Angle-Aware Coverage Control for 3-D Map Reconstruction With Drone Networks

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2021.3135466, IEEE Control Systems Letters

Angle-Aware Coverage Control for 3-D Map Reconstruction with Drone Networks*

Takumi Shimizu[1], Shunya Yamashita[1], Takeshi Hatanaka[1], Kuniaki Uto[2], Martina Mammarella[3], and Fabrizio Dabbene[3]

***Abstract*—In this paper, we address environmental monitoring for 3-D map reconstruction using drone networks. In view of the fact that the 3-D reconstruction requires images from a variety of viewing angles, we first formulate a novel angle-aware coverage control problem, based on a concept of virtual field that combines the position of a monitoring target and the viewing angle. Then, we define an objective function with an importance index, updated according to the history of the past coverage states. In the present scenario, drones are required not only to minimize the objective function but also to take situation-adaptive actions, e.g. a drone should quickly escape a region having low importance indices. To this end, we design a QP-based controller that switches actions depending on the current importance index. We finally demonstrate the effectiveness of the presented controller through simulations on Robot Operating System.**

I. INTRODUCTION

Drones have increasingly been a key technology in environmental monitoring in several fields, e.g., smart farming [1], airborne gas sensing [2], and wildfire tracking [3]. In particular, drone networks with distributed control are expected to enhance the monitoring efficiency and robustness against failures while ensuring computational scalability. Coverage control [4] is a promising distributed solution to environmental monitoring with drone networks, as reported in the literature [5], [6], [7]. Most of such studies assume that the drone moves with a camera overlooking the ground.

A motivation for sampling images over environment is 3-D map reconstruction through techniques like Structure from Motion (SfM), e.g., in the applications of 3-D printable prefabricated structures [8], the landslide inventory map [9], and farm mapping [10]. In a scenario, each point in the environment needs to be sampled from various viewing angles. Nonetheless, standard coverage control [5] does not distinguish samples from different viewing angles. Some papers like [5] reflect the viewing angle in the sensing performance function, but it is irrelevant to the above issue. Additionally, coverage control typically leads drones/robots to a stationary configuration. Meanwhile, in the present case, drones are required to move around over the environment for taking rich images. To address the issue, the papers [11], [12], [13], [14] present persistent coverage control, wherein the drones/robots take patrolling motion persistently by raising or lowering the so-called density function (or importance index) according to the history of the past coverage states. Performance guarantees for such time-varying density functions are also addressed in [15], [16], and an online update of the density is also investigated in [17], [18]. However, the motion is also not always suitable for the considered scenario. In the present case, drones do not need to revisit a point to take the same image.

The above requirement is trivially reflected by prohibiting the importance index from raising, but it poses another challenge. Namely, if a drone is approaching a well-observed region having low importance indices, the objective function gets low and the standard gradient-based coverage schemes tend to decelerate the drones. However, the drones are actually expected to quickly escape such a region with high speed. Rather, they should slow down when they are in monitoring regions with high indices not fully observed in the past. In short, the drones need to take situation-adaptive actions depending on the current importance index.

In this paper, we present a novel angle-aware coverage control scheme for 3-D map reconstruction. We first present a new problem formulation of environmental monitoring, taking into account the viewing angle. In particular, we employ a higher dimensional virtual field that combines not only the position of a monitoring target but also the viewing angle. Then, we present a quadratic programming(QP)-based controller to reflect the aforementioned situational adaptability. Furthermore, in order to ensure the viability of the controller, we present a computationally efficient implementation with safety certificates. Finally, we verify that the proposed controller achieves ideal situation-adaptive and angle-aware monitoring through simulations.

The contributions of this paper are summarized below:

- We present a novel angle-aware coverage control problem suitable for map reconstructions using a concept of a higher dimensional virtual field and monotonically decreasing importance index.
- We present a novel situation-adaptive controller based on the QP-based controller.
- We present a computationally efficient version of the controller in order to ensure implementability.

*This research was partially funded by the project "New technical and operative solutions for the use of drones in Agriculture 4.0" (PRIN 2017, Prot. 2017S559BB) and JSPS KAKENHI Grant Number 21K04104.

[1] T. Shimizu, S. Yamashita, and T. Hatanaka are with School of Engineering, Tokyo Institute of Technology, Tokyo 152-8552, JAPAN. {shimizu@hfg., yamashita.s.ag@hfg., hatanaka@}sc.e.titech.ac.jp

[2] K. Uto is with School of Computing, Tokyo Institute of Technology, Tokyo 152-8552, JAPAN. uto@ks.c.titech.ac.jp

[3] M. Mammarella and F. Dabbene are with Institute of Electronics, Computer and Telecommunication Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy. {martina.mammarella, fabrizio.dabbene}@ieiit.cnr.it

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2021.3135466, IEEE Control Systems Letters

II. PRELIMINARY: ZEROING CONTROL BARRIER FUNCTION AND QP-BASED CONTROLLER

Let us consider the system $\dot{x} = f(x) + g(x)u$, where $x \in \mathbb{R}^N$ is the state, $u \in \mathcal{U} \subseteq \mathbb{R}^M$ is the control input, and $f : \mathbb{R}^N \to \mathbb{R}^N$ and $g : \mathbb{R}^N \to \mathbb{R}^{N\times M}$ are the vector fields, which are Lipschitz continuous. We assume that $x(t)$ has a unique solution on $[t_0, t_1]$.

Following the approach in [19], we consider a continuously differentiable function $b : \mathbb{R}^N \to \mathbb{R}$ and a set $\mathcal{C} := \{x \in \mathbb{R}^N \mid b(x) \geq 0\}$. The function b is said to be *zeroing control barrier function (ZCBF)* for the set $\mathcal{C}$ if there exists a set $\mathcal{D}$ with $\mathcal{C} \subseteq \mathcal{D} \subset \mathbb{R}^N$ such that,

$$\sup_{u\in\mathcal{U}} [L_f b(x) + L_g b(x)u + \alpha(b(x))] \geq 0, \ \forall x \in \mathcal{D},$$

where $L_f b(x), L_g b(x)$ are the Lie derivatives of $b(x)$ along $f(x)$ and $g(x)$, and α is an extended class $\mathcal{K}$ function. The above inequality means that there always exists $u \in \mathcal{U}$ that enforces the constraint $x \in \mathcal{C}$ at the boundary of the set $\mathcal{C}$ as long as b is a ZCBF.

A set $\mathcal{S} \subseteq \mathbb{R}^N$ is said to be *forward invariant* if $x(t) \in \mathcal{S}$ holds for all $t \in [t_0, t_1]$ and for any $x(t_0) \in \mathcal{S}$. It is shown in [19] that the forward invariance of the set $\mathcal{C}$ will be rendered by any Lipschitz continuous controller u which satisfies the constraint $L_f b(x) + L_g b(x)u + \alpha(b(x)) \geq 0$. Given a nominal input $u_{\rm nom}$, [19] also presents the following QP-based (state feedback) controller.

$$u^*(x) = \arg\min_{u\in\mathcal{U}} \|u - u_{\rm nom}\|^2$$
$$\text{s.t. } L_f b(x) + L_g b(x)u + \alpha(b(x)) \geq 0.$$

The controller achieves the closest control action to the nominal one while satisfying the forward invariance of the set $\mathcal{C}$.

III. PROBLEM SETTINGS

A. Drones, Virtual Field, and Geometry

We consider a situation where n drones, with the index set $\mathcal{I} := \{1, \ldots, n\}$, equipped with an on-board camera, live in a 3-D space as shown in Fig. 1. Throughout this paper, we assume that all drones are locally controlled so that their altitudes and attitudes are both constant and common among all drones, and all drones lie on a 2-D plane as illustrated in Fig. 1 [1]. Without loss of generality, the world frame Σ_w is arranged so that its z-axis is perpendicular to the plane. The set of (x, y)-coordinates on the plane is denoted by $\mathcal{P}$.

The position coordinates of the drone $i \in \mathcal{I}$ with respect to Σ_w are represented by x_i, y_i, and z_i. As mentioned above, z_i $(i \in \mathcal{I})$ are assumed to take a common value z_c among all drones. In this paper, we mainly consider a 2-D motion of $p_i := [x_i \ y_i]^T \in \mathcal{P} \subseteq \mathbb{R}^2$. Then, the dynamics of each drone $i \in \mathcal{I}$ is then assumed to follow

$$\dot{p}_i = u_i, \ u_i \in \mathcal{U} \subseteq \mathbb{R}^2,$$

[1]This assumption is motivated by the fact that path planning of a drone for aerial image sampling normally focuses on the 2-D coverage while keeping the flight altitude constant as in Fig. 1 [2]. The present approach is in principle still applicable with a few modifications even in the absence of this assumption, but more detailed investigations on 3-D motion control are left as future work.

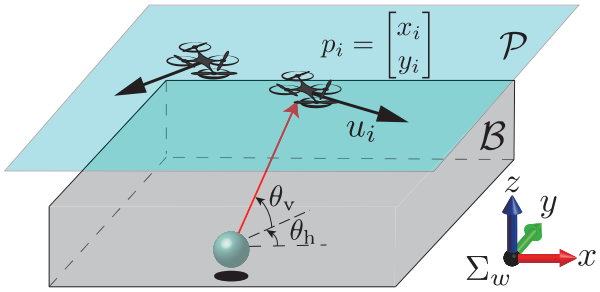


Fig. 1: Illustration of the drone field (blue plane) $\mathcal{P}$ and the target field (gray box) $\mathcal{B}$ with the world frame Σ_w. The i-th drone is monitoring a $[\theta_{\rm h} \ \theta_{\rm v}]$ side of a object inside $\mathcal{B}$.

where u_i is the velocity input to be designed. In the sequel, we denote the collocations of $p_1, \ldots, p_n$ and $u_1, \ldots, u_n$ by $p = (p_1, \ldots, p_n)$ and $u = (u_1, \ldots, u_n)$, respectively.

The target scenario addressed in this paper is the following: the drones take images of the *ground* in order to reconstruct a 3-D map of a compact subset of the ground itself, e.g. using the SfM techniques [8], [9], [10]. To this end, every point on the ground needs to be monitored from rich viewing angles. Now, in view of this objective, the exact shape of the ground is not available for the control of drones. Instead, we assume that a compact set $\mathcal{B} \subset \mathbb{R}^3$, containing the ground surface, is available as a prior knowledge. Then, the objective is relaxed to observe every point in $\mathcal{B}$ from rich viewing angles. In the sequel, $\mathcal{B}$ is termed *target field*.

In order to characterize the viewing angle, we define two angles, $\theta_{\rm h}$ and $\theta_{\rm v}$, for every point in $\mathcal{B}$, as illustrated in Fig. 1. The angle $\theta_{\rm h}$ is a horizontal angle defined as $\theta_{\rm h} \in \Theta_{\rm h} \subseteq [-\pi, \pi)$, and $\theta_{\rm v}$ is a vertical angle defined as $\theta_{\rm v} \in \Theta_{\rm v} \subseteq (0, \pi/2]$. For the 3-D map reconstruction, we have to take images of every point in the target field $[x \ y \ z]^T \in \mathcal{B}$ relative to $\sum_w$ from various $\theta_{\rm h}$ and $\theta_{\rm v}$. To this end, we propose the *virtual field* defined as $(x, y, z, \theta_{\rm h}, \theta_{\rm v})$-coordinated region, which is denoted by a compact set $\mathcal{Q}_c \subset \mathbb{R}^5$. Consequently, we address the angle-aware monitoring problem as the coverage problem of the virtual field.

We next consider the geometry between the drones and the virtual field. The drone's position to monitor $[x \ y \ z]$ from the viewing angle characterized by $\theta_{\rm h}$ and $\theta_{\rm v}$ is uniquely determined as the intersection between $\mathcal{P}$ and the viewing axis (red vector illustrated in Fig. 1). Precisely, we define the mapping $\zeta : \mathcal{Q}_c \to \mathcal{P}$ as

$$\zeta : [x \ y \ z \ \theta_{\rm h} \ \theta_{\rm v}]^T \mapsto \begin{bmatrix} x - (z_c - z)\tan\left(\frac{\pi}{2} - \theta_{\rm v}\right)\cos\theta_{\rm h} \\ y - (z_c - z)\tan\left(\frac{\pi}{2} - \theta_{\rm v}\right)\sin\theta_{\rm h} \end{bmatrix}.$$

We assume that the monitoring performance of the i-th drone at point $q \in \mathcal{Q}_c$ is modeled by the distance between the drone's position p_i and the monitoring position $\zeta(q)$. Specifically, we employ the following performance function $h : \mathcal{P} \times \mathcal{Q}_c \to [0, 1]$ with a design parameter $\sigma > 0$ which should be tuned depending on the characteristics of the sensor so that h is small enough for all $q \in \mathcal{Q}$ such that the first three elements of q, namely $[x \ y \ z]^T$, are outside of the field

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2021.3135466, IEEE Control Systems Letters

of view.

$$h(p_i, q) \coloneqq \exp\left(-\frac{\|p_i - \zeta(q)\|^2}{2\sigma^2}\right). \tag{1}$$

There are various options for the performance function, but the present one provides remarkable benefits as shown in the following sections. In particular, taking $\|p_i - \zeta(q)\|$ gives a computationally efficient solution, as shown in Section V-A.

B. Objective Function

In the present scenario, if a drone observes a point $q \in \mathcal{Q}_c$ once, it is reasonable to assume that the importance of the point decays. To reflect this issue, we discretize the virtual field $\mathcal{Q}_c$ as a collection of m 5-D polyhedra, termed cells in the sequel, and define $\mathcal{Q} \coloneqq \{q_j\}_{j\in\mathcal{M}}$, $\mathcal{M} \coloneqq \{1, \dots, m\}$, where q_j implies the gravity point of j-th cell. Note that each cell is assumed to have the same volume A. Let us now assign a value $\phi_j \in [0, \infty)$, termed importance index, to each cell $j \in \mathcal{M}$. In the sequel, we denote the collocations of ϕ_j by $\phi = (\phi_1, \dots, \phi_m)$.

The importance index ϕ_j should decay if q_j is monitored by drones. It is also reasonable to assume that the decay rate depends on the performance function h defined in (1). Thus, we present the following update rule of the index ϕ_j, i.e.,

$$\dot{\phi}_j = -\delta \max_{i\in\mathcal{I}} h(p_i, q_j)\phi_j \ (\delta > 0). \tag{2}$$

A rule similar to (2) is presented in [11], [12], [13], [16], where the authors increase the index if a point is not observed by any drone. This rule allows drones to take persistently patrolling motion over the field, but such persistent motion is not required in the present scenario. This is why we consider the rule (2) that renders each ϕ_j monotonically decreasing.

Let us now present an aggregate objective function to be minimized as:

$$J \coloneqq \sum_{j=1}^{m} \phi_j A. \tag{3}$$

We see from (1) and (2) that ideal images are expected to be taken by drones if J goes to zero. Thus, the primary objective is to control drones so that J approaches zero. However, ϕ_j in (2) is inherently monotonically decreasing and any control algorithm, e.g., random motion, achieves this objective over an infinitely long time interval. To rule out such inefficient solutions, we impose a secondary objective on the drones motion in the sequel.

To enhance the mission efficiency, each drone needs to change its behavior depending on the progress of the image sampling. Specifically, they are expected to escape a region with small ϕ_j that was well-observed in the past, while they have to be stick to a region with large ϕ_j that has not been fully observed in the past. We add this requirement as the secondary objective. To describe the objective more precisely, we present a metric I_i for the value of taking images at a position p_i. To this end, we present a partition of the set $\mathcal{M}$ as

$$\mathcal{V}_i(p) \coloneqq \{j \in \mathcal{M} \mid \|p_i - \zeta(q_j)\| \le \|p_k - \zeta(q_j)\| \ \ \forall k \in \mathcal{I}\}.$$

The metric I_i is then defined as $I_i \coloneqq \sum_{j\in\mathcal{V}_i(p)} \delta h(p_i, q_j)\phi_j A$ which corresponds to the contribution by drone i to reducing J in (3) since

$$\begin{aligned}\dot{J} &= \sum_{j=1}^{m} \dot{\phi}_j A = -\sum_{j=1}^{m} \delta \max_{i\in\mathcal{I}} h(p_i, q_j)\phi_j A \\ &= -\sum_{i=1}^{n} \sum_{j\in\mathcal{V}_i(p)} \delta h(p_i, q_j)\phi_j A = -\sum_{i=1}^{n} I_i.\end{aligned}$$

Suppose now that drone i is located at p_i such that $I_i < \gamma$, for a given $\gamma > 0$. If the drone is controlled so as to achieve $I_i \ge \gamma$, it must try to escape the region with small ϕ_j. The secondary objective is thus rephrased as controlling each drone i so that it pursues $I_i \ge \gamma$ if $I_i < \gamma$, while it tries to stay at p_i otherwise. Such mode switches are embodied by the QP-based controller in Section II by taking $b_i(p, \phi) \coloneqq I_i - \gamma$ as a candidate of the ZCBF and $u_{\rm nom} = 0$. Additionally, if drones are controlled so that $b_i \ge 0$, namely $I_i \ge \gamma$, the primary objective $J \to 0$ would be expected to be met eventually. This is why we mainly focus on the secondary objective.

IV. QP-BASED CONTROLLER DESIGN

In this section, we present a solution to meet the objective stated in the previous section. To this end, we employ the concept of the QP-based controller in Section II taking $b = b_i(= I_i - \gamma)$ and $u_{\rm nom} = 0$. The function b_i is however not a ZCBF, as confirmed by the scene with $\phi_j = 0 \ \forall j \in \mathcal{M}$. The constraint in the QP-based controller is thus softened as:

$$(u_i^*, w_i^*) = \underset{(u_i, w_i)\in\mathcal{U}\times\mathbb{R}}{\arg\min} \ \epsilon\|u_i\|^2 + |w_i|^2 \tag{4a}$$

$$\text{s.t. } \dot{b}_i + \alpha(b_i) \ge w_i, \tag{4b}$$

where w_i is a slack variable and ϵ is a positive constant that determines strength of the penalty on constraint violations, the second term of (4a). Even though b_i is not a ZCBF, the QP-based controller brings the following two benefits.

First, the constraint (4b) gives a condition for the input u_i to meet $b_i \ge 0$ as below, although the inequality itself does not contain the input.

Theorem 1. Suppose that no q_j $(j \in \mathcal{M})$ is located on the boundary of $\mathcal{V}_i(p)$. When $\alpha : \mathbb{R} \to \mathbb{R}$ is set as a linear function $\alpha(b_i) = ab_i$, where $a > 0$ is a positive scalar, the problem (4) is equivalently reformulated as

$$\begin{aligned}(u_i^*, w_i^*) = \underset{(u_i, w_i)\in\mathcal{U}\times\mathbb{R}}{\arg\min} \ &\epsilon\|u_i\|^2 + |w_i|^2 \\ \text{s.t. } &\xi_{1i}^T u_i + \xi_{2i} \ge w_i,\end{aligned} \tag{5}$$

where

$$\begin{aligned}\xi_{1i} &\coloneqq \sum_{j\in\mathcal{V}_i(p)} -\delta\frac{p_i - \zeta(q_j)}{\sigma^2} h(p_i, q_j)\phi_j A, \\ \xi_{2i} &\coloneqq -a\gamma + \sum_{j\in\mathcal{V}_i(p)} \left(-\delta^2 h^2(p_i, q_j) + a\delta h(p_i, q_j)\right)\phi_j A.\end{aligned}$$

Proof. See Appendix. □

When a q_j is exactly on the boundary of $\mathcal{V}_i(p)$, b_i is indifferentiable. However, it does not affect the applicability of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2021.3135466, IEEE Control Systems Letters

the approach in practice since the boundaries of the Voronoi cells have measure zero and the controller is implemented on a digital processor.

Secondly, the controller embodies the situation-adaptive motion raised as secondary objective. In other words, if b_i is far bigger than 0, the constraint gets inactive and u_i gets equal to zero, which implies that the drone tends to stay at the current position when $\mathcal{V}_i$ contains many cells not fully observed in the past. Conversely, if $b_i \geq 0$ is violated or close to be violated, u_i takes a large value to increase b_i, and the drone tends to escape from a well-observed region.

Given ϕ_j $(j \in \mathcal{V}_i)$, the controller (5) would be implementable in a distributed manner since the Voronoi partition is locally computed. Note however that the importance update (2) must be executed by the central computer, as assumed in [11], [12], [13], since each drone hardly knows if other drones have monitored point $q_j \in \mathcal{Q}$ in the past. Some readers may have a concern about using a central computer, but we remark that the computation at the computer is almost scalable and it is viable in many applications. Please refer to Remark 1 in [16] for more details.

We also add one more remark. [11] and [16] ignore dependency of p_i on the evolution of ϕ_j in the input computation since the first order derivative of J is independent of u_i. On the other hand, the present controller employs the second order derivative and explicitly reflects the dependency.

V. COMPUTATIONALLY EFFICIENT IMPLEMENTATION WITH SAFETY CERTIFICATES

A. Computationally Efficient Implementation

The QP-based controller (5) is hardly implemented in real time since the cardinality of $\mathcal{Q}$ tends to be very large.

To address the issue, we discretize the drone field $\mathcal{P}$ by a collection of l polygons $\mathcal{A} \coloneqq \{\mathcal{A}_1, ..., \mathcal{A}_l\}$ and their gravity points $\mathcal{X} \coloneqq \{\chi_k\}_{k\in\mathcal{L}} \subset \mathcal{P}, \mathcal{L} \coloneqq \{1, ..., l\}$, where $l \ll m$. Note that each polygon has the same area. According to the compression of $\mathcal{Q}$ onto $\mathcal{X}$ by the mapping function ζ, we can compress the importance index ϕ onto $\psi_k \in [0, \infty)$.

$$\psi_k \coloneqq \sum_{j\in\mathcal{M} \text{ s.t. } \zeta(q_j)\in\mathcal{A}_k} \phi_j.$$

Since the calculation is done before monitoring, given $\phi(t_0)$ the initial value of ψ_k can be set as

$$\psi_k(t_0) = \sum_{j\in\mathcal{M} \text{ s.t. } \zeta(q_j)\in\mathcal{A}_k} \phi_j(t_0). \tag{6}$$

We assume that each polygon $\mathcal{A}_k$ is small enough to approximate $\zeta(q_j) \approx \chi_k$ if $\zeta(q_j) \in \mathcal{A}_k$. Once the initial value of ψ_k is defined, it can be updated without ϕ as follows.

$$\begin{aligned}\dot{\psi}_k &= \sum_{j\in\mathcal{M} \text{ s.t. } \zeta(q_j)\in\mathcal{A}_k} \dot{\phi}_j \\ &= \sum_{j\in\mathcal{M} \text{ s.t. } \zeta(q_j)\in\mathcal{A}_k} -\delta \max_{i\in\mathcal{I}} h(p_i, q_j)\phi_j \\ &\approx -\delta \max_{i\in\mathcal{I}} \bar{h}(p_i, \chi_k)\psi_k,\end{aligned}$$

where $\bar{h} : \mathcal{P} \times \mathcal{X} \to [0, 1]$ is the performance function

$$\bar{h}(p_i, \chi) \coloneqq \exp\left(-\frac{\|p_i - \chi\|^2}{2\sigma^2}\right).$$

It is fully expected that the above approximation is reasonable due to the Lipschitz continuity of the function $\bar{h}$. Based on the approximation, the objective function J is also approximated as $J \approx \sum_{k=1}^{l} \psi_k A$.

Under the above approximation, the QP-based controller is then derived in the same way as (5):

$$\begin{aligned}(u_i^*, w_i^*) = \underset{(u_i,w_i)\in\mathcal{U}\times\mathbb{R}}{\arg\min} \ & \epsilon\|u_i\|^2 + |w_i|^2 \\ \text{s.t. } & \bar{\xi}_{1i}^T u_i + \bar{\xi}_{2i} \geq w_i,\end{aligned} \tag{7}$$

where

$$\begin{aligned}\bar{\xi}_{1i} &\coloneqq \sum_{k\in\bar{\mathcal{V}}_i(p)} -\delta\frac{p_i - \chi_k}{\sigma^2} h(p_i, \chi_k)\psi_k A, \\ \bar{\xi}_{2i} &\coloneqq -a\gamma + \sum_{k\in\bar{\mathcal{V}}_i(p)} \left(-\delta^2 h^2(p_i, \chi_k) + a\delta h(p_i, \chi_k)\right)\psi_k A,\end{aligned}$$

and the sets $\bar{\mathcal{V}}_i(p)_{i\in\mathcal{I}}$ is the Voronoi-like partition for $\mathcal{X}$:

$$\bar{\mathcal{V}}_i(p) \coloneqq \{k \in \mathcal{L} \mid \|p_i - \chi_k\| \leq \|p_j - \chi_k\| \ \ \forall j \in \mathcal{I}\}.$$

The QP-based controller (7) keeps track of only l importance indices ψ_k $(k \in \mathcal{L})$ rather than ϕ_j $(j \in \mathcal{M})$. Therefore, its implementation gets much more efficient computationally than (5) by accepting the light approximation $\zeta(q_j) \approx \chi_k$.

B. Safety Certificates

Standard coverage control inherently ensures safety due to territorial partitions in the control algorithm. Meanwhile, since the present algorithm expands the field to be monitored to the 5-D virtual field $\mathcal{Q}_c$, the mapping with ζ of two distant points in $\mathcal{Q}_c$ may be close to each other on $\mathcal{P}$. Accordingly, safety in the proposed algorithm tends to be more critical than standard coverage control.

We thus incorporate the collision avoidance scheme presented in [20] into (7). We denote $p_{i,\text{near}}$ as the nearest drone's position to drone i, and d_{ca} as the acceptable distance between drones. Then, the distance between drone i and the other drones is ensured to be longer than d_{ca} if $b_{\text{ca},i}(p_i) \coloneqq \|p_i - p_{i,\text{near}}\|^2 - d_{\text{ca}}^2$ satisfies $b_{\text{ca},i} \geq 0$. It is proved in [20] that the function $b_{\text{ca},i}$ is a ZCBF for collision avoidance. We can satisfy collision avoidance by just adding the ZCBF condition for $b_{\text{ca},i} \geq 0$ to (7):

$$\begin{aligned}(u_i^*, w_i^*) = \underset{(u_i,w_i)\in\mathcal{U}\times\mathbb{R}}{\arg\min} \ & \epsilon\|u_i\|^2 + |w_i|^2 \\ \text{s.t. } & \bar{\xi}_{1i}^T u_i + \bar{\xi}_{2i} \geq w_i \\ & \left(\frac{\partial b_{\text{ca},i}}{\partial p_i}\right)^T u_i + \beta(b_{\text{ca},i}(p_i)) \geq 0.\end{aligned} \tag{8}$$

VI. SIMULATION RESULTS

We demonstrate our control algorithm through simulations on `ROS` environment, where the quadratic programming of (8) is solved by `CVXOPT`.

The simulation considers three drones ($n = 3$) whose initial positions are selected as $p_1 = [1.0\ 0.2]^T\mathrm{m}$, $p_2 = [-1.0\ -0.2]^T\mathrm{m}$, and $p_3 = [0.0\ 0.5]^T\mathrm{m}$. The speed of drones are restricted by setting input space $\mathcal{U}$ as $[-0.5, 0.5]\mathrm{m/s} \times [-0.5, 0.5]\mathrm{m/s}$. The local controller for each drone is designed to maintain an altitude of $1.0\,\mathrm{m}$. The viewing angle space Θ_h is set to $[-\pi, \pi)$, and Θ_v is set to $[\frac{\pi}{3}, \frac{\pi}{2}]$. We set the target field as a $[-1.0, 1.0]\mathrm{m} \times [-1.0, 1.0]\mathrm{m} \times [0.0, 0.5]\mathrm{m}$ Drones can move through the $3.0\,\mathrm{m} \times 3.0\,\mathrm{m}$ field $\mathcal{P}$ to monitor all position of the target field with all angles. The $\mathcal{Q}_c$ is discretized to a $m = 1.5 \times 10^7$ cells. Each cell is a $0.02\,\mathrm{m} \times 0.02\,\mathrm{m} \times 0.1\,\mathrm{m} \times \frac{\pi}{30}\mathrm{rad} \times \frac{\pi}{30}\mathrm{rad}$ polyhedron whose volume is $A = \frac{4\pi^2}{9} \times 10^{-7}[\mathrm{m}^3\mathrm{rad}^2]$.[2]

The $\mathcal{P}$ is discretized to a $l = 1.0 \times 10^4$ polygons $\mathcal{A}$. Each polygon is $0.03\,\mathrm{m} \times 0.03\,\mathrm{m}$ square. We assign the initial value of importance indices $\phi_j = 1, \forall j \in \mathcal{M}$ and compress it onto $\{\psi_k\}_{k\in\mathcal{L}}$ by (6). We set $\sigma = 0.1$ to let h be almost zero at a drone's viewing range $1.0\,\mathrm{m}$ away. Other parameters are set as $\epsilon = 0.0001$, $d_\mathrm{ca} = 0.5$, $\gamma = 0.1$, $\delta = 5$, and $a = 5$.

We verify from the snapshots in Fig. 2 of the simulation that the primary objective, $J \to 0$, is achieved by the proposed QP-based controller. The color map on the field shows the value of the importance index ψ, where the red region has high importance while the blue one indicates low importance. The drones behave to paint the field with deep blue, that is, they monitor all the field with all angles. Fig. 3(a) shows the time series of J. We see from the figure that J is monotonically decreasing and approaches 0.

Next, we verify that our method meets the secondary objective. Fig. 3(b) shows the time series data of a drone's velocity input $\|u_1\|$ and the sum of importance index near the drone p_1, namely $J_\mathrm{near} := \sum_{k\in\mathcal{L}\ \mathrm{s.t.}\ \|p_1-\chi_k\|<2\sigma} \psi_k A$. This result shows that the speed $\|u_1\|$ tends to be small during the time when J_{near} is high, and vice versa. This suggests that the secondary objective is achieved.

Finally, we verify from the snapshots in Fig. 4 of the simulation that our controller is effective in the angle-aware monitoring. The snapshots are the importance function ϕ of all viewing angles of some points $[-0.5, -0.2, 0], [0, 0, 0], [0.7, 0.7, 0]$ in the target field $\mathcal{B}$ at $t = 40$ s where drones are controlled by (a) our proposed QP-based controller and (b) previous controller proposed in [16]. It is seen that drones monitor these points from almost all viewing angles in (a), while they miss some viewing angles in (b), e.g., the $[0.7, 0.7, 0]$ point from $\theta_\mathrm{v} = \frac{\pi}{2}$. The result exemplifies the benefit of angle-aware coverage control itself. Finally, the average computational time required by the algorithm in Section V at each time instant was $0.0579\,\mathrm{s}$ for a PC with CPU Intel®Core™i7-8650U running at 1.90GHz, 4 Cores, 8 Threads and RAM 7.5 GB, while the PC was down in the original algorithm with 5-D cells. The result exemplifies the benefit of the algorithm in Section V.

VII. CONCLUSIONS

In this paper, we presented a new formulation of angle-aware coverage control for 3-D map reconstruction and a QP-based controller for situation-adaptive actions. To ensure viability, a computationally efficient implementation of the controller was presented. We finally demonstrated the algorithm through simulation with three drones.

[2]We have determined the cells in an ad-hoc manner. More careful selection of the cells might enhance the performance but this exceeds the scope of this paper.

APPENDIX: PROOF OF THEOREM 1

First, the term $\dot{b}_i$ in (4b) can be calculated as

$$\dot{b}_i = \sum_{j\in\mathcal{V}_i(p)} \frac{\partial b_i}{\partial \phi_j}\dot{\phi}_j + \left(\frac{\partial b_i}{\partial p_i}\right)^T u_i. \tag{9}$$

The first term $\sum_{j\in\mathcal{V}_i(p)} \frac{\partial b_i}{\partial \phi_j}\dot{\phi}_j$ in (9) can be calculated as $\frac{\partial b_i}{\partial \phi_j}\dot{\phi}_j = -\delta^2 h(p_i, q_j)^2 \phi_j A$. In addition, the second term can be calculated as

$$\begin{aligned}\left(\frac{\partial b_i}{\partial p_i}\right)^T u_i &= \left(\frac{\partial (I_i - \gamma)}{\partial p_i}\right)^T u_i \\ &= \left(\frac{\partial \sum_{j\in\mathcal{V}_i(p)} \delta h(p_i, q_j)\phi_j A}{\partial p_i}\right)^T u_i \\ &= \left(\sum_{j\in\mathcal{V}_i(p)} -\delta \frac{p_i - \zeta(q_j)}{\sigma^2} h(p_i, q_j)\phi_j A\right)^T u_i.\end{aligned}$$

Furthermore, remarking $\alpha(b_i) = ab_i$, we obtain

$$\alpha(b_i) = -a\gamma + a\sum_{j\in\mathcal{V}_i(p)} \delta h(p_i, q_j)\phi_j A.$$

By substituting the above to (9), we can find that $\dot{b}_i + ab_i = \xi_{1i}^T u_i + \xi_{2i}$ holds. Thus, the problem (4) is equivalent to (5).

REFERENCES

[1] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss, "Uav-based crop and weed classification for smart farming," in *Proceedings of 2017 IEEE International Conference on Robotics and Automation*, 2017, pp. 3024–3031.

[2] M. Rutkauskas, M. Asenov, S. Ramamoorthy, and D. T. Reid, "Autonomous multi-species environmental gas sensing using drone-based fourier-transform infrared spectroscopy," *Opt. Express*, vol. 27, no. 7, pp. 9578–9587, 2019.

[3] H. X. Pham, H. M. La, D. Feil-Seifer, and M. Deans, "A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking," in *Proceedings of 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 6648–6653.

[4] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM: Control Optimisation and Calculus of Variations*, vol. 11, no. 4, pp. 691–719, 2005.

[5] 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.

[6] W. Bentz, T. Hoang, E. Bayasgalan, and D. Panagou, "Complete 3-D dynamic coverage in energy-constrained multi-UAV sensor networks," *Autonomous Robots*, vol. 42, no. 4, pp. 825–851, 2018.

[7] R. Funada, M. Santos, J. Yamauchi, T. Hatanaka, M. Fujita, and M. Egerstedt, "Visual coverage control for teams of quadcopters via control barrier functions," in *Proceedings of 2019 International Conference on Robotics and Automation*, 2019, pp. 3010–3016.

[8] S. Daftry, C. Hoppe, and H. Bischof, "Building with drones: Accurate 3d facade reconstruction using mavs," in *Proceedings of 2015 IEEE International Conference on Robotics and Automation*, 2015, pp. 3487–3494.

[9] S. K. Gupta and D. P. Shukla, "Application of drone for landslide mapping, dimension estimation and its 3d reconstruction," *Journal of the Indian Society of Remote Sensing*, vol. 46, no. 6, pp. 903–914, 2018.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2021.3135466, IEEE Control Systems Letters

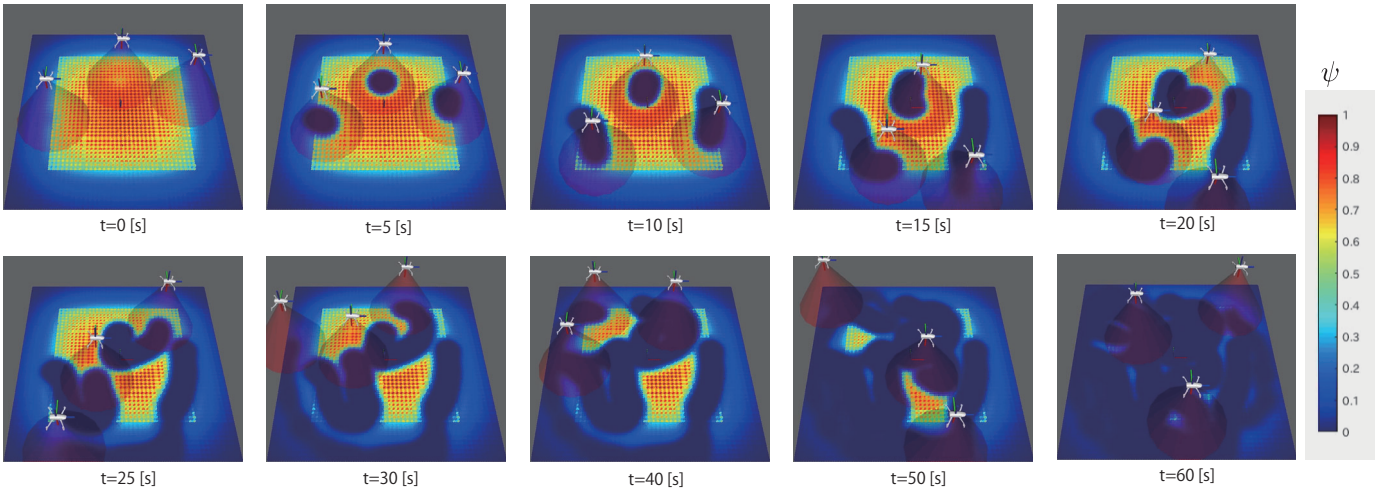
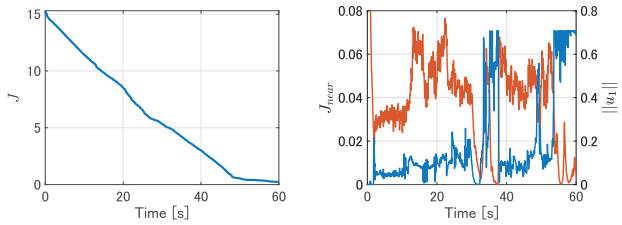


Fig. 2: Evolution of the importance function ψ, where red circle denote the view of drones.



(a) (b)

Fig. 3: (a): Time series of the objective function J. (b): Time series of drone's velocity $\|u_1\|$ (blue), and J_{near} (orange).

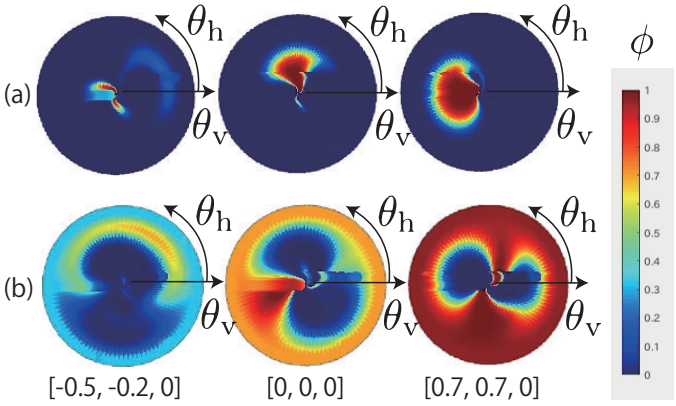


Fig. 4: The importance function ϕ of points $[-0.5, -0.2, 0], [0, 0, 0], [0.7, 0.7, 0]$ in $\mathcal{B}$ at $t = 40$ s: (a) proposed method and (b) method in [16].

[10] M. Mammarella, L. Comba, A. Biglia, F. Dabbene, and P. Gay, "Cooperation of unmanned systems for agricultural applications: A theoretical framework," *Biosystems Engineering*, in Print, 2021.

[11] K. Sugimoto, T. Hatanaka, M. Fujita, and N. Huebel, "Experimental study on persistent coverage control with information decay," in *Proceedings of 2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan*, 2015, pp. 164–169.

[12] J. M. Palacios-Gasós, E. Montijano, C. Sagüés, and S. Llorente, "Distributed coverage estimation and control for multirobot persistent tasks," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1444–1460, 2016.

[13] Y.-W. Wang, M.-J. Zhao, W. Yang, N. Zhou, and C. G. Cassandras, "Collision-free trajectory design for 2-d persistent monitoring using second-order agents," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 545–557, 2020.

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

[15] M. Santos, S. Mayya, G. Notomista, and M. Egerstedt, "Decentralized minimum-energy coverage control for time-varying density functions," in *Proceedings of 2019 International Symposium on Multi-Robot and Multi-Agent Systems*, 2019, pp. 155–161.

[16] H. Dan, T. Hatanaka, J. Yamauchi, T. Shimizu, and M. Fujita, "Persistent object search and surveillance control with safety certificates for drone networks based on control barrier functions," *Frontiers in Robotics and AI*, p. 333, 2021.

[17] M. Khaledyan, A. P. Vinod, M. Oishi, and J. A. Richards, "Optimal coverage control and stochastic multi-target tracking," in *Proceedings of 2019 IEEE 58th Conference on Decision and Control*, 2019, pp. 2467–2472.

[18] Y.-C. Liu, T.-C. Lin, and M.-T. Lin, "Indirect/direct learning coverage control for wireless sensor and mobile robot networks," *IEEE Transactions on Control Systems Technology*, pp. 1–16, 2021.

[19] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[20] M. Egerstedt, J. N. Pauli, G. Notomista, and S. Hutchinson, "Robot ecology: Constraint-based control design for long duration autonomy," *Annual Reviews in Control*, vol. 46, pp. 1–7, 2018.