Persistent Monitoring for Indoor Farming using Static and Mobile Sensors ⋆

Karlisa Priandana * **Made Widhi Surya Atman** ** **Azwirman Gusrialdi** **

* *Department of Computer Science, IPB University, Bogor 16680 Indonesia (e-mail: karlisa@apps.ipb.ac.id).*
** *Faculty of Engineering and Natural Sciences, Tampere University, Tampere 33720 Finland (e-mail: widhi.atman@tuni.fi, azwirman.gusrialdi@tuni.fi)*

Abstract: This paper considers the use of static and mobile sensors (quadrotors) for persistent monitoring in an indoor farming scenario. The focus is on autonomous navigation of the quadrotors to monitor the coverage holes generated by a number of broken static sensors. To that end, a three-layer strategy is proposed by (i) dividing the coverage holes into several region-of-interests (ROIs); (ii) selecting the quadrotor(s) responsible for monitoring each ROI using a novel Modified Ant Colony Optimization; (iii) designing coverage control with time-varying density function for ensuring persistent monitoring. Simulation results demonstrate the efficacy of the proposed strategy in persistently minimizing the coverage holes over time.

Keywords: Persistent monitoring, clustering, automation, task allocation, coverage control, mobile sensors, quadrotors, indoor farming.

1. INTRODUCTION

Food security has become an important issue due to several arising concerns: (i) lack of farms workers, (ii) limited or decreasing agricultural lands, and (iii) increasing demands of food. In response to the above issue, innovative ways of farming have started to emerge by utilizing agricultural automation, such as urban indoor farming and smart vertical farming (Saad et al., 2021). A key aspect of automation in these approaches is persistent plant monitoring which enables real-time analysis and manipulations to maintain good environmental condition for the plantations.

Plant monitoring in indoor farming can be realized using static sensors, mobile sensors or the combination of both. Static sensors, e.g., via wireless sensor network (WSN), can perform monitoring task for a long period (days) before recharging their batteries (Sharma et al., 2019). Static sensors are often fixed into a given infrastructure which makes further changes or reconfiguration of the sensors, e.g., due to broken sensors, to be costly. As a result, broken static sensors will generate coverage holes which degrade the monitoring's quality. On the other hand, mobile sensors based on Unmanned Aerial Vehicles (UAVs) offer robustness and flexibility due to their ability to (autonomously) reconfigure (reposition) themselves (Elmokadem, 2019). Among the UAVs, micro or nano quadrotors are particularly suitable for indoor farming application due to their low cost and small size. However, deploying them for long period is not efficient as most energy is used for the movement of the quadrotors. Combining both static sensors and UAV-based mobile sensors thus has the potential to exploit benefits from both solutions. The work by Popescu et al. (2019) investigate collaborative UAV–WSN scheme from the viewpoint of data communication. Its application for crop monitoring is reported in Popescu et al. (2020). The discussion focuses on the use of a single UAV and multiple ground sensors for effective data transmission at the network level. However, the strategy to guarantee persistent monitoring was not yet discussed.

In this paper, we consider the combination of static and mobile sensors (quadrotors) for persistent monitoring in an indoor farming scenario. In contrast to Elmokadem (2019), the quadrotors act as backup and the main role is to cover the coverage holes due to broken static sensors. The goal is to navigate the quadrotors to minimize the coverage holes over time. To that end, a three-layer strategy is proposed. First, the coverage holes are divided into several regions to minimize quadrotors' flight distance. Then, a novel modified Ant-Colony Optimization is presented to select the quadrotor(s) responsible in each region for balancing the coverage while minimizing its (their) movement. Finally, coverage control algorithm with time-varying density function is then designed to ensure that the selected quadrotors persistently monitor a given region.

The paper is organized as follows. After formally formulating the problem in Section 2, the proposed persistent monitoring strategy is described in Section 3. The performance of the proposed strategy is demonstrated and evaluated via simulations in Section 4. Finally, concluding remarks are given in Section 5.

⋆ The work of A. Gusrialdi was supported by the Academy of Finland under Academy Project decision number 330073. The work of K. Priandana was supported by the Ministry of Education, Culture, Research and Technology Republic of Indonesia grant number 2688/E4/KK.04.03/2021 and 0054/E5/AK.04/2022.

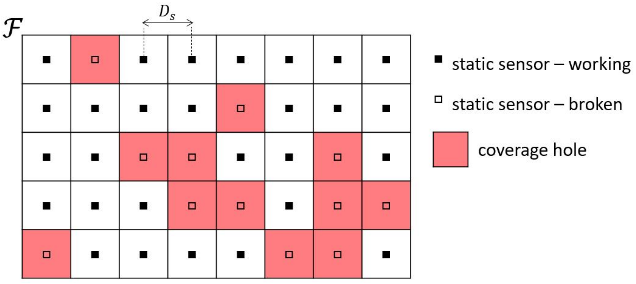

Fig. 1. Illustration of coverage holes due to some broken static sensors in indoor farming

2. PROBLEM STATEMENT

In this paper, we consider an indoor farming scenario where the *field* area is given by a rectangular shaped region (compact set) $\mathcal{F} \subset \mathbb{R}^2$. For monitoring purposes, the field $\mathcal{F}$ are fully covered by static sensors that are deployed in equal distance D_s. Each static sensor is covering a grid, as depicted in Fig. 1. Consider a problem where coverage holes occur inside of $\mathcal{F}$ due to several broken static sensors. Specifically, let the set $\mathcal{M}$ denote a collection of indices of m number of broken static sensors whose locations in the field $\mathcal{F}$ are represented by $h_i \in \mathcal{F} \subseteq \mathbb{R}^2$, $i \in \mathcal{M}$. The coverage holes $\mathcal{H}$ illustrated in Fig. 1 are then defined as

$$\mathcal{H} = \bigcup_{i\in\mathcal{M}} \{q \in \mathcal{F} \mid ||q - h_i||_\infty \le D_s/2\}. \quad (1)$$

Accordingly, we consider n number of autonomous mobile sensors moving in (x, y)-coordinates whose set of identifiers is denoted by $\mathcal{I} = \{1, \ldots, n\}$. The mobile sensors are realized by equipping a fleet of small quadrotors (or similar multirotor UAVs) with sensors to monitor the area in the coverage holes $\mathcal{H}$ without any human's intervention, e.g., in replacing or rearranging the static sensors.

Specifically, each quadrotor i is assumed to be able to sense a subset of area in the field within a fixed sensing radius $R_i > 0$. The sensing region of quadrotor i is defined as

$$\mathcal{B}_i(p_i) = \{q \in \mathcal{F} \mid ||q - p_i|| \le R_i\} \quad (2)$$

where $p_i = [x_i \ y_i]^T \in \mathcal{F}$ represents the position of quadrotor $i \in \mathcal{I}$. Within this paper, let us denote the size of a given region $\mathcal{R}$ as $A(\mathcal{R})$. We can then quantify the sensing capability of each quadrotor as the size of its sensing region, i.e., $A(\mathcal{B}_i) := \pi R_i^2$. Furthermore, it is assumed that the position of each quadrotor is updated according to the kinematic model:

$$\dot{p}_i = u_i \quad (3)$$

where $u_i \subseteq \mathbb{R}^2$ is the velocity input to be designed. As can be observed from (3), in the paper we assume that the quadrotors maintain their altitudes and (x, y)-plane of all quadrotors to be parallel to the field $\mathcal{F}$. In addition, it is assumed that the quadrotor's altitude is embedded into the sensing radius parameter (i.e., $R_i(z_i)$) as in the case of a quadrotor equipped with the downward-facing camera (Funada et al., 2020; Dan et al., 2021).

Hence, in this work we focus our problem on navigating n-number of quadrotors to monitor the coverage holes $\mathcal{H} \subseteq \mathcal{F}$. In particular, the objective is to design control input u_i in (3) for the quadrotors (mobile sensors) in order to minimize the coverage holes at all time, i.e., achieving persistent monitoring over $\mathcal{H}$, while at the same time minimizing the flight distance of each quadrotor.

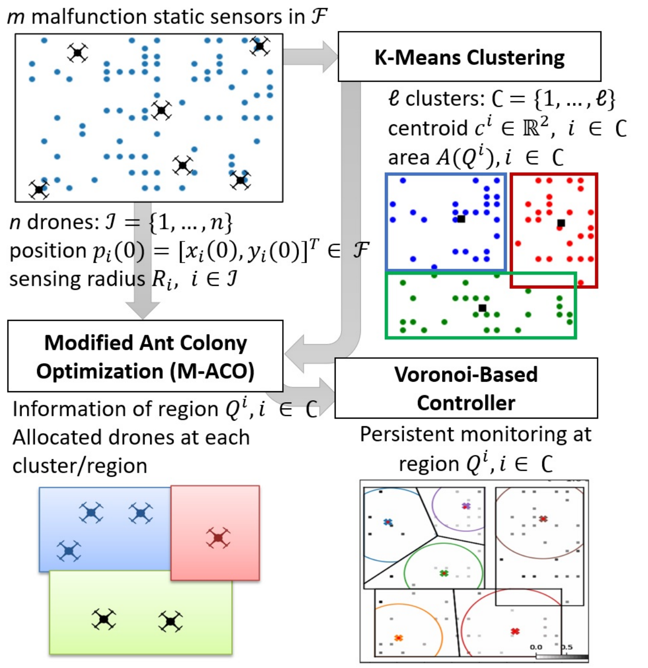

Fig. 2. An overview of the proposed three-layer strategy for indoor farming persistent monitoring

3. PROPOSED AUTOMATED PERSISTENT MONITORING STRATEGY

One of the challenges of utilizing quadrotor for persistent monitoring is its limited flight time due to the battery capacity. One quadrotor may only have a maximum of 20 to 30 minutes flight time (Figliozzi et al., 2018). Therefore, we need to ensure that the total flight distance of each quadrotor is minimized. In this paper, we propose a three-layer strategy focusing on deploying n mobile sensors (quadrotors) into ℓ smaller regions as illustrated in Fig. 2. Here, K-Means clustering is used to divide the coverage holes $\mathcal{H}$ (due to broken static sensors shown in blue points in the top left of the figure) into $\ell \le n$ number of regions, where each region is identified as $\mathcal{Q}^j$, $j \in \mathcal{C} := \{1, \cdots, \ell\}$ with its centroid $c^j \in \mathbb{R}^2$, $j \in \mathcal{C}$. Modified Ant Colony Optimization (M-ACO) is then used to allocate the quadrotors to regions $\mathcal{Q}^i$ by considering the size of each region $A(\mathcal{Q}^j)$, $j \in \mathcal{C}$, the sensing capability of the quadrotor $A(\mathcal{B}_i)$, $i \in \mathcal{I}$, and the distance between the quadrotor's initial position and the region. Finally, the control input to each quadrotor is computed based on voronoi-based coverage algorithm in each assigned region.

3.1 K-Means Clustering

In this subsection, we will describe how K-Means clustering is used to cluster the coverage holes $\mathcal{H}$ as defined in (1) into $\ell \le n$ number of clusters. K-means clustering algorithm is chosen since it is a simple and straight-forward clustering method that can decide whether an object belongs to one cluster or not (Wu and Wu, 2020).

In order to cluster the region $\mathcal{F}$ based on the coverage holes $\mathcal{H}$, we first consider the locations of all broken static sensors, i.e., $h_i,\ \forall i \in \mathcal{M}$. We then set the number of clusters ℓ so that the set of clusters is denoted $\mathcal{C}$. In practice, ℓ is chosen by considering the number of available quadrotors, the number of available charging stations and the size of field $\mathcal{F}$.

The procedure of K-Means clustering is summarized as follows:

(1) First, initialize ℓ number of random points inside $\mathcal{F}$ as the initial centroids of the cluster.
(2) Each point located at $h_i,\ \forall i \in \mathcal{M}$ is then assigned to the closest centroid, by calculating the Euclidean distances between the point and the centroids.
(3) The centroid positions are then updated by taking the average of all points that were assigned to it.
(4) Steps 2 and 3 are repeated until the centroid positions converge, i.e., do not change anymore.

The output of this process is ℓ groups of broken sensors with each group $j \in \mathcal{C}$ is denoted by $\mathcal{M}^j \subseteq \mathcal{M}$ and its centroid c^j. Note that in this work, we approximate the region to be monitored in each cluster as a rectangle by taking the outer points at x-axis and y-axis with additional $D_s/2$ offset. This ensures that the combination of the regions overlap the coverage holes $\mathcal{H}$, i.e., $\cup_{j\in\mathcal{C}} \mathcal{Q}^j \supseteq \mathcal{H}$.

Note that when the regions are overlapping, then the broken sensors in that overlapping regions should be assigned to only one region (e.g., to the one with more broken sensors around the overlapping regions). This will ensure no collision between the quadrotors from different regions as only the quadrotors in one region know about the broken sensors in the overlapping region.

3.2 Modified Ant Colony Optimization

After dividing the coverage holes into ℓ clusters, in this subsection we propose a novel Modified Ant Colony Optimization (M-ACO) for task allocation, that is to assign available quadrotors to one of the clusters. Specifically, the M-ACO utilizes the following information

(1) the size of each task/region, $A(\mathcal{Q}^j),\ j \in \mathcal{C}$;
(2) sensing capability of each quadrotor, $A(\mathcal{B}_i),\ i \in \mathcal{I}$;
(3) distance between the quadrotor's initial position and the centroid of the region, $d_{ij} = \|p_i(0) - c^j\|,\ i \in \mathcal{I},\ j \in \mathcal{C}$.

The proposed M-ACO algorithm is based on the original Ant-Colony Optimization (ACO) algorithm (Dorigo and Caro, 1999), one of the meta-heuristic approaches for solving e.g., Traveling Salesman Problem by mimicking the behavior of ants in finding their food using pheromones to communicate and left tracks for other ants. The goal of the original ACO algorithm is to find a shortest path from the nest to food. The use of ACO for Task Allocation was introduced as Collective Path Ant Colony Optimization (Wang et al., 2013) where ACO was modified to find the most efficient agent coalition in a region in a multiple agents and multiple capabilities scenario. This is different with our case, where multiple agents are to be assigned to a region in a multiple regions (tasks) scenario.

Our proposed M-ACO method aims at finding the most suitable agents (quadrotors) to cover the tasks (regions). The goal is to obtain an equal distribution of quadrotors in covering the regions of broken sensors and optimizing the battery utilization of the quadrotors, i.e., minimizing the distance between the allocated quadrotors and the selected region. To that end, both the agents and the tasks are represented as nodes while ants are the computational units to find the most optimum coalition of agents to cover each region.

Initially, each ant randomly chooses a task and an agent using the following probability function:

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha [\eta_{ij}^k(t)]^\beta}{\sum_{a,b\subset allowed_k} [\tau_{ab}(t)]^\alpha [\eta_{ab}^k(t)]^\beta}; & i,j \in allowed_k \\ 0; & \text{otherwise} \end{cases} \tag{4}$$

where p_{ij}^k is the probability of ant k in pairing agent i with task/region j, τ_{ij} is the pheromone concentration at path ij, η_{ij}^k is the heuristic function of path ij for ant k, α and β denote the importance of the pheromone and the heuristic value respectively. Here, $allowed_k$ is the set of agents and tasks that can be chosen by ant k, i.e., quadrotors that have not been allocated to any region and also regions that still have "uncovered" area. The function η_{ij}^k is defined by:

$$\eta_{ij}^k = \frac{A(\mathcal{Q}^j) - \sum_{k\in\mathcal{I}^j} A(\mathcal{B}_k)}{d_{ij}} \tag{5}$$

where $\mathcal{I}^j \subset \mathcal{I}$ is the set of quadrotors already selected for region $j \in \mathcal{C}$. This heuristic function are defined such that the ants tend to pair the region that has not been fully covered yet and then pair it with the closest quadrotor.

In contrast to the original ACO (Dorigo and Caro, 1999) and CP-ACO (Wang et al., 2013), here one agent can only be chosen once. Moreover, the region can only be chosen if its size, $A(\mathcal{Q}^j)$, is larger than the total capability of the agents that are already selected for that task, $\sum_{k\in\mathcal{I}^j} A(\mathcal{B}_k)$. The iteration of pairing the quadrotor and the region continues until all quadrotors are allocated to the regions, or until all the regions are fully covered, which is defined as one tour.

After one tour, each ant updates the pheromones of the chosen paths according to

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t), \tag{6}$$

$$\Delta\tau_{ij}^k = \begin{cases} P\epsilon^k; & \text{if ant } k \text{ select agent } i \text{ for task } j \\ 0; & \text{otherwise} \end{cases} \tag{7}$$

where τ_{ij} is the pheromone concentration for pairing agent i and task j, ρ is the pheromone evaporation coefficient, $\Delta\tau_{ij} = \sum_N \Delta\tau_{ij}^k$ is the change of pheromone concentration for pair ij caused by N number of ants, P is the pheromone strength, and ϵ^k is the efficiency factor of ant k in one tour, defined as

$$\epsilon^k = \frac{\sum_{j\in\mathcal{C}} \left(A(\mathcal{Q}^j) - \left|\sum_{k\in\mathcal{I}^j} A(\mathcal{B}_k) - A(\mathcal{Q}^j)\right|\right)}{\sum_{j\in\mathcal{C}} \left(\sum_{k\in\mathcal{I}^j} d_{kj}\right)} \tag{8}$$

It can be observed that the pheromone concentration is iteratively updated by minimizing the following two costs: (i) the difference between the size of the region and the total sensing capability of the selected quadrotors; (ii) the distance between the paired quadrotors and regions. At the end of iteration, the chosen solution is the ant tour which yields a maximum value of efficiency factor ϵ^k.

To this end, the output of M-ACO is the set of index $\mathcal{I}^j \subseteq \mathcal{I}$, $j \in \mathcal{C}$ which ensures a distinct selection of quadrotors to each given region, i.e., $\mathcal{I}^j \cap \mathcal{I}^k = \emptyset$ for $j \neq k$. The information of each region $\mathcal{Q}^j$, $j \in \mathcal{C}$, the locations of broken static sensors within (i.e., h_i, $\forall i \in \mathcal{M}^j$), and the selected quadrotors (i.e., $\mathcal{I}^j$) are then being used to compute the voronoi-based coverage control algorithm.

3.3 Voronoi-Based Coverage Control Algorithm

In this subsection, we describe the formulation of control input to ensure all the assigned quadrotors persistently cover their designated field by adopting the idea from Hübel et al. (2008); Sugimoto et al. (2015) and Dan et al. (2021). As the quadrotors are designated to cover (move within) the area to which they are initially assigned, in the remaining of the discussion we focus on a single region of $\mathcal{Q}^j$ and the quadrotors in the set $\mathcal{I}^j$ where $j \in \mathcal{C}$.

In order to develop coverage control algorithm for the fleet of quadrotors, we first model the important areas to be monitored/covered (i.e., coverage holes) as a density function $\hat{\phi}^j : \mathcal{F} \to [0, 1]$ which is formulated using a mixture of Gaussian functions:

$$\hat{\phi}^j(q) := \max_{i \in \mathcal{M}^j} \left(\exp\left[-\frac{1}{2}(q - h_i)^T \Sigma^{-1} (q - h_i) \right] \right), \quad (9)$$

where Σ is the covariance matrix.

The formulation above considers identical gaussian function over a maximum function which ensures equal importance of all the broken static sensors. However, it may come at a price of non-smoothness between two important points. Note that throughout extensive simulations we observe no practical issues with this formulation. If necessary, a smoothing procedure using diffussion process as in Perona and Malik (1990) can be further implemented.

Next, consider the collocations of p_i for all $i \in \mathcal{I}^j$ as p^j. Let us introduce the Voronoi partition of each region $\mathcal{Q}^j$ from Cortés et al. (2005), namely the collection of the sets $\{\mathcal{V}_i(p^j)\}_{i \in \mathcal{I}^j}$ defined as

$$\mathcal{V}_i(p^j) = \{q \in \mathcal{Q}^j \mid ||q - p_i|| \leq ||q - p_j||, \ \forall j \in \mathcal{I}^j \backslash \{i\}\}. \quad (10)$$

Moreover, let us define the feasible sensing area $\mathcal{S}_i(p^j)$ as

$$\mathcal{S}_i(p^j) := \mathcal{B}_i(p_i) \cap \mathcal{V}_i(p^j).$$

It is shown in Cortés et al. (2005) that the set $\mathcal{S}_i(p^j)$ depends only on the position of quadrotors that lie within radius $2R_i$ from p_i. Therefore, $\mathcal{S}_i(p^j)$ can be computed in a distributed fashion, e.g., by allowing quadrotors to exchange position information within $2R_i$ radius.

In the remaining discussion, let us consider a practical case when the total sensing capabilities of the assigned quadrotors is much less than the total area to be covered which requires the quadrotors to periodically visit the broken sensors locations. To that end, we consider the following time-varying density function $\phi^j : \mathcal{Q}^j \times \mathbb{R}_{\geq t_0} \to [0, 1]$, updated according to

$$\frac{\mathrm{d}\phi^j(q,t)}{\mathrm{d}t} = \begin{cases} -\underline{\delta}\phi^j(q,t), & \text{if } q \in \cup_{i \in \mathcal{I}^j} \mathcal{S}_i \\ \bar{\delta}(\hat{\phi}^j(q) - \phi^j(q,t)) & \text{otherwise} \end{cases} \quad (11)$$

with $\bar{\delta}, \underline{\delta} > 0$. The update rule (11) implies that the importance of a point being monitored by a quadrotor is decreasing with rate $\underline{\delta}$, and will then be increased with rate $\bar{\delta}$ if it is left unmonitored and thus requiring the quadrotor to revisit that point in order to maintain persistent monitoring of the coverage holes. Note that the parameters $\bar{\delta}, \underline{\delta}$ are assigned by the designer by taking into account the characteristic of the sensor attached to the quadrotor. For example, if the sensor requires more time to take measurement, then $\underline{\delta}$ can be set to be small.

The control input u_i for all $i \in \mathcal{I}^j$ can then be computed based on the gradient ascent algorithm to maximize the following objective function

$$J(p^j, t) := -\sum_{i \in \mathcal{I}^j} \int_{\mathcal{S}_i} ||q - p_i||^2 \phi^j(q,t)\, \mathrm{d}q. \quad (12)$$

To that end, let us then consider a (partially) distributed computation given by

$$\frac{\partial J(p^j, t)}{\partial p_i} = 2\text{mass}(\mathcal{S}_i(p^j))(\text{cent}(\mathcal{S}_i(p^j)) - p_i) \quad (13)$$

where

$$\begin{aligned} \text{mass}(\mathcal{S}_i(p^j)) &:= \int_{\mathcal{S}_i} \phi^j(q,t)\, \mathrm{d}q \\ \text{cent}(\mathcal{S}_i(p^j)) &:= \frac{1}{\text{mass}(\mathcal{S}_i(p^j))} \int_{\mathcal{S}_i} q\phi^j(q,t)\, \mathrm{d}q \end{aligned} \quad (14)$$

During initial deployment, the position of each quadrotor may be outside of the designated region $\mathcal{Q}^j$. Thus, a proportional control is introduced for the quadrotor to navigate towards the centroid of $\mathcal{Q}^j$ (i.e., c^j) until it enters the region $\mathcal{Q}^j$. To summarize, the computation of u_i for $i \in \mathcal{I}^j$ is given by

$$u_i = \begin{cases} \dfrac{\partial J(p^j, t)}{\partial p_i}, & \text{if } p_i \in \mathcal{Q}^j \\ \gamma(c^j - p_i) & \text{otherwise} \end{cases} \quad (15)$$

where $\gamma > 0$ is the proportional gain.

The Voronoi-based control law implicitly prevents collision between the quadrotors as discussed in Gusrialdi et al. (2009). It is worth to note that collision avoidance can be explicitly incorporated in (15) by using e.g., artificial potential field method (Gusrialdi and Yu, 2014) or control barrier function approach (Dan et al., 2021).

Finally, in practice the computation of the density update in (11) will be be performed by a central system for each region j, since each quadrotor hardly knows if other quadrotors have visited each $q \in \mathcal{Q}^j$. On the other hand, given the information of $\mathcal{Q}^j$, c^j, and $\phi^j(q,t)$ in $\mathcal{S}_i$ from region j's central system, each quadrotor $i \in \mathcal{I}^j$ can distributively compute u_i.

4. SIMULATION

In this section, we demonstrate and evaluate the proposed three-layer strategy in a simulation. To this end, we consider a field of size $\mathcal{F} = [0, 20]$ m $\times$ $[0, 20]$ m, equipped with some static sensors where the inter-sensor distance (grid size) is $D_s = 1$ m. We set the number of broken sensors $m = 80$ where the positions of the broken static sensors (coverage holes) are shown in Fig. 3. First, the coverage holes is divided into several clusters/regions using K-means clustering where we set the number of clusters $\ell = 3$. The resulting regions $\mathcal{Q}^j$ are depicted in Fig. 3.

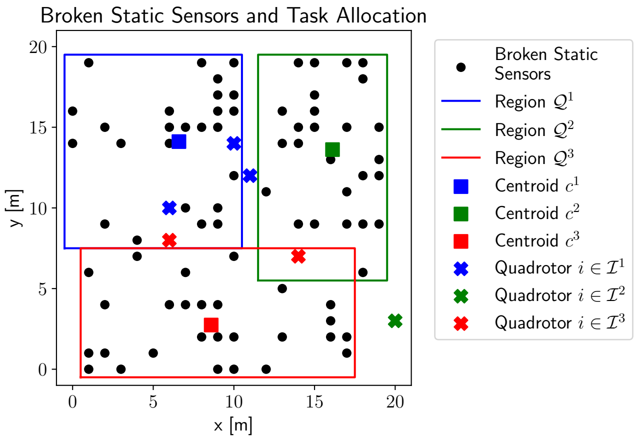

Fig. 3. Locations of the broken static sensors, division of regions by K-Means clustering with $\ell = 3$, and quadrotors allocation by the M-ACO algorithm.

Table 1. Initial position, sensing radius, and sensing capability of each quadrotor

$i \in \mathcal{I}$	1	2	3	4	5	6
$x_i(0)$ [m]	11	6	6	14	10	20
$y_i(0)$ [m]	12	8	10	7	14	3
R_i [m]	4.37	3.99	3.57	5.05	3.09	4.72
$A(\mathcal{B}_i)$ [m^s]	60	50	40	80	30	70

Table 2. The parameters of M-ACO

Parameter	Symbol	Value
Number of ants	N	10
Number of iterations (tours)	$iter$	1000
Initial pheromone value	$P(0)$	1
Pheromone strength	P	1
Importance of pheromone	α	1
Importance of heuristic function	β	1
Pheromone evaporation coefficient	ρ	0.2

Next, we assume that the number of quadrotors to be deployed is $n = 6$ whose initial positions (e.g., correspond to locations of the charging stations), sensing radius, and sensing capabilities are summarized in Table 1. Then, we utilize the M-ACO with parameters depicted in Table 2 to allocate individual quadrotor to each region. The results of M-ACO task allocation are also shown in Fig. 3, where quadrotor(s) of the same color with a region's centroid are allocated to that particular region. The allocation results showed that: (i) region $\mathcal{Q}^1$ with size $A(\mathcal{Q}^1) = 132m^2$ is paired with quadrotors 1, 3 and 5 with total sensing capability of $\sum_{k\in\mathcal{I}^1} A(\mathcal{B}_k) = (60 + 40 + 30)m^2 = 130m^2$; (ii) region $\mathcal{Q}^2$ with size $A(\mathcal{Q}^2) = 112m^2$ is paired with quadrotor 6 with sensing capability of $70m^2$; and (iii) region $\mathcal{Q}^3$ with size $A(\mathcal{Q}^3) = 136m^2$ is paired with quadrotors 2 and 4 with total sensing capability of $\sum_{k\in\mathcal{I}^3} A(\mathcal{B}_k) = (50 + 80)m^2 = 130m^2$. These results showed that the proposed M-ACO can allocate the quadrotors to the region equally, by considering the region's size, the quadrotor's sensing capabilities and the distance between the quadrotor's initial position and the region's centroid.

After allocating the quadrotors to each region Q^j, we proceed with discussing the voronoi-based coverage control algorithm. The density function $\hat{\phi}$ is computed as in (9) with covariance matrix $\Sigma = D_s \, \mathrm{diag}(0.5, 0.5)$. The time varying density functions ϕ^j for all $j \in \mathcal{C}$ are updated using $\bar{\delta} = 0.05$ and $\underline{\delta} = 1.0$. The movement of the quadrotors based on the control input u_i in (15) is illustrated in Fig. 4.

Initially, three quadrotors (shown in brown, blue and orange markers in Fig. 4) are outside of their designated region and moving towards the centroid of the region. Once inside, they cooperatively cover the important area in the region modeled by the density function. It is apparent by observing the movement of quadrotor 6 (shown in brown marker) that the designed controller ensures the quadrotors to persistently cover the areas (i.e., locations of broken static sensors) by repeatedly revisiting past observed areas. The grey scale color map on Fig. 4 illustrates the density function on each broken sensor's position, where darker region has high density while lighter color region means low density. From Fig. 4, we can observe that the proposed controller ensures that the density value on the broken sensors are being kept low, which directly linked to a good coverage performance.

Note that the case of fully coverage on all broken sensors at the same time are denoted by zero density function over the whole region. However, this cannot be achieved in the current simulation as the total sensing capabilities of the assigned quadrotors is less than the size of the area to be covered. To evaluate the performance of the coverage control algorithm, let us define the ratio of the currently required region to be covered versus the total desired coverage within a given region $\mathcal{Q}^j$ as

$$\zeta^j(t) := \frac{\int_{\mathcal{Q}^j} \phi^j(q,t)\,\mathrm{d}q}{\int_{\mathcal{Q}^j} \hat{\phi}^j(q,t)\,\mathrm{d}q}. \tag{16}$$

The plot of the $\zeta^j(t)$ for the simulation is shown in Fig. 5 for a longer time duration than the snapshots in Fig. 4. It can be observed that the proposed control algorithm minimizes the coverage holes indicated by low value on density function. Hence, persistent monitoring is ensured.

5. CONCLUSION AND FUTURE WORK

A network of quadrotors is utilized for achieving persistent monitoring in indoor farming under the presence of broken static sensors. To this end, a novel three-layer strategy is proposed for navigating the quadrotors to minimize the coverage holes generated by the broken static sensors. Specifically, after dividing the coverage holes (broken static sensors) into a number of regions, a novel M-ACO is then proposed to allocate the available quadrotors to each region for ensuring balanced coverage between the regions and minimizing flight distances. A coverage control with time-varying density function is then designed to control the movement of quadrotors in ensuring persistent monitoring. Simulation results demonstrate that the proposed strategy persistently minimizes the coverage holes over time. In the future, we aim to evaluate the proposed strategy using a robotic experimental testbed and also to incorporate charging's scheduling of the quadrotors.

REFERENCES

Cortés, J., Martínez, S., and Bullo, F. (2005). Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM: Control, Optimisation and Calculus of Variations*, 11(4), 691–719.

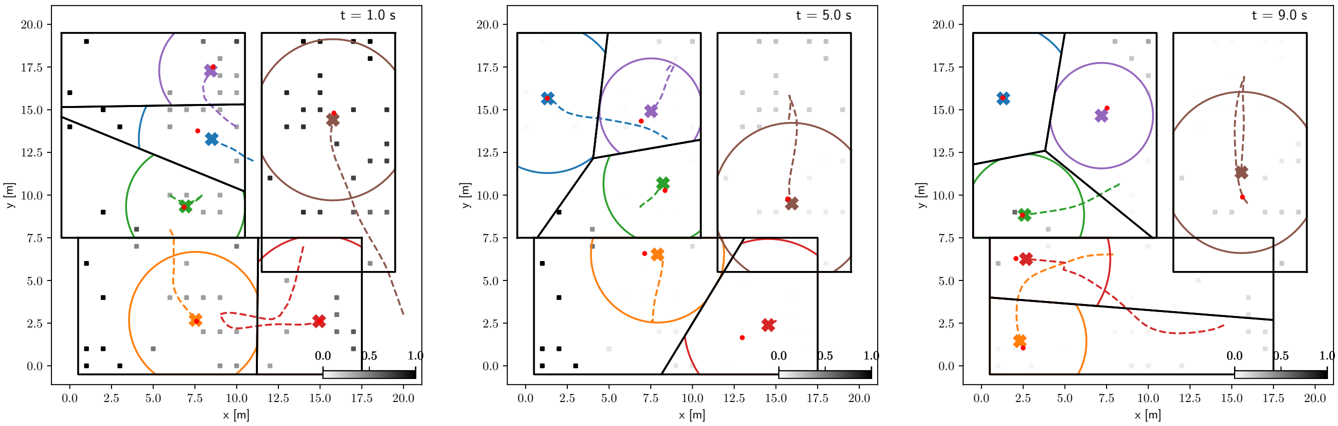

Fig. 4. Snapshots of quadrotors' movements. The small squares illustrate the density function at broken static sensors whose value ranging from 0 (white) to 1 (black). Each quadrotor with identifier 1 to 6 is respectively shown as color: blue, orange, green, red, purple, and brown. The 'x' mark and the dashed lines depict quadrotors' current position and trajectories over the past 4 seconds. The colored solid line depicts each quadrotor's sensing range within its own voronoi partition (solid black lines within each $\mathcal{Q}^j$). The video of the simulation can be viewed in `https://youtu.be/KFgY-qIJx5Y` .

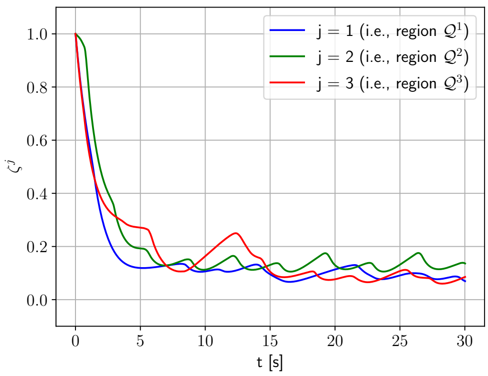

Fig. 5. Time series plot of $\zeta^j(t)$ for each cluster.

Dan, H., Hatanaka, T., Yamauchi, J., Shimizu, T., and Fujita, M. (2021). Persistent Object Search and Surveillance Control With Safety Certificates for Drone Networks Based on Control Barrier Functions. *Frontiers in Robotics and AI*, 8, 740460.

Dorigo, M. and Caro, G.D. (1999). Ant colony optimization: A new meta-heuristic. In *Proceedings of the Congress on Evolutionary Computation*, 1470–1477.

Elmokadem, T. (2019). Distributed coverage control of quadrotor multi-uav systems for precision agriculture. *IFAC-PapersOnLine*, 52(30), 251–256.

Figliozzi, M., Tucker, C., and Polikakhina, P. (2018). Drone deliveries logistics, efficiency, safety and last mile trade-offs. In *Proceedings 7th International Conference on Information Systems, Logistics and Supply Chain*.

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

Gusrialdi, A., Hirche, S., Asikin, D., Hatanaka, T., and Fujita, M. (2009). Voronoi-based coverage control with anisotropic sensors and experimental case study. *Intelligent Service Robotics*, 2(4), 195–204.

Gusrialdi, A. and Yu, C. (2014). Exploiting the use of information to improve coverage performance of robotic sensor networks. *IET Control Theory & Applications*, 8(13), 1270–1283.

Hübel, N., Hirche, S., Gusrialdi, A., Hatanaka, T., Fujita, M., and Sawodny, O. (2008). Coverage Control with Information Decay in Dynamic Environments. *IFAC Proceedings Volumes*, 41(2), 4180–4185.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7), 629–639.

Popescu, D., Stoican, F., Stamatescu, G., Chenaru, O., and Ichim, L. (2019). A survey of collaborative uav–wsn systems for efficient monitoring. *Sensors*, 19(21), 4690.

Popescu, D., Stoican, F., Stamatescu, G., Ichim, L., and Dragana, C. (2020). Advanced uav–wsn system for intelligent monitoring in precision agriculture. *Sensors*, 20(3), 817.

Saad, M.H.M., Hamdan, N.M., and Sarker, M.R. (2021). State of the art of urban smart vertical farming automation system: Advanced topologies, issues and recommendations. *Electronics*, 10(12).

Sharma, H., Haque, A., and Jaffery, Z.A. (2019). Maximization of wireless sensor network lifetime using solar energy harvesting for smart agriculture monitoring. *Ad Hoc Networks*, 94, 101966.

Sugimoto, K., Hatanaka, T., Fujita, M., and Huebel, N. (2015). Experimental study on persistent coverage control with information decay. In *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 164–169.

Wang, L., Wang, Z., Hu, S., and Liu, L. (2013). Ant colony optimization for task allocation in multi-agent systems. *China Communications*, 10(3), 125–132.

Wu, Z. and Wu, Z. (2020). An enhanced regularized k-means type clustering algorithm with adaptive weights. *IEEE Access*, 8, 31171–31179.