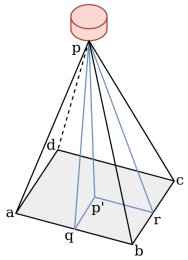



Fig. 1: Camera setup.

is the distance between $a$ and $b$, while the width is the distance between $b$ and $c$. The aspect ratio, $\rho$, is the ratio between the length and the width of the coverage area. For a given camera altitude $h$, we may express the coverage length and coverage width, respectively, as $l = 2h\tan(\alpha/2)$, and $w = 2h\tan(\beta/2)$, where their relationship can be expressed using the aspect ratio as $l = \rho w$.

## III. Mission Formulation

In this section, we formulate the class of area coverage missions in an environment using a formation of flying cameras.

Consider a surveillance area of a two-dimensional space $\mathcal{Q}_S$. Inside the surveillance area, there are several inspection areas, denoted by $\mathcal{Q}_I$, whose size is significantly smaller than the surveillance area. In the area, we deploy $N$ quadcopters whose position relative to the origin on the ground and heading are denoted by $p_i = [p_{x,i}\ p_{y,i}\ p_{z,i}]^\top$ and $\psi_i$, respectively, for $i = 1, 2, \ldots, N$. Each quadcopter carries a camera – with identical and non-adjustable camera resolution, camera angle, and aspect ratio – parallel to the ground where the camera's position and heading align with the quadcopter's. It is achievable by mounting the camera on a gimbal stabiliser. The camera coverage area on the ground has the length and width of $l_i$ and $w_i$, respectively.

The operation scenarios of the system are twofold: surveillance and inspection. Given an expected image coverage area, the surveillance is an operation of a formation of UAVs aiming to cover all points in the surveillance area while maintaining the formation pattern. The inspection is an operation where several agents are assigned to fly through and capture a higher image quality of an inspection area while the remaining agents adjust their image coverage size such that there is no coverage gap. After the inspection area is fully covered, they return to the surveillance operation.

This paper aims to design a mission manager that provides a time-parametrised trajectory of the formation and desired distances among the agents to execute the surveillance and inspection operations. The trajectory and desired distance are then provided to the quadcopter's guidance control for execution.

## IV. Trajectory Generation

In this section, we propose a computationally-efficient trajectory planning strategy for a formation of UAVs to execute the surveillance and inspection operations. The architecture of the proposed strategy to execute the operations is presented in Fig. 2

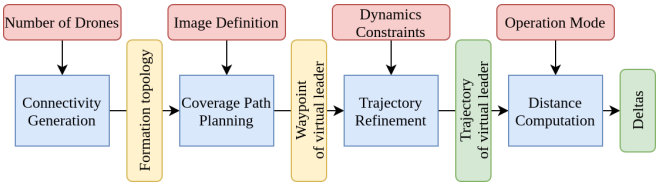



Fig. 2: Computation architecture of mission manager. Red, blue, yellow, and green blocks represent the user input, computation block, local output and main output of the mission manager, respectively.

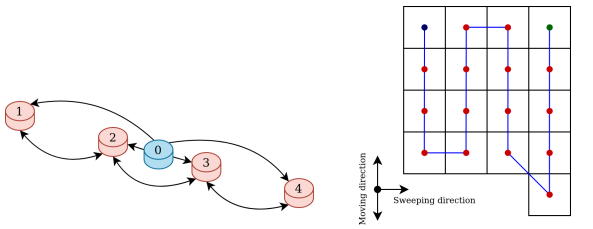
### A. Connectivity Generation

For all operation scenarios, we propose a virtual-leader based formation of UAVs to execute the mission. Literature related to virtual leader topology can be found in, for example, [16]. This approach can be represented via a graph $\mathcal{G}_f = (\mathcal{V}_f, \mathcal{E}_f)$ where the vertices $\mathcal{V}_f = \{0, 1, \ldots, N\}$ contains all the agents, both the virtual and physical; and the edges $\mathcal{E}_f \subseteq \mathcal{V}_f \times \mathcal{V}_f$ denotes their connectivity. The virtual agent, indexed $i = 0$, is assigned as the formation leader; and the physical quadcopters are assigned as the followers indexed $i = 1, 2, \ldots, N$. By utilising the virtual-leader based formation, it is sufficient to generate a trajectory for the leader and the desired distance among the agents.

In the problem of driving UAV formation to cover the surveillance area, the geometrical shape of the formation contributes to the efficiency of the execution. The formation shape is chosen where gaps among the coverage areas are not allowed, and each point in the surveillance area should not be observed by more than one camera simultaneously. Hence, we assign the UAVs to follow a line formation perpendicular to the direction of the movement. Furthermore, by assuming that the camera angles

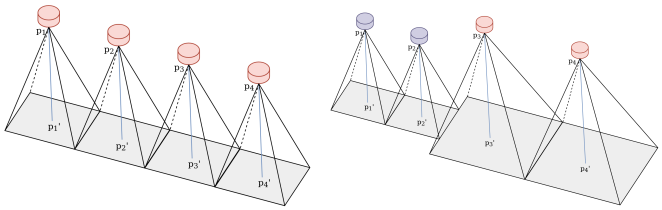
are non-adjustable, the only way to control the image quality is by adjusting the altitude of the UAVs. However, adjusting the altitude requires the UAV to adjust its $x$ and $y$ positions to prevent coverage gaps. Therefore, to maintain a line formation both in the surveillance and inspection operations, we also adjust the distance between the agents. If the agents are in surveillance, the distance among them is uniform; if they are in inspection, the assigned ones have a smaller distance than the unassigned ones since the unassigned ones compensate the coverage gap appearing as the assigned ones adjust their altitudes.

The illustrations about the formation setup consisting of one virtual leader and four followers are provided in Figs. 3a and 4. In Fig. 3a, the virtual leader provides information utilisable by all physical agents, but not vice versa. Fig. 4 illustrates the formation among the agents in surveillance and inspection operations. Fig. 4a illustrates that the distance between $p_i$ and $p_{i+1}$, for $i = 1, 2, 3$, are similar, while Fig. 4b shows that the distance among the assigned agents, $p_1$ and $p_2$, is smaller than the unassigned agents, $p_3$ and $p_4$.




(a) Network connectivity. (b) Sweeping mechanism.

Fig. 3: Network setup and sweeping mechanism.



(a) Surveillance operation. (b) Inspection operation.

Fig. 4: Network topology for the surveillance and inspection scenarios. The assigned agents are represented by the purple cylinders and the unassigned agents by the red ones.

### B. Coverage Path Planning

The proposed coverage path planning aims to generate a set of waypoints for the virtual leader throughout the surveillance area such that the formation visits each point at most once, that is, a boustrophedon motion. The algorithm requires the vertices of the area boundaries and the expected image quality before execution, where the latter determines the grid resolution in the area partitioning process of the surveillance area. The method for solving this problem is based on [11], [12], [13], [14], and [15].

In the coverage path planner, the area of interest is modelled via a polygon described by an ordered set of polygonal vertices $\mathcal{P} = \{v_1^p, v_2^p, \ldots, v_m^p\}$, for $m$ denotes the number of the polygon vertices and $v_1^p \neq v_m^p$. Each vertex $v_k^p$ is characterised by a pair of coordinates $(v_{x,k}^p, v_{y,k}^p, v_{z,k}^p)$. In this workr, we will only consider $v_{z,k}^p = 0.0$, that is, the area coverage is located on the ground.

Given the polygonal vertices $\mathcal{P}$, the algorithm starts with identifying the start point and the sweeping direction of the motion. It determines the start point of the UAVs, which reduces the number of turns and accordingly lowers the energy consumption of the execution. The optimal direction of the motion is the direction leading to the largest width of the polygon. First, the algorithm calculates the vector difference between $v_k^p$ and $v_{k+1}^p$ via $\delta_{k,k+1}^{vp} = v_{k+1}^p - v_k^p$, for $k = 1, 2, \ldots, (m-1)$, and the distance among them, $d_{k,k+1}^{vp} = \|\delta_{k,k+1}^{vp}\|$. Secondly, the algorithm calculates the highest distance value $d_{\max}^{vp} = \max(d_{1,2}^{vp}, d_{2,3}^{vp}, \ldots, d_{m-1,m}^{vp})$. Finally, the sweeping direction is then equal to the vector difference which corresponds to the highest distance value, that is, $\gamma_{\text{sweep}} = \arg\max d_{\max}^{vp}$; while the start position, $v^{ps}$, is at $v_k^p$ whose $k$ corresponds to vector with highest distance value.

Afterwards, the algorithm transforms the polygon vertices into grid coordinates and determines the endpoint by selecting the farthest grid in the direction of the sweeping motion. The grid size corresponding to the grid resolution of the surveillance area is obtained based on the expected image quality. The total length of area covered by the drone formation formulated as $r^g = \sum_{i=1}^{m} l_i^{\text{d}}$ where $l_i^{\text{d}}$ where $l_i$ is the length of the coverage area of agent $i$ obtained according to the relationship with the camera angle formula. Finally, the algorithm determines the endpoint by selecting the farthest grid in the direction of the sweeping motion.

The grid map inside the polygon is then transformed into a graph $\mathcal{G}_g = (\mathcal{V}_g, \mathcal{E}_g)$, where $\mathcal{V}_g$ is the set containing all grid vertices inside the polygon and $\mathcal{E}_g$ is the relationship among those vertices. The nearest vertices of vertex $g_i$ is all vertices $g_j \neq g_i$ that directly connects with $g_i$, denoted by $N_i$. Starting from grid $g_i$, the cost of point $g_i$ choosing point $g_j \in N_i$ as its successor is proportional to the sine of the angle between $(g_j - g_i)$ and the moving direction $\gamma_{\text{move}}$, that is, $\sin\theta = \|\gamma_{\text{move}} \times (g_j - g_i)\| / \|\gamma_{\text{move}}\| \|(g_j - g_i)\|$. Once the neighbour selection process is completed and the graph is fully generated, we may enlist the successors of the vertices starting from the vertex $i = 1$ at the start point to the endpoint. The set containing the sorted vertices from the start to endpoints is denoted by $\mathcal{W} = \{w_1, w_2, \ldots, w_M\}$, for $M = |\mathcal{V}_g|$ being the number of vertices in the grid map. The set will be referred to as the waypoints of the trajectories. The illustration of this method is provided in Fig. 3b.

### C. Trajectory Refinement

Polynomial trajectories can be considered natural for various aerial robots. The trajectories can easily be obtained via the minimisation of a performance index of the path derivatives. In this mission, the generated waypoints need refinement to yield a smooth time-parametrised trajectory, not violating the limitation of the quadcopter dynamics. Based on the works reported in [17], [18] and [19] and [20], we employ the constrained quadratic programming (QP) problem to formulate and solve minimum-jerk polynomial trajectories of the virtual leader.

Based on the generated $M$ waypoints, we consequently have $m = M - 1$ intervals. The time-parametrised trajectory $\sigma(t) = [p_x^d, p_y^d, p_z^d, \psi_f^d]^\top$ can be modelled via piecewise polynomial functions of order $k_n$ over $m$ intervals as follows

$$\sigma(t) = \begin{cases} \sum_{j=0}^{k_n} x_{j1} t^j & \text{for } t_0 \le t < t_1 \\ \sum_{j=0}^{k_n} x_{j2} t^j & \text{for } t_2 \le t < t_2 \\ \dots & \\ \sum_{j=0}^{k_n} x_{jm} t^j & \text{for } t_{m-1} \le t \le t_m. \end{cases} \tag{1}$$

Augmenting those polynomial coefficients into a decision vector $\hat{x} \in \mathbb{R}^{4nm+1}$ with $\hat{x} = [x_1, x_2, \dots, x_m]$, the performance index to minimise the jerk of the trajectory can be formulated as

$$J = \int_0^\tau \gamma_p \left\| \frac{d^3\sigma(t)}{dt^3} \right\| dt = x^\top H x \tag{2}$$

where $H$ is a Hessian matrix corresponding to the desired penalty on the polynomial derivatives.

The constraints are considered in this mission: the waypoint, the continuity and the dynamic limitation. The first constraint is to ensure that the leader starts from the start point, flying through the waypoints and finish at the endpoint. The start and endpoint with waypoints constraints can be modelled via

$$A_0^{\text{start}} x_0 = b_0,\ A_\tau^{\text{end}} x_\tau = b_\tau,\ \text{and } A_i^{\text{start}} x_i = b_i. \tag{3}$$

The second constraint aims to ensure the continuity of the trajectories, that is, the end of $i$-th segment of the generated trajectory needs to be exactly equal to the start of the $i+1$-th segment. This can be formulated via

$$A_i^{\text{end}} x_i = A_{i+1}^{\text{start}} x_{i+1}. \tag{4}$$

The above matrices $A_k^{\text{start}}$ and $A_k^{\text{end}}$ maps the coefficients $x_k$ to the beginning and the end of the polynomial derivatives, respectively. Vector $b_k$ in the above equation is the waypoints that the virtual leader should reach.

The dynamics constraint is also applied to ensure no violation of the quadcopter dynamics. This constraint can be imposed via

$$A^{\text{dyn}} \hat{x} \le b^{\text{dyn}}, \tag{5}$$

where $A^{\text{dyn}}$ is the mapping matrix, and the $b^{\text{dyn}}$ is a vector containing the dynamics limitation of the leader. In this mission, we impose the the maximum velocity and acceleration of the leader.

In this paper, we employ the computation method proposed by [18]. The output of this trajectory refinement is an optimised time-parametrised position, velocity, acceleration and heading $\sigma$ of the virtual leader as given in (1).

### *D. Distance Computation*

Given a set of all deployed UAVs $\mathcal{V}_f$, we classify the UAVs into two sets: the unassigned and assigned sets. The unassigned set $\mathcal{V}_\alpha \subseteq \mathcal{V}_f$ contains all UAVs which are not or not yet assigned to an inspection area. The assigned set $\mathcal{V}_\beta \subset \mathcal{V}_f$ contains all UAVs assigned to an inspection area. If the system is in surveillance mode, all deployed UAVs are in the unassigned set. If it is in inspection mode, then some UAVs move to the assigned set depending on the request from the user, while the rests remain in the unassigned one. Notice that the set inequality of the assigned set implies that at least a UAV is not assigned to the inspection area. This ensures at least a UAV to compensate for the coverage gap left by the assigned UAVs.

Let $s = [p_0^\top, l_\alpha, l_\beta]^\top$ be the state of the mission manager at time $t$, where $p_0 = [p_{x,0}, p_{y,0}, p_{z,0}]^\top, l_\alpha$ and $l_\beta$ are the planar position of the virtual leader, the length of area covered by the unassigned UAVs, and the length of area covered by the assigned UAVs. Accordingly, the dynamics and the proposed control command of the formation can be modelled, respectively, as

$$\dot{s} = u_s, \text{ and } u_s = -k_s(s - s^d). \tag{6}$$

In above equation, $u_s = [u_p^\top, u_\alpha, u_\beta]^\top$, $k_s = \text{diag}(k_p, k_\alpha, k_\beta)$, and $s^d = [(p_0^d)^\top, l_\alpha^d, l_\beta^d]^\top$ denote the control input, control gain, and desired state of the formation, respectively. The desired position of the virtual leader can be expressed as

$$p_0^d = p^d + R(\psi_f^d) p_{\text{line}}^d, \tag{7}$$

where $p^d$ and $\psi_f^d$ are the position and heading of the virtual leader obtained from the trajectory refinement. The second term on the righthand side is to adjust the position of the virtual leader according to a given operation mode where $R(\psi_f^d)$ represents the rotation matrix with respect to the heading of the formation and

$$p_{\text{line}}^d = \eta([(l_\alpha^d + l_\beta^d)/2,\ 0.0,\ 0.0]^\top - p^d). \tag{8}$$

We set $\eta = 1$ if the inspection area is located on the right-hand side of the trajectory, $\eta = -1$ if the inspection area is located on the left-hand side of the trajectory, and $\eta = 0$ if there is no inspection area.

In the surveillance mode, the length of the unassigned area is stored as the surveillance coverage length, $l_f^s = l_\alpha$. This information is utilised in the inspection mode where, given the length of the inspection area $l_{\text{insp}}$ and its centre point $p_{\text{insp}} = [p_{\text{insp},x}, p_{\text{insp},y}]^\top$, and planar trajectory of the virtual leader $\hat{\sigma} = [p_x^d, p_y^d]^\top$, the desired coverage length of the unassigned and assigned UAVs can be obtained, respectively, via

$$l_\alpha^d = \|p_{\text{insp}} - \hat{\sigma}\| + \frac{l_f^s - l_{\text{insp}}}{2},\ \text{and } l_\beta^d = l_{\text{insp}}. \tag{9}$$

Based on the network topology whose graph is illustrated in Fig. 3a, the position of agent $i$ depends on the position of agent $i-1$, for $i = 1, 2, \dots, N$ where, accordingly, the position of agent $i$ can be expressed as

$$p_i = p_{i-1} + R(\psi_{\text{f}}^d)(p_{i-1} + \delta_{i-1,i}^p - p_0), \tag{10}$$

with $\delta_{i-1,i}^p = [\delta_{x,i-1,i}^p, \delta_{y,i-1,i}^p, \delta_{z,i-1,i}^p]$. For line formation with respect to $x$ axis in the local coordinate, the vector difference in $y$ and $z$ axes are zero. The vector difference between agent 0 and agent 1 on $x$ axis is calculated utilising the state $s$ via

$$\delta_{x,0,1}^p = \frac{l_\alpha + l_\beta - l_1}{2},\ \text{and } \delta_{y,0,1}^p = 0.0. \tag{11}$$

For $i = 1, 2, \dots, N$, to produce a line formation on $x$-axis in the local coordinate, the vector difference between agent $i-1$ and $i$ can be obtained via

$$\delta_{x,i-1,i}^p = \frac{l_{i-1} + l_i}{2},\ \text{and } \delta_{y,i-1,i}^p = 0.0, \tag{12}$$

where $l_i$ denotes the length of area coverage by agent $i$ calculated via $l_i = ((1-\lambda_i)l_\alpha^d + \lambda_i l_\beta^d)/((1-\lambda_i)|\mathcal{V}_\alpha| + \lambda_i|\mathcal{V}_\beta|)$, with $|\cdot|$ denoting the cardinality of the corresponding set. The scalar $\lambda_i = 0$ if agent $i$ belongs to the set of unassigned UAVs, $\mathcal{V}_\alpha$; and $\lambda_i = 1$ if agent $i$ belongs to the set of assigned UAVs, $\mathcal{V}_\beta$. For $i = 2, \ldots, N$, Employing the relationship between the coverage length and the altitude of a camera, the vector difference in the direction of $z$ can be obtained via:

$$\delta_{z,0,1}^p = \frac{l_1}{2\tan\left(\frac{\alpha}{2}\right)} - p_{z,0}^d, \text{ and } \delta_{z,i-1,i}^p = \frac{l_i - l_{i-1}}{2\tan\left(\frac{\alpha}{2}\right)}. \quad (13)$$

Finally, the outputs of the mission manager given to the UAVs are the position trajectory of the virtual leader $p_0$, the heading trajectory of the formation $\psi_\mathrm{f}$ and the vector difference $\delta_{i-1,i}^p$.

## V. Simulation

Consider $N = 7$ UAVs deployed to execute the surveillance and inspection operations to validate the designed algorithm. The UAVs move with a constant velocity of 10.0m/s or 36.0km/h. The camera angles are $\alpha = 0.73\mathrm{rad}$ and $\beta = 0.42\mathrm{rad}$. The aspect ratio $\rho$ of the camera is $16:9$ with resolution $1920 \times 1080$. The size of surveillance areas throughout this validation is kept identical. The simulations are carried out on a Linux-base computer with ROS2 and Python programming language. The details about the simulation scenarios are as follows.

**Scenario 1: Different image quality level** – This simulation scenario considers two simulation setups of expected image qualities: low and high. The setup and result for the surveillance and inspection operations are provided in Tables I and II, respectively.

| Quality | Formation's Coverage Area | | | Alt | $\sum$Dist | $\sum$Time |
|---|---|---|---|---|---|---|
| | $\sum L$(m) | $L$(m) | px(mm) | (m) | (km) | (h) |
| Low | 800.00 | 114.29 | 59.52 | 148.86 | 45.33 | 1.26 |
| High | 300.00 | 42.86 | 22.32 | 55.82 | 174.23 | 4.84 |

TABLE I: Surveillance Mode of Scenario 1.

| Quality | $\sum UAV$ | Assigned Area | | Unassigned Area | |
|---|---|---|---|---|---|
| | Assigned | $L/i$(m) | px(mm) | $L/i$(m) | px(mm) |
| Low | 5 | 27.00 | 14.06 | 425.00 | 221.35 |
| High | 5 | 27.00 | 14.06 | 157.48 | 82.02 |

TABLE II: Inspection Mode of Scenario 1.

The results of this first scenario can be observed in Fig. 5, for low and high image definitions. The axes are all in kilometer.

Based on the simulation results, it is observable that as the expected image quality in the surveillance mode is increased, and the pixel size is reduced, the UAVs need to operate at a lower altitude. As the altitude decreases, the fleets need to fly farther to cover the whole surveillance area. With constant velocity, farther distance implies longer execution time. If the higher image quality is expected, a surveillance mission consumes more time to complete. Otherwise, the UAV formation takes less operation distance and, therefore, consumes less execution time.

Given an equal number of assigned UAVs in the inspection mode, the results demonstrate that the pixel sizes in all inspection areas, which also correspond to image quality, are identical regardless of their image quality on the surveillance mode. This phenomenon happens because the number of the assigned UAVs and the size of inspection are also identical. The image quality in the unassigned area varies according to the total coverage length in the surveillance mode. Lower image quality leads to a larger size of the unassigned area than the one with higher image quality since the unassigned agents compensate for the more significant gap left by the assigned agents.

**Scenario 2: Different size of inspection area** – Two simulation setups consisting of small and big inspection area sizes are considered in this scenario. The setup and result for the surveillance and inspection operations are provided in Tables III and IV, respectively.

| Area | Formation's Coverage Area | | | Alt | $\sum$Dist | $\sum$Time |
|---|---|---|---|---|---|---|
| | $\sum L$(m) | $L$(m) | px(mm) | (m) | (km) | (h) |
| Small | 500.00 | 71.43 | 37.20 | 93.04 | 87.53 | 2.43 |
| Large | 500.00 | 71.43 | 37.20 | 93.04 | 87.53 | 2.43 |

TABLE III: Surveillance Mode of Scenario 2.

| Area | $\sum UAV$ | Assigned Area | | Unassigned Area | |
|---|---|---|---|---|---|
| | Assigned | $L/i$(m) | px(mm) | $L/i$(m) | px(mm) |
| Small | 5 | 27.00 | 14.06 | 307.50 | 160.16 |
| Large | 5 | 54.00 | 28.13 | 240.00 | 125.00 |

TABLE IV: Inspection Mode of Scenario 2.

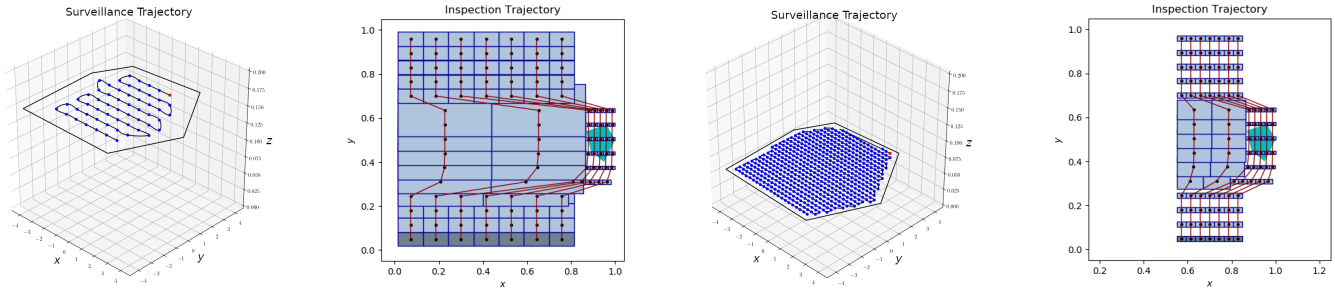
The results of this second scenario can be observed in Fig. 6, for small and large inspection areas, respectively. The axes are all in kilometer.

In this scenario, the expected image qualities in the surveillance mode are identical; therefore, the total distance and total execution time are identical. However, given an equal number of the deployed agent, we can observe that covering a larger inspection area leads to larger pixel size and low image quality. Since smaller assigned areas left larger unassigned areas, the image quality in the unassigned area deteriorates accordingly.

The simulation results demonstrate that the proposed strategy can successfully generate trajectories of the UAV formation and the desired distances among them for executing surveillance and inspection missions. The image quality of a surveillance area can be improved either by increasing the number of deployed UAVs or the grid resolution. In the inspection mode, more drones need to be deployed to improve the image quality of a given inspection area. However, deploying UAVs to the inspection area needs also consider the unassigned UAVs to prevent coverage gap caused by the assigned UAVs.

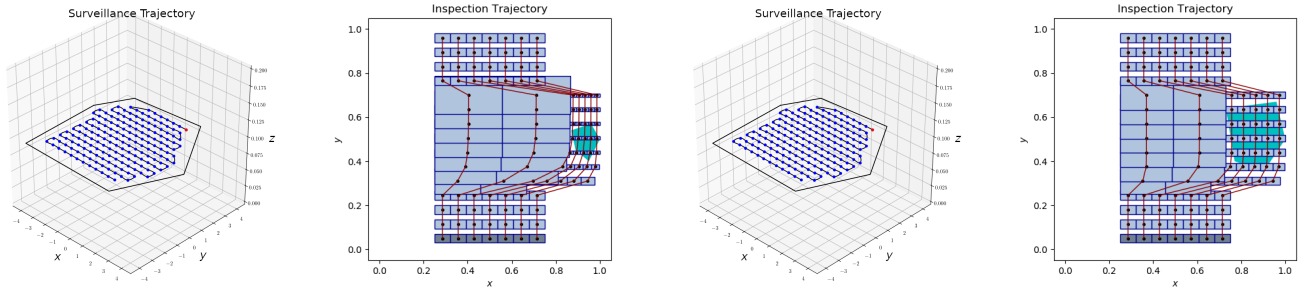
## VI. Conclusion

In this paper, we presented an algorithm for a mission manager to drive a formation of quadcopters with cameras to execute the surveillance and inspection operation in a given area. The proposed algorithm utilised the virtual leader network topology to drive the formation and computes the minimum-jerk optimal trajectories of coverage path planning waypoints that accommodate the UAV's dynamics limitation. The trajectories were




(a) Low quality surveillance. (b) Low quality inspection. (c) High quality surveillance. (d) High quality inspection.

Fig. 5: Scenario 1: Trajectories of the UAVs with different image quality level.




(a) Small-area surveillance. (b) Small-area inspection. (c) Large-area surveillance. (d) Large-area inspection.

Fig. 6: Scenario 2: Trajectories of the UAVs with big-sized inspection area.

designed to satisfy the position, velocity and acceleration constraints. In addition, the algorithm also considered the formation dynamics and utilised a feedback controller to dynamically adjust the formation parameters based on the inputs for the inspection operation. Finally, a set of numerical simulations have validated the effectiveness of the algorithm.

## REFERENCES


[1] R. Funada, M. Santos, T. Gencho, J. Yamauchi, M. Fujita, M. Egerstedt, Visual Coverage Maintenance for Quadcopters Using Nonsmooth Barrier Functions, Proceedings - IEEE International Conference on Robotics and Automation (2020) 3255–3261.

[2] E. Galceran, M. Carreras, A survey on coverage path planning for robotics, Robotics and Autonomous Systems 61 (12) (2013) 1258–1276.

[3] M. Schwager, B. J. Julian, M. Angermann, D. Rus, Eyes in the sky: Decentralized control for the deployment of robotic camera networks, Proceedings of the IEEE 99 (9) (2011) 1541–1561.

[4] T. Hatanaka, R. Funada, M. Fujita, 3-D visual coverage based on gradient descent algorithm on matrix manifolds and its application to moving objects monitoring, Proceedings of the American Control Conference (0) (2014) 110–116. arXiv:1309.5702, doi:10.1109/ACC.2014.6858663.

[5] O. Arslan, H. Min, D. E. Koditschek, Voronoi-Based Coverage Control of Pan/Tilt/Zoom Camera Networks, Proceedings - IEEE International Conference on Robotics and Automation (2018) 5062–5069.

[6] M. Schwager, B. J. Julian, D. Rus, Optimal coverage for multiple hovering robots with downward facing cameras (2009) 3515–3522.

[7] S. Papatheodorou, A. Tzes, Y. Stergiopoulos, Collaborative visual area coverage, Robotics and Autonomous Systems 92 (2017) 126–138.

[8] M. Tzes, S. Papatheodorou, A. Tzes, Collaborative Visual Area Coverage by Aerial Agents under Positioning Uncertainty, MED 2018 - 26th Mediterranean Conference on Control and Automation (2018) 149–154doi:10.1109/MED.2018.8442489.

[9] R. Funada, M. Santos, J. Yamauchi, T. Hatanaka, M. Fujita, M. Egerstedt, Visual coverage control for teams of quadcopters via control barrier functions, Proceedings - IEEE International Conference on Robotics and Automation 2019-May (2019) 3010–3016.

[10] S. Huang, H. Yang, W. L. Leong, R. Teo, Improved Multi-Camera Coverage Control of Unmanned Multirotors, 2020 International Conference on Unmanned Aircraft Systems, ICUAS 2020 (2020) 1103–1112doi:10.1109/ICUAS48674.2020.9213835.

[11] W. H. Huang, Optimal line-sweep-based decompositions for coverage algorithms, Proceedings - IEEE International Conference on Robotics and Automation 1 (2001) 27–32.

[12] J. S. Kim, B. K. Kim, Minimum-time grid coverage trajectory planning algorithm for mobile robots with battery voltage constraints, ICCAS 2010 - International Conference on Control, Automation and Systems (2010) 1712–1717.

[13] G. S. Avellar, G. A. Pereira, L. C. Pimenta, P. Iscold, Multi-UAV routing for area coverage and remote sensing with minimum time, Sensors (Switzerland) 15 (11) (2015) 27783–27803.

[14] C. Di Franco, G. Buttazzo, Energy-aware coverage path planning of UAVs, Proceedings - 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2015 (2015) 111–117.

[15] C. Di Franco, G. Buttazzo, Coverage Path Planning for UAVs Photogrammetry with Energy and Resolution Constraints, Journal of Intelligent and Robotic Systems: Theory and Applications 83 (3-4) (2016) 445–462.

[16] F. Bullo, J. Cortés, S. Martínez, Distributed control of robotic networks: A mathematical approach to motion coordination algorithms, 2009.

[17] D. Mellinger, V. Kumar, Minimum snap trajectory generation and control for quadrotors, Proceedings - IEEE International Conference on Robotics and Automation (2011) 2520–2525.

[18] C. Richter, B. Adam, R. Nicholas, Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments, Cailiao Yanjiu Xuebao/Chinese Journal of Materials Research 17 (5) (2013) 459–465.

[19] M. W. Mueller, M. Hehn, R. Dandrea, A Computationally Efficient Motion Primitive for Quadrocopter Trajectory Generation, IEEE Transactions on Robotics 31 (6) (2015) 1294–1310.

[20] A. Bry, C. Richter, A. Bachrach, N. Roy, Aggressive Flight of Fixed-Wing and Quadrotor Aircraft in Dense Indoor Environments 37 (June) (2015) 969–1002.