Interactive Multi-Robot Aerial Cinematography Through Hemispherical Manifold Coverage

Xiaotian Xu[1], Guangyao Shi[2], Pratap Tokekar[3], and Yancy Diaz-Mercado[1]

***Abstract*— This paper presents a distributed interactive framework to provide high-level position instructions for multi-robot aerial cinematography based on coverage over a hemisphere. The control strategy based on optimization of the coverage functional and geometric relationships over a hemisphere is presented. It enables multiple Unmanned Aerial Vehicles (UAVs) to coordinate their motion while tracking a dynamic (real or virtual) target, and can accommodate high-level human inputs to influence UAV concentration. In this framework, each UAV uses local information combined with exogenous inputs to determine its motion. The two inputs to the system, i.e., the predicted trajectory of the target and user-defined aesthetic preferences, are agnostic to the size of the multi-robot system (MRS). The proposed framework is validated using the PX4 SITL Autopilot simulator in Gazebo, and the scalability of the framework is verified via simulations.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are increasingly being used for both individual and commercial aerial cinematography due to their flexibility and maneuverability. Commercial products (e.g., DJI Mavric [1], Skydio, [2]) can automatically follow humans while avoiding obstacles or are operated by human pilots in a one-to-one fashion. For aesthetic objectives of UAV cinematography, both learning-based [3]–[5] and non-learning-based methods [6], [7] are integrated with trajectory optimization frameworks. Other approaches extend single UAV aerial cinematography to multiple UAVs. A series of works from [8]–[10] presents a control scheme for aerial cinematography with multiple UAVs by using a trajectory optimization method and defining canonical shot types with a parametric description of attributes. A high-level control framework is proposed in [11] to control drones by defining a parameter space for camera motion where the desired cinematographic properties and motion constraints are defined. The work in [12] studies multi-UAV aerial cinematography problems from the perspective of formation control, but the interactive control of the drones is not considered. A greedy strategy is used in [13] to coordinate the motion of multiple UAVs, where the authors consider a hemisphere structure centered at the actor for safety, but the geometry of a hemisphere is not explicitly used for control as we do in this paper.

This material is based upon research supported by, or in part by, the U.S. Office of Naval Research under award number N00014-21-1-2410.

[1]Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, USA. Email: xxu0116@umd.edu, yancy@umd.edu.

[2]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA. Email:gyshi@umd.edu.

[3]Department of Computer Science, University of Maryland, College Park, MD 20742, USA. Email: tokekar@umd.edu .

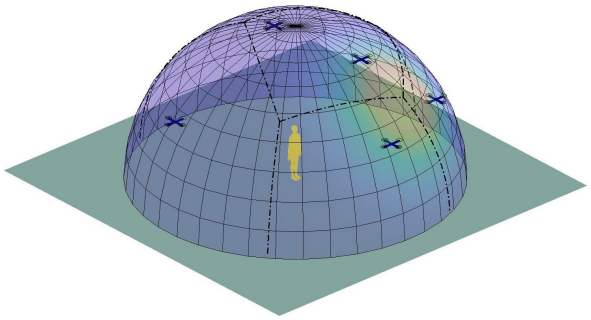


Fig. 1. An illustration of multi-robot cinematography using coverage over a hemisphere with UAVs observing an object from different points of view. UAVs form a configuration over the hemisphere based on the user's artistic preference (i.e., by emphasizing "hot spots" over the spherical surface).

Coverage control for multi-robot systems (MRS) allows for the robots to spread out over a domain of interest in order to minimize the proximity of every point to a robot. The formulation has been considered for static settings [14], [15], dynamic settings [16], and where the space must be explored persistently [17]. The nature of coverage control introduces a sense of inter-robot behavioral coordination for an MRS and provides an abstraction of the ensemble, i.e., simplifying and representing the state of the team by using time-varying densities [18] and time-varying boundaries [19], [20]. We aim to enable control of larger UAV teams for aerial cinematography based on coverage over time-varying hemispheres with time-varying densities that prescribe aesthetic preference by specifying the desired concentration of UAVs without explicitly controlling individual units, and the interactive control can be realized in real-time.

In this paper, we propose an interactive multi-robot aerial cinematography framework to provide high-level position instructions to the UAVs in a distributed fashion by leveraging MRS coverage control. The following principles are considered when designing a control strategy: 1) the UAVs should stay some distance from the target; 2) the MRS can dynamically observe the target from different points of view, as shown in Fig. 1; 3) the pilot should be able to manipulate the UAVs from some high-level specifications, i.e., the system should be user-friendly and easy to express different artistic styles, such that the pilot does not have to learn complex case-by-case commands. To meet these requirements, we assume a virtual hemisphere with its center located at the real or virtual target, and UAVs maintain a prescribed distance to the hemisphere's center. The UAVs

are deployed by defining their distribution over the hemisphere to provide different points of view. Furthermore, the abstraction of the state of the MRS can be used as two exogenous inputs, i.e., predicted motion of the hemisphere (target) and the density over the hemispherical surface, to manipulate the MRS's behavior. These two inputs can be determined either in decentralized or centralized manners, e.g., obtained from decentralized autonomous sensor-based techniques, or provided by a pilot using a human-computer interface such as a joystick or a tablet (for more information on interfaces for MRS, see [21] and references therein). The proposed control strategy allows an MRS to distribute over the hemisphere and track the exogenous inputs efficiently. Although we do not explicitly consider collision avoidance between UAVs, MRS coverage naturally achieves safety by spreading agents out over a domain. In this work, we do not consider environmental obstacles, as is the case in most existing applications.

The contributions of this paper are multi-fold: **1)** A framework for generating high-level position instructions for a multi-robot aerial cinematography application is proposed. The coverage over a moving hemisphere with time-varying density functions is studied to design a distributed control strategy. Control policies for a large group of robots are synthesized by the deployment of UAVs over the hemisphere. The proposed control strategy is scalable, enabling applications to a large group of robots. **2)** The control strategy can efficiently track the two exogenous inputs, i.e., the motion of the hemisphere and the time-varying "hot spots" (density), which can be used to influence the behavior of UAVs for meeting aesthetic objectives in real-time, and the canonical shot types defined in [8] can be realized. **3)** The mathematical expressions for components in the control law, e.g., generalized mass and centroid for the spherical manifold, and their partial derivatives, are provided based on geometric relationships over a hemisphere. The properties of the control law are discussed. **4)** The control strategy is validated by the PX4 Autopilot simulator in Gazebo, and its scalability is verified in simulations.

The organization of this paper is as follows: Section II investigates the generalized coverage over a hemisphere as well as the geometric relationships in the formulation. Section III gives the proposed control framework and discusses the properties of the control laws. Section IV presents simulation results and provides discussions on the performance of the control strategy. Finally, Section V provides conclusions.

II. Coverage Control Over A Hemisphere

Let $x_i \in \mathcal{D} \subseteq \mathbb{R}^3$ be the position of the i^{th} robot, $i \in \{1, \dots, N\}$, in the robot workspace $\mathcal{D}$, such that $X(t) = \left[x_1^T(t), \cdots, x_N^T(t)\right]^T$ is the configuration of the multi-agent team. Further define a time-varying hemisphere $\mathcal{S}(t) \subset \mathcal{D}$, and let $p_i(t)$ be the projection of robot i's position onto hemispherical surface $\mathcal{S}(t)$ with respect to the center of hemisphere. Thus, we have $p_i(t) \in \mathcal{S}(t)$, $\forall t$, and the projected configuration of robots on hemisphere $P(t) = \left[p_1^T(t), \cdots, p_N^T(t)\right]$.

A. Generalized Coverage Formulation

Motivated by the classical coverage over convex domains with the Euclidean metrics [15], the coverage formulation is generalized to manifolds under Riemannian metrics [22]. The coverage functional, also called locational cost [14], provides a metric of the coverage performance and is defined as

$$\mathcal{H}(P(t), t) = \sum_{i=1}^{N} \int_{V_i(P(t),t)} f(g(q, p_i))\phi(q, t)\, dq \quad (1)$$

where $f(\cdot) : \mathbb{R}_+ \to \mathbb{R}$ is a smooth and strictly increasing function; $g(q, p_i)$ is a choice of metric; $\phi(q, t) : \mathcal{S}(t) \times [0, \infty) \to (0, \infty)$ is a density function that captures the relative importance of the points on the spherical surface at time t, differentiable in both arguments; and $V_i(P(t), t)$ denotes region of dominance of i^{th} robot obtained from a partition of the total coverage manifold. It is worthy to mention that the coverage functional (1) is slightly modified here to accommodate the time-variations in the density function and coverage manifold. In our problem $f(\cdot) = (\cdot)^2$, and $g(q, p_i)$ is the geodesic distance between point q and i^{th} robot's projected position p_i on the hemisphere.

The hemisphere is properly partitioned into regions of dominance for each agent. We employ Voronoi tessellation and generalize it based on the geodesic metric,

$$V_i(P, t) = \{q \in \mathcal{S}(t) \mid g(q, p_i) \leq g(q, p_j)\ \forall j\} \quad (2)$$

where $\bigcup_{i=1}^{N} V_i(P, t) = \mathcal{S}(t)$, and for ease of notation we have dropped the explicit time dependency on the configuration of the multi-robot system.

B. Generalized Centroidal Voronoi Tessellations

In the case of convex domains with Euclidean metric, a well-known necessary condition for minimizing the coverage functional (1) is that the robots form a centroidal Voronoi tessellation (CVT) configuration [14] over the domain of interest, i.e., $p_i(t) = c_i(P, t)\ \forall i$, where $c_i(P, t)$ is the center of mass (i.e., centroid) of i^{th} Voronoi cell $V_i(P, t)$, and $c_i(P, t) \in V_i(P, t)$, $\forall t$ due to the convexity of the domain and Euclidean metric, but this is not always true for a generalized manifold with non-Euclidean metrics. In our problem, we study the generalized centroid under Voronoi tessellation (2). Before doing this, we state the following lemma which is needed for the investigation,

Lemma 1 (Leibniz Integral Rule [23]): Let $\Omega(p, t)$ be a region that depends smoothly on t and that has a well-defined boundary $\partial\Omega(p, t)$. If $F(p, t) = \int_{\Omega(p,t)} f(p, t, q)\, dq$ for differentiable f, then

$$\frac{\partial F}{\partial t} = \int_{\Omega(p,t)} \frac{\partial f}{\partial t}\, dq + \int_{\partial\Omega(p,t)} f(p, t, q)\hat{n}(q)^T \frac{\partial q}{\partial t}\, dq$$

where $\hat{n}(q)$ is the unit outward normal for $q \in \partial\Omega(p, t)$.

Let us neglect the time-variations embedded in the system temporarily, then to minimize the coverage functional (1), we first find the gradient of it with respect to robots' projected

positions,

$$\frac{\partial \mathcal{H}}{\partial p_i} = \frac{\partial}{\partial p_i}\int_{V_i} g(q,p_i)^2\phi(q,t)\,dq + \frac{\partial}{\partial p_i}\sum_{j\in\mathcal{N}_i}\int_{V_j} g(q,p_j)^2\phi(q,t)\,dq + \frac{\partial}{\partial p_i}\sum_{j\notin\mathcal{N}_i\cup i}\int_{V_j} g(q,p_j)^2\phi(q,t)\,dq \quad (3)$$

where $\mathcal{N}_i$ denotes the Delaunay graph neighbor set of i^{th} robot. One can immediately find that the last term is zero due to its independence on p_i. By applying Lemma 1 the first two terms in (3) respectively, we get

$$\frac{\partial}{\partial p_i}\int_{V_i} g(q,p_i)^2\phi(q,t)\,dq = \int_{V_i} 2g(q,p_i)\phi(q,t)\frac{\partial g(q,p_i)}{\partial p_i}\,dq + \sum_{j\in\mathcal{N}_i}\int_{\partial V_{ij}} g(q,p_i)^2\phi(q,t)\hat{n}_{ij}^T(q)\frac{\partial q}{\partial p_i}\,dq \quad (4)$$

and

$$\frac{\partial}{\partial p_i}\sum_{j\in\mathcal{N}_i}\int_{V_j} g(q,p_j)^2\phi(q,t)\,dq = \sum_{j\in\mathcal{N}_i}\int_{\partial V_{ij}} g(q,p_j)^2\phi(q,t)\hat{n}_{ji}^T(q)\frac{\partial q}{\partial p_i}\,dq \quad (5)$$

where $\partial V_{ij} = V_i \cup V_j$ denotes the shared boundary (i.e., bisector) of two Voronoi cells; $\hat{n}_{ij}(q)$ is the unit outward normal on the shared boundary with respect to p_i, likewise the $\hat{n}_{ji}$. It is worthy to emphasize that both $\hat{n}_{ij}(q)$ and $\hat{n}_{ji}(q)$ lie in the tangential plane attached to q and in opposite directions, i.e., $\hat{n}_{ij}(q) = -\hat{n}_{ji}(q)$. Therefore,

$$\frac{\partial \mathcal{H}}{\partial p_i} = \int_{V_i} 2g(q,p_i)\phi(q,t)\frac{\partial g(q,p_i)}{\partial p_i}\,dq. \quad (6)$$

The shortest path (geodesy) between two points on the surface of a sphere is always a segment of a great circle (orthodrome). The length of this shortest path is called the great-circle distance, orthodromic distance, or spherical distance. The geometric relationships associated to spherical surfaces offer a chance to derive the specific expression for ${}^{\partial g(q,p_i)}/_{\partial p_i}$ in (6). Before that, we define some notations for describing these geometric relationships.

Let p and q be points on a spherical surface $\mathcal{S}$ with center o and radius r, and vector $v_{pq} = q - p$ have its Euclidean length $l_{pq} = \|v_{pq}\|$ and versor (unit vector) $u_{pq} = v_{pq}/\|v_{pq}\|$. One could find the variation of a unit vector,

$$\delta u_{pq} = \frac{1}{l_{pq}} P_{u_{pq}^{\perp}}\delta v_{pq}, \quad P_{u_{pq}^{\perp}} = I - u_{pq}u_{pq}^T \quad (7)$$

The geodesy (great-circle distance) between points p and q is simply the arclength $g(p,q) = r\Delta\theta$, where $\Delta\theta$ is the angle between vectors v_{op} and v_{oq}. By the definition of inner product, we have

$$g(p,q) = r\cdot\arccos\langle u_{op},\, u_{oq}\rangle. \quad (8)$$

Denoting $\text{in}_{qp}^u = \langle u_{oq},\, u_{op}\rangle$, the variation of $g(p,q)$ is found,

$$\begin{aligned}\delta g(p,q) &= -\frac{r}{\sqrt{1-(\text{in}_{qp}^u)^2}}\delta\,\text{in}_{qp}^u \\ &= -\frac{r}{\sqrt{1-(\text{in}_{qp}^u)^2}}\left(u_{oq}^T\,\delta u_{op} + u_{op}^T\,\delta u_{oq}\right)\end{aligned} \quad (9)$$

then based on (7)-(9), we can find that

$$\frac{\partial g(q,p_i)}{\partial p_i} = -\frac{1}{\sqrt{1-(\text{in}_{qp_i}^u)^2}}u_{oq}^T\,P_{u_{op_i}^{\perp}} = \frac{-1}{r\sqrt{1-(\text{in}_{qp_i}^u)^2}}\left(q + \left(\text{in}_{qp_i}^u - 1\right)o - \text{in}_{qp_i}^u\,p_i\right)^T. \quad (10)$$

By defining

$$F_{i,1}(q,p_i) = \frac{g(q,p_i)}{r\sqrt{1-(\text{in}_{qp_i}^u)^2}}\left(q + \left(\text{in}_{qp_i}^u - 1\right)o\right), \quad (11)$$

and

$$F_{i,2}(q,p_i) = \frac{g(q,p_i)\,\text{in}_{qp_i}^u}{r\sqrt{1-(\text{in}_{qp_i}^u)^2}}, \quad (12)$$

the generalized mass and centroid can be defined as,

$$\sigma_i = \int_{V_i} F_{i,1}(q,p_i)\phi(q,t)\,dq, \quad (13)$$

$$m_i(P,t) = \int_{V_i} F_{i,2}(q,p_i)\phi(q,t)\,dq, \quad (14)$$

$$c_i(P,t) = \sigma_i/m_i, \quad (15)$$

and we can thus rearrange (3) to be

$$\frac{\partial \mathcal{H}}{\partial p_i} = 2m_i(p_i - c_i)^T. \quad (16)$$

With the generalized mass $m_i(P,t)$ and centroid $c_i(P,t)$, one can rewrite (1) into (16) which is similar to the continuous time version of Lloyd's algorithm [18]. However, as we mentioned previously, it is not true that $c_i(P,t) \in V_i(P,t)$, $\forall\, t$. Thus, we conclude the following lemma,

Lemma 2 (Location of Generalized Centroid): The generalized centroid c_i for robot i's Voronoi cell on a spherical surface does not lie on the spherical surface but lies on the tangential plane attached to location p_i except for a CVT configuration.

Proof: To show that c_i lies on the tangential plane attached to p_i other than lies on the surface of sphere as long as $c_i \neq p_i$. It suffices to show that the vectors $(c_i - p_i)$ and $(p_i - o)$ are orthogonal to each other, i.e.,

$$\left\langle \frac{\partial \mathcal{H}}{\partial p_i},\, (p_i - o)\right\rangle = 0.$$

Given that $g(q,p_i)$ and $\phi(q,t)$ in the integrand of $\partial\mathcal{H}/\partial p_i$ are scalars, we have that

$$\begin{aligned}\frac{\partial g(q,p_i)}{\partial p_i}(p_i-o) &= \frac{\partial g(q,p_i)}{\partial p_i} l_{op_i} u_{op_i} \\ &= \frac{-l_{op_i}}{\sqrt{1-(\mathrm{in}^u_{qp_i})^2}} u_{oq}^T \left(I - u_{op_i} u_{op_i}^T\right) u_{op_i} \\ &= \frac{-l_{op_i}}{\sqrt{1-(\mathrm{in}^u_{qp_i})^2}} \left(u_{oq}^T u_{op_i} - u_{oq}^T u_{op_i} u_{op_i}^T u_{op_i}\right) = 0\end{aligned}$$

due to $u_{op_i}^T u_{op_i} = 1$. On the other hand, for any $q \neq p_i \in \mathcal{S}$, the vectors $(q-p_i)$ and (p_i-o) cannot be orthogonal, completing the proof. ∎

From Lemma 2, the following corollary is induced,

Corollary 1: The point $F_{i,1}(q,p_i)/F_{i,2}(q,p_i) \in \mathbb{R}^3$ in (15), where $F_{i,2}(q,p_i) \in \mathbb{R}$, lies on the tangential plane attached to p_i.

Proof: From (10) - (12), the following relationship exists,

$$F_{i,1} - F_{i,2}\, p_i = -g(q,p_i)\frac{\partial g(q,p_i)}{\partial p_i};$$

and from the proof of Lemma 2, we have that

$$\left\langle (F_{i,1} - F_{i,2}\, p_i),\, (p_i - o) \right\rangle = 0.$$

Since $F_{i,2} \in \mathbb{R}$, the proof follows. ∎

Lemma 2 suggests that the implementation of the generalized Lloyd's algorithm alone would lead agents to always move tangentially to the spherical surface and gradually leave the surface. In addition, we need to carefully design the control strategy to also account for the time-variations in the system. We introduce the proposed control strategy in the following section to account for these two issues.

III. Control Strategy Design

For interactive multi-robot aerial cinematography, we have to control the pose of each robot in the multi-robot system as well as the orientation of the camera gimbal installed on robots. In this paper, we only discuss high-level robot position control; we design translational velocity $\dot{x}_i$ of the UAVs and assume single integrator dynamics. A dynamics model and low-level pose control of UAVs are provided in simulation, e.g., using the ArduPilot SITL (Software-In-The-Loop). In real implementations, many approaches can be used to incorporate high-level instructions into quadrotor models [24]. We omit the orientation control of camera gimbals installed on UAVs, but many vision-based control techniques [25], [26] that could be integrated with our high-level position control, and their discussion fall outside of the scope of the main problem considered in this paper.

A. Translational Position Control of Robots

Based on the analysis above, we seek a control strategy that, 1) drives agents along with the tangential control signal (16) due to gradient descent of the coverage functional (i.e., a tracking term to move towards the corresponding generalized centroid); 2) has a normal component with respect to the spherical surface to make agents remain on the surface; and 3) has a term to efficiently track the time-variations from both time-varying densities and time-varying hemisphere.

Therefore, we propose the overall control law as follows,

$$\dot{x}_i = u_{i,t} + u_{i,n} + u_o \tag{17}$$

where $u_{i,t} = \dot{p}_i$ which is a control signal from TVD-D_0 (one can also use TVD-D_n: *time-varying densities/domains, distributed case with n-hop adjacency information*) which is tangential to the spherical surface and tracks the time-varying density functions; the TVD-D_0 and TVD-D_n, as well as their pros and cons will be introduced and discussed in the following content; $u_{i,n}$ is a normal control signal for maintaining UAVs on the spherical surface; and u_o is the feedforward term for tracking the movement of the object of interest located at the center of the hemisphere.

As we discussed in Section II-B, by leveraging the gradient descent approach, i.e., along the negative direction of (16), agents will move tangentially to the spherical surface. This will cause problems in the real implementation which makes agents get off the spherical surface until CVT configuration is achieved, such that the quality of cinematography will be undermined. The situation could be even worse when time-variations are added to the system.

Therefore, a control signal that is normal to the spherical surface and makes robots maintain a proper distance from the tracked object, i.e., asymptotically moves to and rests on the hemisphere, will be favorable. An energy-like function is designed where the energy increases when agents get off the spherical surface and reaches its minimum when agents are on the surface,

$$\mathcal{E}(\|x_i - o\|) = \frac{1}{2}\left(\|x_i - o\| - r\right)^2,$$

then we define the normal control signal as

$$u_{i,n} = -\kappa_n \frac{\partial \mathcal{E}}{\partial x_i}^T = -\kappa_n \left(\|x_i - o\| - r\right) u_{ox_i} \tag{18}$$

where $\kappa_n > 0$ is a control gain. For the normal component of the control signal above, we have the following claim,

Lemma 3: With (18), the normal component of agents' dynamics will asymptotically converge to the desired radius r, as long as $u_{i,t}$ does not contain a normal component. This suggests that agents will asymptotically approach and stay on the prescribed hemisphere.

Proof: Let the energy itself $\mathcal{E} > 0$ be a Lyapunov candidate, and consider $u_{i,n}$ is the only normal component, then we have

$$\dot{\mathcal{E}} = \frac{\partial \mathcal{E}}{\partial x_i}\dot{x}_i = -\kappa^n \left\| \frac{\partial \mathcal{E}}{\partial x_i} \right\|^2 < 0$$

except at equilibrium. Thus the control signal asymptotically drives UAVs back to the hemisphere once they get off. ∎

It is worth mentioning that each UAV can keep its own distance r_i from the center of hemisphere o, and this would not affect any results we are presenting and will enhance

the flexibility of configurations of the UAVs in the space for cinematography. Without loss of generality, we set $r_i = r$, $\forall i$ to enforce UAVs stay on the surface of the hemisphere. Another remarkable thing is that $u_{i,t} = \dot{p}_i$ could contain a normal component if $\dot{p}_i$ is obtained from TVD-D$_n$ ($n \geq 1$). We will discuss it in the next section.

B. Accounting for Time-Variations

Under the Euclidean metric, a control law exponentially drives robots to CVT configurations under time-varying density and time-varying domain cases are developed in [18], [19]. The TVD-C (stands for *time-varying densities/ domains, centralized case*) is given by

$$\dot{P} = \left(I - \frac{\partial C}{\partial P}\right)^{-1} \left(\kappa(C(P,t) - P) + \frac{\partial C}{\partial t}\right), \qquad (19)$$

where $C(P,t) = \left[c_1^T(P,t), \cdots, c_N^T(P,t)\right]^T$; the matrix $\partial C/\partial P$ is a block matrix with sparse structure that encodes Delaunay (Voronoi) adjacency information; the inverse of matrix $(I - \partial C/\partial P)$ requires overall adjacency information over the entire network, which makes the control law centralized; the $\partial C/\partial t$ term is the feedforward signal to capture time-variations due to density functions and domains. We refer the reader to [18] for the system stability analysis with control law (19). To bypass the difficulty of computing the matrix inverse in (19), and at the same time, decentralize the control law, the use of Neumann series results in TVD-D$_n$,

$$\dot{P} = \sum_{\ell=0}^{n} \left(\frac{\partial C}{\partial P}\right)^{\ell} \left(\kappa(C(P,t) - P) + \frac{\partial C}{\partial t}\right), \qquad (20)$$

where n sets how many hops (number of edges in the shortest path over the network graph) of information to be used by an agent for planning motion.

We are going to track the time-varying density functions through the term $\partial C/\partial t$ as stated above, but we track the time-varying hemisphere by feedforwarding the motion of the hemisphere directly to the overall control signal to bypass the necessity of applying Leibniz integral rule to a 2D moving surface immersed in 3D space [27]. The $\partial c_i/\partial t$ for time-varying density alone can be found as

$$\begin{aligned} \left.\frac{\partial c_i}{\partial t}\right|_{\phi} &= \frac{1}{m_i^2}\left(\frac{\partial \sigma_i}{\partial t} m_i - \sigma_i \frac{\partial m_i}{\partial t}\right) \\ &= \frac{1}{m_i}\int_{V_i} (F_{i,1} - c_i F_{i,2}) \frac{\partial \phi(q,t)}{\partial t}\, dq. \end{aligned} \qquad (21)$$

Based on Lemma 2 and Corollary 1, we can immediately claim the following statement.

Corollary 2: The signal vector $\left.\frac{\partial c_i}{\partial t}\right|_{\phi} \in \mathbb{R}^3$ in (21) for tracking the time-varying density in the system lies on the tangential plane attached to p_i.

As we mentioned, the matrix $\partial C/\partial P$ encodes the Delaunay (Voronoi) adjacency information of the network of the system. Given the expression in (20), the TVD-D$_0$, which does not include matrix $\partial C/\partial P$, requests nothing but neighbors' positions for computing Voronoi cell. However, the requirement of the minimum amount of information, in turn, undermines the efficiency of tracking time-variations slightly. In contrast, the TVD-D$_1$, which contains the matrix $\partial C/\partial P$, not only requires positions of neighbors in 1-hop, but also neighbors' information including centroids and time-variation of centroids. The more information a robot possesses, the more efficiently it can track the time-variation. However, in our scenario, i.e., coverage over a hemisphere under a geodesic metric, the statement is not completely true. When agents are in proximity of a CVT configuration, they can more efficiently track the time-variations, but the Jacobian matrix $\partial C/\partial P$ will make the behavior of agents undesirable when the agents are away from a CVT configuration, generating some normal velocity components that drive agents off the surface. In other words, the tracking performance of the control strategy that includes $\partial C/\partial P$ improves once robots get near a CVT configuration. For the completeness of paper, we show the expressions for $\partial c_i/\partial p_i$ and $\partial c_i/\partial p_j$ but omit the derivation due to space constraints. Application of Lemma 1 and geometric relationship on hemisphere results in

$$\begin{aligned} \frac{\partial c_i}{\partial p_i} = \frac{1}{m_i}\Bigg(&\int_{V_i} \left(\frac{\partial F_{i,1}}{\partial p_i} - c_i \frac{\partial F_{i,2}}{\partial p_i}\right) \phi(q,t)\, dq \\ &+ \sum_{j \in \mathcal{N}_i} \int_{\partial V_{ij}} (F_{i,1} - c_i F_{i,2})\, \phi(q,t)\, \hat{n}_{ij}^T(q) \frac{\partial q}{\partial p_i}\, dq\Bigg), \end{aligned} \qquad (22)$$

$$\frac{\partial c_i}{\partial p_j} = \frac{1}{m_i}\int_{\partial V_{ij}} (F_{i,1} - c_i F_{i,2})\, \phi(q,t)\, \hat{n}_{ij}^T(q) \frac{\partial q}{\partial p_j}\, dq, \qquad (23)$$

where

$$\begin{aligned} &\frac{\partial F_{i,1}}{\partial p_i} - c_i \frac{\partial F_{i,2}}{\partial p_i} = \\ &\quad \frac{g(q,p_i) - r \mathrm{in}_{qp_i}^u \sqrt{1 - (\mathrm{in}_{qp_i}^u)^2}}{r^2 (1 - (\mathrm{in}_{qp_i}^u)^2)^{3/2}} (o - c_i)\ u_{oq}^T P_{u_{op_i}^{\perp}} \\ &+ \frac{g(q,p_i)\mathrm{in}_{qp_i}^u - r\sqrt{1 - (\mathrm{in}_{qp_i}^u)^2}}{r^2 (1 - (\mathrm{in}_{qp_i}^u)^2)^{3/2}} (q - o)\ u_{oq}^T P_{u_{op_i}^{\perp}}, \end{aligned} \qquad (24)$$

and

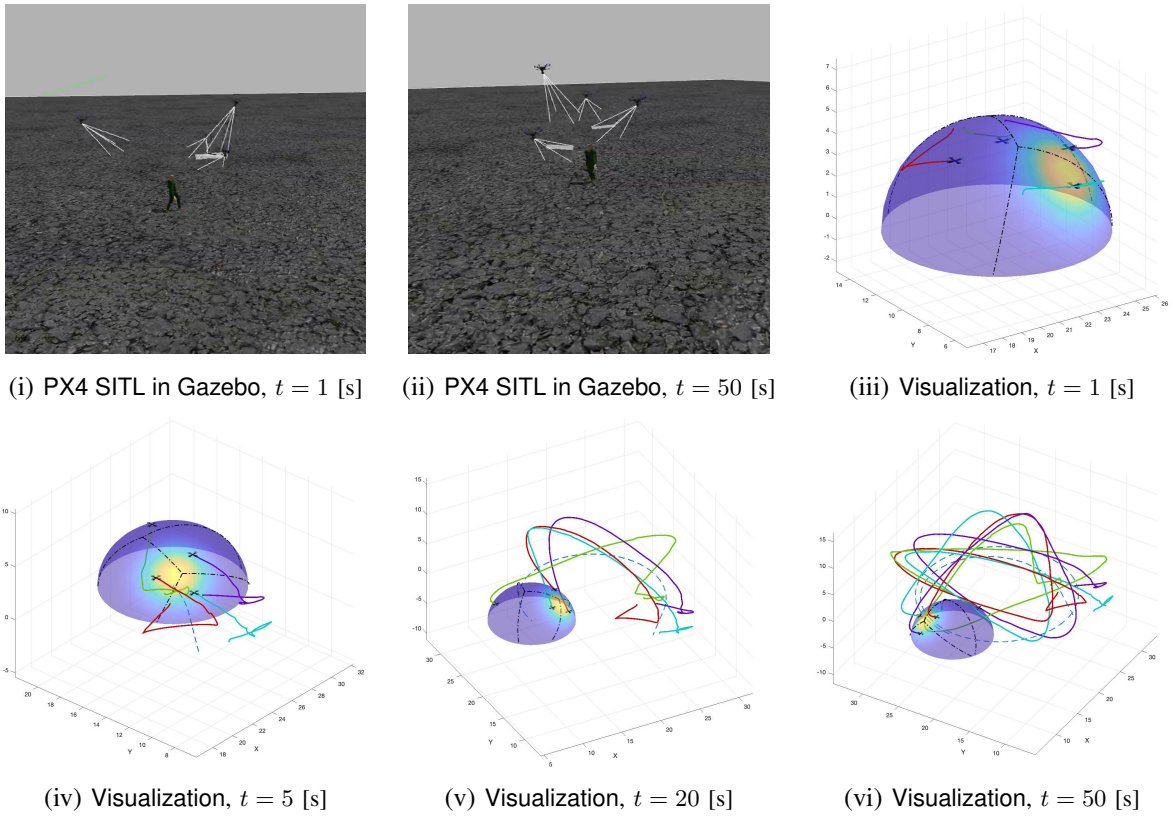
$$\hat{n}_{ij}^T(q) \frac{\partial q}{\partial p_i} = \frac{1}{\|u_{op_j} - u_{op_i}\|} u_{oq}^T P_{u_{op_i}^{\perp}}, \qquad (25)$$

$$\hat{n}_{ij}^T(q) \frac{\partial q}{\partial p_j} = -\frac{1}{\|u_{op_i} - u_{op_j}\|} u_{oq}^T P_{u_{op_j}^{\perp}}. \qquad (26)$$

Based on the discussion and expressions above, for coverage control over a hemisphere under geodesic metric, the Jacobian matrix $\partial C/\partial P$ in control law (20) would generate additional normal velocity components that lead robots to move off the spherical surface when $c_i \neq p_i$ for agent i. Let us take TVD-D$_1$ as an example and break it down at node level,

$$\begin{aligned} \dot{p}_i = &\left(\kappa\,(c_i - p_i) + \frac{\partial c_i}{\partial t}\right) + \frac{\partial c_i}{\partial p_i}\left(\kappa\,(c_i - p_i) + \frac{\partial c_i}{\partial t}\right) \\ &+ \sum_{j \in \mathcal{N}_i} \frac{\partial c_i}{\partial p_j}\left(\kappa\,(c_j - p_j) + \frac{\partial c_j}{\partial t}\right) \end{aligned} \qquad (27)$$

One can immediately find that the first and the third terms in (27) are in the tangential plane of the spherical surface at



(i) PX4 SITL in Gazebo, $t = 1$ [s] (ii) PX4 SITL in Gazebo, $t = 50$ [s] (iii) Visualization, $t = 1$ [s]

(iv) Visualization, $t = 5$ [s] (v) Visualization, $t = 20$ [s] (vi) Visualization, $t = 50$ [s]

Fig. 2. Visualization of one example simulation (i.e., Sim. Case #4 in TABLE I) in Gazebo environment using PX4 SITL Autopilot simulator with $N = 4$ UAVs. We show snapshots of Gazebo environment at two time instances, and snapshots of virtualization of information including virtual hemisphere, density function, Voronoi tessellation, UAVs' trajectories, etc. at different time instances.

p_i as a consequence of Lemma 2, Corollary 1, and Corollary 2. However, for the second term in (27), while the integral along arcs in (22), i.e., the second part of $\partial c_i/\partial p_i$, will lead to a vector in the tangential plane at p_i, it is hard to mathematically disprove that the first part of $\partial c_i/\partial p_i$ has the same property since it is a result of integral over an area on a spherical surface, but one can find that for any $q \in V_i$ the expression (24) would not result in a tangential signal in (27). We validate this numerically in simulations.

IV. Simulations

In this section, the proposed framework for interactive multi-robot aerial cinematography is validated in the Gazebo environment by using the PX4 SITL Autopilot simulator [28]. In the simulations, a group of UAVs is commanded to follow the translational position and velocity instructions induced from the proposed control strategy. We validate the control strategy under various scenarios, i.e., tracking static human objects (the virtual hemisphere is static) and dynamic human objects (the hemisphere is moving) with and without user-defined aesthetic preferences. Meanwhile, we test the performance of different control laws, i.e., TVD-D_1, TVD-D_0, and Lloyd's algorithm. During the simulations, the motion of the human is captured from the Gazebo environment in real-time and then is feedforwarded to the multi-robot system. The aesthetic preference is defined by assigning "hot spots" over the spherical surface using time-varying density functions. UAVs will concentrate on the "hot spots" to demonstrate their relative importance. A time-varying density function is designed as follows to test our strategy,

$$\phi(q,t) = A(t)\exp\left(-\frac{g(q,p_c(t))^2}{\sigma_g^2}\right), \tag{28}$$

where $A(t) = (a+1) + a\sin(2\pi f \cdot t) \geq 1$ denotes the value of density changing along time with frequency f; $g(q,p_c(t))$ denotes the geodesic distance on the surface with the time-varying "hot spot" $p_c(t)$ which can be user-defined. Thus the time-derivative of ϕ can be found

$$\frac{\partial \phi}{\partial t} = \left(\frac{2A(t)g(q,p_c)}{\sigma_g^2\sqrt{1-(\text{in}^u_{qp_c})^2}}u_{oq}^T P_{u_{\frac{\perp}{op_c}}}\dot{p}_c + 2\pi f a\cos(2\pi f t)\right)\exp\left(-\frac{g(q,p_c)^2}{\sigma_g^2}\right) \tag{29}$$

The parameters in simulations are designed as follows, number of UAVs $N = 4$, radius of virtual hemisphere $r = 5$ [m], amplitude of density function $A(t) = 10$ with $\dot{A}(t) = 0\,(f = 0)$, variance associated to Gaussian function $\sigma_g = 2.2$, "hot spot" at $p_c = [\pi/4, -\pi/4]$ with velocity $\dot{p}_c = [0, -\pi/10]$ in spherical coordinates (which is defined by [polar angle, azimuthal angle]). The two metrics we used

to evaluate the performance are the aggregate off-surface error $E_{\mathcal{S}} = \sum_{i=1}^{N} \text{abs}\left(\|p_i - o\| - r\right)$, and the aggregate CVT error $E_{\text{CVT}} = \sum_{i=1}^{N} \|p_i - c_i\|$. These two metrics evaluate how well the UAVs track the two exogenous inputs, and low value of the two metrics indicates that UAVs follow and address human shot ideas accurately.

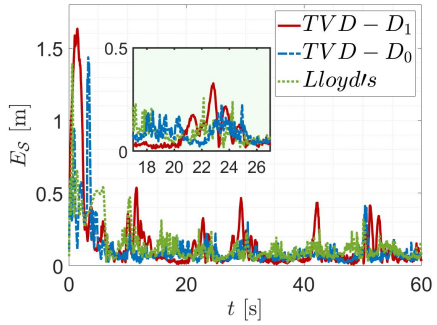
The parameters and mean value of the two performance metrics (i.e., $\overline{E}_{\mathcal{S}}$ and $\overline{E}_{\text{CVT}}$) of simulations using PX4 SITL Autopilot simulator in Gazebo environment are shown in TABLE I. The snapshots of an example (the case #4) including various information, e.g., virtual hemisphere, trajectories, and density over the surface, are shown in Fig. 2; and the time profiles of $E_{\mathcal{S}}$ and E_{CVT} of different control laws (Sim. case #3, #4, and #5) are compared in Fig. 3. It can be seen from the data that the average deviation of UAVs getting off the hemisphere surface is around 0.0423 [m] per UAV, and the average error of centroid tracking is around 0.25 [m] per UAV under the proposed control framework. Although the simulator includes various uncertainties like the real world such that noisy responses are obtained, from the results, it is not hard to find that in terms of E_{CVT}, the control law "TVD-D_1" outperforms "TVD-D_0" and "Lloyd's", which indicates that the feedforward term captures the time-variations in the system. On the other hand, in terms of $E_{\mathcal{S}}$, "TVD-D_1" drives UAVs off the hemisphere surface when UAVs are not in the proximity of CVT configurations, which conforms to our discussion about the $\partial C/\partial P$ in the last section. These effects are more obvious in the single-integrator simulations which will be discussed next.

The scalability of the proposed control strategy is also tested with single integrator dynamics via MATLAB simulations. In these simulations, the radius of the virtual hemisphere $r = 8$ [m], and the circular motion of the object is that the object moves along a circular trajectory with radius $R = 10$ [m] and frequency 0.033 [Hz] (i.e., it takes 30 [s] per cycle). Moreover, the density are designed the same as stated in (28) and (29). The different cases and

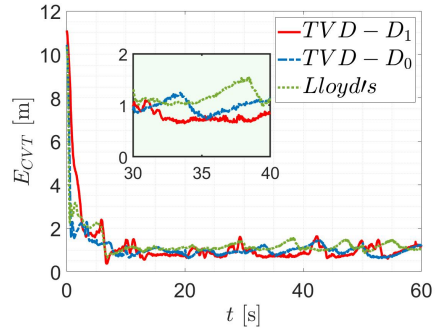
TABLE I

SIM. #1 - #5: PARAMETERS AND RESULTS OF PERFORMANCE METRICS FOR 4 UAVS USING PX4 SITL AUTOPILOT SIMULATOR IN GAZEBO. SIM. #6 - #10: PARAMETERS AND RESULTS OF PERFORMANCE METRICS FOR 30 UAVS WITH SINGLE INTEGRATOR DYNAMICS.

Sim.	Object Motion	$\phi(q,t)$	Control Law	$\overline{E}_{\mathcal{S}}$ [m]	$\overline{E}_{\text{CVT}}$ [m]
$N = 4$					
#1	Static	Gaussian	TVD-D_0	0.4999	1.9342
#2	Circular	Uniform	TVD-D_0	0.0830	0.2507
#3	Circular	Gaussian	TVD-D_1	0.0968	0.9200
#4	Circular	Gaussian	TVD-D_0	0.0757	0.9548
#5	Circular	Gaussian	Lloyd's	0.0910	1.1497
$N = 30$					
#6	Static	Gaussian	TVD-D_0	0.0019	4.1122
#7	Circular	Uniform	TVD-D_0	0.0067	0.4888
#8	Circular	Gaussian	TVD-D_1	0.0084	3.6121
#9	Circular	Gaussian	TVD-D_0	0.0072	4.2853
#10	Circular	Gaussian	Lloyd's	0.0071	5.3207



(i) Aggregate off-surface error $E_{\mathcal{S}}$ v.s. time t



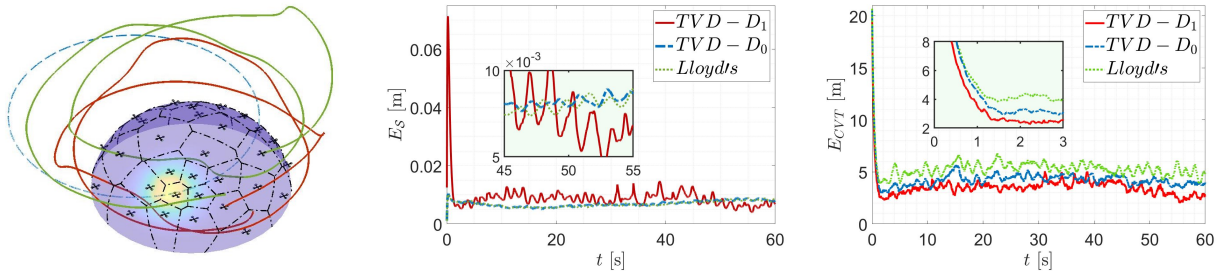
(ii) Aggregate CVT error E_{CVT} v.s. time t

Fig. 3. (i) Profile of $E_{\mathcal{S}}$ v.s. t, and (ii) E_{CVT} v.s. t for the simulation shown in Fig. 2 (case #3, #4 and #5 in TABLE I).

their corresponding results of $\overline{E}_{\mathcal{S}}$ and $\overline{E}_{\text{CVT}}$ are designed and listed in TABEL I (case #6 - #10). From the data obtained, the proposed control framework is capable of maintaining UAVs on the spherical surface (the average deviation of UAVs off the surface is 2.087e^{-4} [m] per UAV), and tracking the time-variations in the system (the average of $\|c_i - p_i\|$ is 0.1188 [m] per UAV). Similarly, a visualization and comparisons of the performance of control laws "TVD-D_1", "TVD-D_0", and "Lloyd's" are shown in Fig. 4. Without introducing nonlinearities and uncertainties, it is easier to see the characteristic of the proposed control strategy with different control laws. As shown in Fig. 4(ii), the "TVD-D_1" drives UAVs off the surface from time to time, especially at (approximately) $t = 1$ since the initial configuration of UAVs are far away from a CVT configuration where the $\partial C/\partial P$ plays a dominating role and generates normal velocity components to the hemispherical surface. However, one can find from Fig. 4(iii) that "TVD-D_1" is the fastest one that converges due to every UAV possesses more local information under "TVD-D_1" than that under "TVD-D_0" and "Lloyd's". On the other hand, "TVD-D_0" beats "Lloyd's" since the presence of feedforward term $\partial c_i/\partial t$. The outcomes of all the simulations above are as expected and conform with our discussions in previous sections.

V. CONCLUSIONS

To tackle the difficulty of "one pilot many robots" that lies in the field of aerial cinematography, a framework is proposed for interactive control of multiple UAVs. The



(i) Simulation of 30 UAVs (ii) $E_{\mathcal{S}}$ v.s. t (iii) E_{CVT} v.s. t

Fig. 4. (i) A visualization (only two UAVs' trajectories are plotted), (ii) profile of aggregate off-surface error $E_{\mathcal{S}}$ v.s. time t, and (iii) profile of aggregate CVT error E_{CVT} v.s. time t for the simulation of 30 UAVs (case #8, #9 and #10 in TABLE I).

framework allows UAVs to efficiently track the predicted motion of a real or virtual object and collaboratively complete the filming tasks according to shot idea input by humans. As demonstrated by the simulation results, the control law provides accurate tracking of the subject while maintaining the desired distance and distributing according to the commanded aesthetic. The designed control strategy provides an efficient way to enable human-swarm interaction such that a pilot can easily influence the behavior of the multi-robot system to express aesthetic filming preferences. The control framework can be applied to a large group of systems.

REFERENCES

[1] DJI Official, "Mavic pro–DJI," 2022, Accessed 2 February 2022. [Online]. Available: https://www.dji.com/en/mavic

[2] Skydio, "Skydio 2," 2022, Accessed 2 February 2022. [Online]. Available: www.skydio.com

[3] R. Bonatti, W. Wang, C. Ho, A. Ahuja, M. Gschwindt, E. Camci, E. Kayacan, S. Choudhury, and S. Scherer, "Autonomous aerial cinematography in unstructured environments with learned artistic decision-making," *Journal of Field Robotics*, vol. 37, no. 4, pp. 606–641, 2020.

[4] C. Huang, Z. Yang, Y. Kong, P. Chen, X. Yang, and K.-T. T. Cheng, "Learning to capture a film-look video with a camera drone," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 1871–1877.

[5] Y. Dang, C. Huang, P. Chen, R. Liang, X. Yang, and K.-T. Cheng, "Imitation learning-based algorithm for drone cinematography system," *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[6] T. Nägeli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges, "Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1696–1703, 2017.

[7] R. Bonatti, C. Ho, W. Wang, S. Choudhury, and S. Scherer, "Towards a robust aerial cinematography platform: Localizing and tracking moving targets in unstructured environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 229–236.

[8] A. Alcántara, J. Capitán, A. Torres-González, R. Cunha, and A. Ollero, "Autonomous execution of cinematographic shots with multiple drones," *IEEE Access*, vol. 8, pp. 201 300–201 316, 2020.

[9] A. Alcántara, J. Capitán, R. Cunha, and A. Ollero, "Optimal trajectory planning for cinematography with multiple unmanned aerial vehicles," *Robotics and Autonomous Systems*, vol. 140, p. 103778, 2021.

[10] V. Krátký, A. Alcántara, J. Capitán, P. Štěpán, M. Saska, and A. Ollero, "Autonomous aerial filming with distributed lighting by a team of unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7580–7587, 2021.

[11] Q. Galvane, C. Lino, M. Christie, J. Fleureau, F. Servant, F.-L. Tariolle, and P. Guillotel, "Directing cinematographic drones," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 3, pp. 1–18, 2018.

[12] R. Tallamraju, E. Price, R. Ludwig, K. Karlapalem, H. H. Bülthoff, M. J. Black, and A. Ahmad, "Active perception based formation control for multiple aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4491–4498, 2019.

[13] A. Bucker, R. Bonatti, and S. Scherer, "Do you see what i see? coordinating multiple aerial cameras for robot cinematography," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7972–7979.

[14] J. Cortés, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks: Variations on a theme," in *Mediterranean Conference on Control and Automation*. Lisbon, Portugal Lisbon, Portugal, 2002, pp. 9–13.

[15] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[16] I. I. Hussein and D. M. Stipanovic, "Effective coverage control using dynamic sensor networks," in *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 2747–2752.

[17] X. Xu, E. Rodrıguez-Seda, and Y. Diaz-Mercado, "Persistent awareness-based multi-robot coverage control," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 5315–5320.

[18] Y. Diaz-Mercado, S. G. Lee, and M. Egerstedt, "Human–swarm interactions via coverage of time-varying densities," *Trends in Control and Decision-Making for Human–Robot Collaboration Systems*, pp. 357–385, 2017.

[19] X. Xu and Y. Diaz-Mercado, "Multi-agent control using coverage over time-varying domains," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 2030–2035.

[20] ——, "Multi-robot control using coverage over time-varying non-convex domains," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4536–4542.

[21] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 9–26, 2015.

[22] S. Bhattacharya, R. Ghrist, and V. Kumar, "Multi-robot coverage and exploration on riemannian manifolds with boundaries," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 113–137, 2014.

[23] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *SIAM review*, vol. 41, no. 4, pp. 637–676, 1999.

[24] T. Milburn, "Analysis of advanced control methods for quadrotor trajectory tracking," Ph.D. dissertation, The Ohio State University, 2018.

[25] E. Malis, "Survey of vision-based robot control," *ENSIETA European Naval Ship Design Short Course, Brest, France*, vol. 41, p. 46, 2002.

[26] J. Chen and P. Carr, "Autonomous camera systems: A survey," in *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[27] H. Flanders, "Differentiation under the integral sign," *The American Mathematical Monthly*, vol. 80, no. 6, pp. 615–627, 1973.

[28] K. Xiao, S. Tan, G. Wang, X. An, X. Wang, and X. Wang, "Xtdrone: A customizable multi-rotor uavs simulation platform," in *2020 4th International Conference on Robotics and Automation Sciences (ICRAS)*. IEEE, 2020, pp. 55–61.