

PANDA: Placement of Unmanned Aerial Vehicles Achieving 3D Directional Coverage

Weijun Wang* Haipeng Dai* Chao Dong† Xiao Cheng* Xiaoyu Wang* Guihai Chen* Wanchun Dou*

*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210024, CHINA

†Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210007, CHINA

Emails: {haipengdai, gchen, wcdou}@nju.edu.cn, dch999@gmail.com, {weijunwang, xiaocheng, xiaoyuwang}@smail.nju.edu.cn

Abstract—This paper considers the fundamental problem of Placement of unmanned Aerial vehicles achieviNg 3D Directional coverAge (PANDA), that is, given a set of objects with determined positions and orientations in a 3D space, deploy a fixed number of UAVs by adjusting their positions and orientations such that the overall directional coverage utility for all objects is maximized. First, we establish the 3D directional coverage model for both cameras and objects. Then, we propose a Dominating Coverage Set (DCS) extraction method to reduce the infinite solution space of PANDA to a limited one without performance loss. Finally, we model the reformulated problem as maximizing a monotone submodular function subject to a matroid constraint, and present a greedy algorithm with $1 - 1/e$ approximation ratio to address this problem. We conduct simulations and field experiments to evaluate the proposed algorithm, and the results show that our algorithm outperforms comparison ones by at least 75.4%.

I. INTRODUCTION

Camera sensor network has attracted great attention in recent years as it provides detailed data of environment by retrieving rich information in the form of images and videos [1], [2]. It has found a wide range of applications, such as surveillance, traffic monitoring, and crowd protection, *etc.* For some temporary situations, such as assembly, concerts, matches, and outdoor speeches, establishing stationary camera sensor network in advance may cost too much time and money, and may be inconvenient or even impossible. Fortunately, the development of Unmanned Aerial Vehicle (UAV) technology in the past few years [3], [4] offers a promising way to address this issue. With the low-cost and agile UAVs, camera sensor network can be deployed dynamically to capture real-time, reliable and high-quality images and videos. For example, *DJI Phantom 4* UAV can fly at 72 km/h , rise at 6 m/s , swerve at $250^\circ/\text{s}$, and provide $2K$ real-time images and videos [5].

To guarantee high quality of monitoring, UAVs should jointly adjust their coordinates and the orientations of their cameras to cover objects at or near their frontal view [6]. Though some sensor placement methods have emerged to study directional sensor or camera sensor coverage problem [7]–[19], none of them considers that sensors and objects are placed in 3D environment. Although a few camera sensor coverage methods consider 3D environment [20]–[22], almost all of them assume that all objects are still distributed on 2D plane. However, in many scenarios, the objects cannot be served as distributed on 2D plane. For example, in face

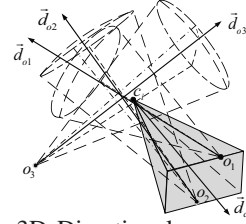


Fig. 1: 3D Directional coverage model

recognition [23] UAVs won't be more than 15 m far and 5 m high from the human faces, which implies the height of humans cannot be ignored, especially they are on different floors. Another example, in crop inspection [24] UAVs need to monitor crop closely to capture the details of them, hence the height of crop also cannot be neglected.

In this paper, we study the problem of Placement of Unmanned Aerial Vehicles achieviNg 3D Directional coverAge (PANDA). In our considered scenario, some objects are distributed in 3D space with known facing direction, and we have a given number of UAVs to deploy in the free 3D space whose cameras can freely adjust their orientation. The practical 3D directional coverage model for cameras and objects is established as Figure 1. The coverage space of camera is modeled as a straight rectangle pyramid and the efficient coverage space of object is modeled as a spherical base cone, which are both fundamentally different from previous work. Moreover, we define *directional coverage utility* to characterize the effectiveness of directional coverage for all objects. Formally, given a fixed number of UAVs and a set of objects in the space, PANDA problem is to deploy the UAVs in the 3D free space, *i.e.*, to determine their coordinates and orientations (the combinations of which we define as *strategies*), such that the directional coverage utility for all objects is maximized.

We face two main technical challenges to address PANDA. The first challenge is that the problem is essentially continues and nonlinear. The problem is continues because both coordinates and orientations of UAVs are continuous values. It is also nonlinear because the angular constraints for both camera model and object model, and the constraint of pitching angle of camera in objective function. The second challenge is to develop an approximation algorithm which needs to bound the performance gap to the optimal solution.

To address these two challenges, first, we propose a Dominating Coverage Set extraction method to reduce the continuous search space of strategies of UAVs to a limited number of strategies without performance loss. Hence, PANDA problem becomes to select a fixed number of strategies from a set of candidate ones to maximize the overall object directional coverage utility, which is discrete, and we address the first challenge. Then, we prove that the reformulated problem falls into the scope of the problem of maximizing a monotone submodular function subject to a matroid constraint, which allows a greedy algorithm to achieve a constant approximation ratio. Consequently, we address the second challenge.

II. RELATED WORKS

Directional sensor coverage and camera sensor coverage.

Directional sensor coverage works can be classified into object coverage and area coverage, whose respective goal is maximizing the number of covered objects [7], [25] and area coverage ratio [8], [26], but most of them do not take the objects' facing direction into account [10]. In addition, although most of camera sensor coverage works consider objects' facing direction [9], [27], [28] and address directional coverage problems under various scenarios, *i.e.*, barrier coverage [27], they only consider 2D sector model which is also the assumption of most of directional sensor coverage works.

3D camera sensor coverage. There exists a few works focusing on camera sensor coverage in 3D environment. Ma *et al.* in [20] proposed the first 3D camera coverage model and developed an algorithm for area coverage on 2D plane with the projecting quadrilateral area of 3D camera coverage model. Based on this model, Yang *et al.* in [29] introduced coverage correlation model of neighbor cameras to decrease the number of cameras. Han *et al.* in [30] and Yang *et al.* in [31] took energy and storage of camera sensors into account and proposed high-efficient resource utility coverage algorithm. Si *et al.* in [32] considered the intruders' facing direction and the size of face in barrier coverage. Hosseini *et al.* in [33] addressed the problem of camera selection and configuration problem for object coverage by binary integer programming solution. Li *et al.* in [21] and Peng *et al.* in [22], [34] established a more practical 3D camera coverage model, and studied three area coverage problems based on this model. However, most of above works simply use 3D camera coverage model to address those problems whose covered objects or area are only distributed on 2D plane, and thus none of them can solve our problem.

III. PROBLEM STATEMENT

Suppose we have M objects $O = \{o_1, o_2, \dots, o_M\}$ to be monitored in a 3D free space, each object o_j has a known orientation, which is denoted by a vector $\vec{d}_{o_j}$. We also have N UAVs $C = \{c_1, c_2, \dots, c_N\}$ equipped with cameras which can hover anywhere in the 3D free space. Because of hardware limitation, their cameras can only rotate in the vertical plane. However, this limitation has no influence to the coverage orientation, because UAV can hover in the air and rotate itself

TABLE I: Notations

Symbol	Meaning
c_i	UAV i , or its 3D coordinate
o_j	Object j to be monitored, or its 3D coordinate
N	Number of UAVs to be deployed
M	Number of objects to be monitored
$\vec{d}_{ci}$	Orientation of camera of UAV i
γ	Pitching angle of camera
γ_{min}	Minimum pitching angle
γ_{max}	Maximum pitching angle
α	Horizontal offset angles of the FoV around $\vec{d}_{ci}$
β	Vertical offset angles of the FoV around $\vec{d}_{ci}$
$\vec{d}_{oj}$	Orientation of object o_j
θ	Efficient angle around $\vec{d}_{oj}$ for directional coverage
D	Farthest sight distance of camera with guaranteed monitoring quality

Edge/Plane/Point	
$c_i P \cap \overline{BD}$	P'
$c_i P \cap \overline{BD}$	O'
$c_i P \cap \overline{BD}$	C'
$c_i P \cap \overline{BD}$	O''

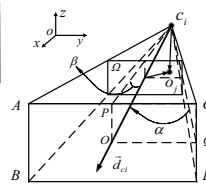


Fig. 2: Camera model

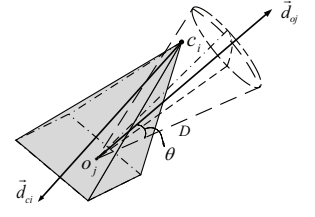


Fig. 3: Directional coverage model

to face any horizontal orientation. By a little abuse of notation, c_i and o_j also denote the coordinate of UAV and object. Table I lists the notations we use in this paper.

A. Camera Model

More complicated than previous sector model of 2D directional sensor, camera model in 3D environment needs to be modeled as a straight rectangular pyramid as shown in Figure 2. Due to cameras only can rotate in vertical plane, edge $\overline{AC}$ and $\overline{BD}$ are always parallel to the ground.

We use a 5-tuple $(c_i, \vec{d}_{ci}, \gamma, \alpha, \beta)$ to denote the camera model. As illustrated in Figure 2, point c_i denotes the coordinate of UAV c_i , its value is (x_0, y_0, z_0) . Vector $\vec{d}_{ci}$ denotes the orientation of c_i 's camera which is perpendicular to undersurface $ABCD$ and its unit vector equals to (x_1, y_1, z_1) . Point O is the centre of rectangle $ABCD$ and the distance $|\vec{c_i O}| = D$, where D is the farthest distance from camera which can guarantee the quality of monitoring of every object on $ABCD$. Thus, the coordinate of point O is $(x_0 + Dx_1, y_0 + Dy_1, z_0 + Dz_1)$. Clearly, we can mathematically express plane $ABCD$ as

$$\vec{d}_{ci} \cdot \begin{pmatrix} x - (x_0 + Dx_1) \\ y - (y_0 + Dy_1) \\ z - (z_0 + Dz_1) \end{pmatrix} = 0. \quad (1)$$

Connecting point c_i to midpoint P of $\overline{AC}$ and Q of $\overline{CD}$ respectively, we can get plane $c_i OP$ and $c_i OQ$. As cameras only can rotate in vertical field, plane $c_i OP$ is parallel to z -axis. Thus, plane $c_i OP$ can be expressed as

$$-y_1x + x_1y + x_0y_1 - y_0x_1 = 0. \quad (2)$$

As shown in Figure 2, $\overline{OQ} \perp \overline{OP}$, $\overline{OQ} \perp \overline{Oc_i}$, thus $\overline{OQ} \perp c_iOP$. By Equation (2) and $\angle Oc_iQ$ equals to the horizontal offset angle α , so $|\overline{OQ}| = D \cdot \tan \alpha$. Thus, we can obtain

$$\overrightarrow{OQ} = D \cdot \tan \alpha \cdot (-y_1, x_1, 0). \quad (3)$$

Similar, $\overline{OP} \perp c_iOQ$, $\angle Oc_iP$ equals to the vertical offset angle β , then we have $|\overline{OP}| = D \cdot \tan \beta$. Combine the equation of plane c_iOP , vector $\overrightarrow{OP}$ can be obtained as

$$\overrightarrow{OP} = (x_1 z_1, y_1 z_1, -y_1^2 - x_1^2) \cdot D \cdot \tan \beta. \quad (4)$$

To a given object o_j , if o_j is covered by c_i , it must be in some rectangle which is parallel to $ABCD$, i.e. rectangle Ω between c_i and $ABCD$ in Figure 2. According to this idea, we can illustrate the camera model as follows. Point O' is the centre of the rectangle and its coordinate is easy to figure out by o_j and normal vector $\vec{d}_{ci}$. Utilize normal vectors $\overrightarrow{OQ}$ in Equation (3) and $\overrightarrow{OP}$ in Equation (4), if o_j satisfies the following constraint, point o_j is covered by camera c_i .

$$F_c(c_i, o_j, \vec{d}_{ci}, \vec{d}_{oj}) = \begin{cases} 1, & \text{Prj}_{\overrightarrow{OP}} \overrightarrow{O'o_j} \leq |\overrightarrow{OP'}|, \\ & \text{Prj}_{\overrightarrow{OQ}} \overrightarrow{O'o_j} \leq |\overrightarrow{O'Q'}|, \\ & \text{Prj}_{\vec{d}_{ci}} \overrightarrow{c_i o_j} \leq D. \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

s.t. $|\overline{c_i O'}| = \text{Prj}_{\vec{d}_{ci}} \overrightarrow{c_i o_j}$, $|\overline{O'P'}| = |\overline{c_i O'}| \cdot \tan \beta$,
 $|\overline{O'Q'}| = |\overline{c_i O'}| \cdot \tan \alpha$.

B. 3D Directional Coverage Model

First, we define 3D directional coverage as follows.

Definition 3.1: (3D directional coverage) For an given object o_j and its facing direction $\vec{d}(x, y, z)$, there is an UAV c_i with camera orientation $\vec{d}_{ci}$, such that o_j is covered by c_i and $\alpha(\vec{d}, \overrightarrow{o_j c_i}) \leq \theta$ (θ is called the efficient angle), then object o_j is 3D directional covered by c_i .

According to Definition 3.1, object model can be established as a spherical base cone as Figure 3. Let Object o_j be the vertex, rotate a sector of θ central angle and D radius around vector $\vec{d}$ for one revolution, then we obtain the object model as follows.

$$\begin{cases} |\overline{c_i o_j}| \leq D, \\ \alpha(\overrightarrow{o_j c_i}, \vec{d}_{oj}) \leq \theta. \end{cases} \quad (6)$$

Based on Equation (6), UAV c_i can efficiently cover object o_j only when it locates in the cone of object o_j where can guarantee $|\overline{c_i o_j}| \leq D$ and $\alpha(\overrightarrow{o_j c_i}, \vec{d}_{oj}) \leq \theta$. Thereby, combine Equation (5) and (6), we can obtain the *directional coverage function* as

$$F_v(c_i, o_j, \vec{d}_{ci}, \vec{d}_{oj}) = \begin{cases} 1, & \text{Prj}_{\overrightarrow{OP}} \overrightarrow{O'o_j} \leq |\overrightarrow{OP'}|, \\ & \text{Prj}_{\overrightarrow{OQ}} \overrightarrow{O'o_j} \leq |\overrightarrow{O'Q'}|, \\ & \text{Prj}_{\vec{d}_{ci}} \overrightarrow{c_i o_j} \leq |\overline{c_i o_j}| \leq D. \\ & \alpha(\overrightarrow{o_j c_i}, \vec{d}_{oj}) \leq \theta. \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

s.t. $|\overline{c_i O'}| = \text{Prj}_{\vec{d}_{ci}} \overrightarrow{c_i o_j}$, $|\overline{O'P'}| = |\overline{c_i O'}| \times \tan \beta$,
 $|\overline{O'Q'}| = |\overline{c_i O'}| \cdot \tan \alpha$.

Then, the *directional coverage utility* can be defined as

$$U_v(c_i, \vec{d}_{ci}, o_j, \vec{d}_{oj}) = \begin{cases} 1, & \sum_{i=1}^N F_v(c_i, o_j, \vec{d}_{ci}, \vec{d}_{oj}) \geq 1, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

C. Problem Formulation

Let the tuple $\langle c_i, \vec{d}_{ci} \rangle$, called *strategy*, denotes the coordinate of UAV c_i and orientation of its camera $\vec{d}_{ci}$. Due to the limited resource of UAV, our task is to determine the strategies for all N UAVs to optimize overall directional coverage utility for all M objects, namely the number of directional covered objects. Formally, the PANDA problem is defined as follows.

$$\begin{aligned} (\mathbf{P1}) \quad & \max \sum_{j=1}^M U_v \left(\sum_{i=1}^N F_v(c_i, o_j, \vec{d}_{ci}, \vec{d}_{oj}) \right), \\ \text{s.t.} \quad & \gamma_{min} \leq \gamma \leq \gamma_{max}. \end{aligned}$$

Since the function $F_v(\cdot)$ is nonlinear and the constraints of **P1** is continuous, PANDA falls in the realm of nonlinear programs which are NP-hard [35]. Hence, we have

Theorem 3.1: The PANDA problem is NP-hard.

IV. SOLUTION

In this section, we present an algorithm with approximation ratio $1 - 1/e$ to address PANDA. Before describing the details of it, we first present the key intuitions and solution overview.

A. Key Intuitions and Solution Overview

1) *Key Intuitions:* (1) *Focus on possible covered sets of objects of UAVs rather than candidate positions and orientations of UAVs.* As we can deploy UAVs on any position and set the orientation of their cameras arbitrarily, the number of candidate strategies of UAVs is infinite, or the solution space of PANDA is infinite. However, many strategies are essentially equivalent if they cover the same set of objects. Apparently, we only need to consider one representative strategy among its associated class of all equivalent strategies, and the number of all such representative strategies is finite because the number of all possible covered sets of objects is finite. (2) *Focus on those strategies that cover large sets of objects.* If a representative strategy covers the set of objects $\{o_1, o_2, o_3, o_4\}$, undoubtedly you don't need to consider strategies that cover its subsets, such as $\{o_1, o_2\}$ or $\{o_2, o_3, o_4\}$. Our goal is to find those representative strategies that possibly cover maximal covered sets of objects while avoiding the others. (3) *Find or "create" constraints to help determine representative strategies.* To uniquely determine a strategy, mathematically we need at least 9 equations. Then, even if we know the associated covered set of objects for a class of equivalent strategies how can we efficiently determine at least one representative strategy? Our solution is to imagine that given a feasible strategy, we can adjust its position and orientation such that one or more objects touch some sides of the pyramid of the strategy while keeping no objects out of coverage. Obviously, the obtained strategy after the adjustment is also feasible and can be selected as a representative strategy. Then, we in turn use the touching conditions as constraints and formulate them as equations to help determine the representative strategies.

Apart from such kind of constraints, we also find an additional constraint regarding positions of representative strategies for their determination in this paper.

2) *Solution Overview*: First, any object only can be efficiently covered by an UAV in its efficient coverage space, which is modeled as a spherical base cone, by the 3D Directional Coverage Model. The whole solution space is thus the union of all spherical base cones for all objects. We further present a space partition approach to partition the whole solution space into multiple subspaces, which can be considered separately. Especially, for each subspace, the set of all possibly covered objects by adjusting the orientation of an UAV at any point in the subspace is exactly the same.

Second, we propose the notion of Dominating Coverage Set (DCS), which covers the maximal set of objects and has no proper superset of covered objects by other strategies. Then, our goal turns to find all candidate DCSs and their associated representative strategies. Specifically, we first prove that the positions of associated strategies of DCSs must lie on the boundaries of subspaces, which serves as a constraint to help determine the representative strategies. Next, we imagine that we adjust the positions and orientations of strategies to create touching conditions and thus new constraints. More importantly, we prove that it is sufficient to enumerate the seven different cases where 1 to 7 objects touch on the sides of pyramid in order to extract all possible DCSs, and the enumeration process must be done in polynomial time. Consequently, our problem is transformed into choosing N strategies among the obtained candidate DCS strategies to maximize the number of covered objects.

Finally, we prove that the transformed problem falls into the realm of maximizing a monotone submodular optimization problem subject to a uniform matroid, and propose a greedy algorithm to solve it with performance guarantee.

B. Space Partition

As mentioned in 3D Directional Coverage Model, efficient coverage space of each object is modeled as a spherical base cone. These spherical base cones intersect among each other and form many 3D partitions called *subspaces*.

Due to geometric symmetry, only the UAVs locating in subspaces have chance to cover objects, and their potentially covered objects vary from one subspace to another. For example, in Figure 1, UAV c_i locates in the common subspace of o_1 and o_2 and it can cover o_1 and o_2 simultaneously. Then we focus on the upper bound of the number of subspaces. First, we prove that the number of partitioned subareas Z for n uniform sectors in 2D plane is subject to $Z < 5n^2 - 5n + 2$ with Euler characteristic [36]. Then, based on above result, we prove the upper bound of the number of partitioned subspaces for uniform cones by reduction, as formally stated in the following theorem. We omit the detailed proof due to space limit.

Theorem 4.1: The number of partitioned subspaces is subject to $Z = O(M^2)$.

C. Dominating Coverage Set (DCS) Extraction

After the space partition, we only need to consider the relationship between objects and UAVs in each subspace, which depends on the coordinates and orientations of UAVs. In this subsection, we show that instead of enumerating all possible covered sets of objects, we only need to consider a limited number of representative covered sets of objects, which are defined as Dominating Coverage Sets (DCSs), and figure out their corresponding strategies. Our ultimate goal is to reduce the problem to a combinatorial optimization problem which is selecting N strategies from a limited number of strategies obtained by DCS extraction.

1) *Preliminaries*: To begin with, we give the following definitions to assist analysis.

Definition 4.1: (Dominance) Given two strategies $\langle c_1, \vec{d}_{c1} \rangle$, $\langle c_2, \vec{d}_{c2} \rangle$ and their covered sets of objects O_1 and O_2 . If $O_1 = O_2$, $\langle c_1, \vec{d}_{c1} \rangle$ is equivalent to $\langle c_2, \vec{d}_{c2} \rangle$, or $\langle c_1, \vec{d}_{c1} \rangle \equiv \langle c_2, \vec{d}_{c2} \rangle$; If $O_1 \supset O_2$, $\langle c_1, \vec{d}_{c1} \rangle$ dominates $\langle c_2, \vec{d}_{c2} \rangle$, or $\langle c_1, \vec{d}_{c1} \rangle \succ \langle c_2, \vec{d}_{c2} \rangle$; And if $O_1 \supseteq O_2$, $\langle c_1, \vec{d}_{c1} \rangle \succeq \langle c_2, \vec{d}_{c2} \rangle$.

Definition 4.2: (Dominating Coverage Set) Given a set of objects O_i covered by a strategy $\langle c_i, \vec{d}_{ci} \rangle$, if there does not exist a strategy $\langle c_j, \vec{d}_{cj} \rangle$ such that $\langle c_j, \vec{d}_{cj} \rangle \succ \langle c_i, \vec{d}_{ci} \rangle$, then O_i is a Dominating Coverage Set (DCS).

For a given subspace, it is possible only a few objects in the ground set of objects can be covered by an UAV locating in this subspace. We formally give the following definition.

Definition 4.3: (Candidate Covered Set of Objects) The candidate covered set of objects $\hat{O}_i$ for subspace S_k are those objects possible to be covered by UAV c_i with some orientation $\vec{d}_{ci}$ in S_k .

Obviously, any DCS of a subspace is a subset of its candidate covered set of objects $\hat{O}_i$.

As selecting DCSs is always better than selecting its subsets, we focus on figuring out all DCSs as well as their strategies.

2) *DCS Extraction*: First, we define four kinds of transformations as follows.

Definition 4.4: (Translation) Given a strategy $\langle c_1, \vec{d}_{c1} \rangle$, keep the orientation unchanged and move the UAV from coordinate c_1 to coordinate c_2 .

Definition 4.5: (Rotation Around Camera (RAC)) Given a strategy $\langle c_1, \vec{d}_{c1} \rangle$, keep the coordinate unchanged and rotate the orientation from $\vec{d}_{c1}$ to $\vec{d}_{c2}$.

Definition 4.6: (Rotation Around Objects (RAO)) Given a strategy $\langle c_1, \vec{d}_{c1} \rangle$, keep the object(s) on the touching side of pyramid, i.e. left side in Figure 4(c), and move the UAV from $\langle c_1, \vec{d}_{c1} \rangle$ to $\langle c_2, \vec{d}_{c2} \rangle$.

Definition 4.7: (Projection) Given a strategy $\langle c_1, \vec{d}_{c1} \rangle$, keep the orientation unchanged and move the UAV along the reverse direction of orientation $\vec{d}_{c1}$ until reaching some point c_2 on the boundary of subspace, i.e. $\langle c_2, \vec{d}_{c1} \rangle = f_{\perp}(\langle c_1, \vec{d}_{c1} \rangle)$.

Figure 4 depicts four instances of these four transformations. Figure 4(c) illustrates the RAO subject to objects o_3 and o_4 .

According to the definition of *projection*, we have the following lemma.

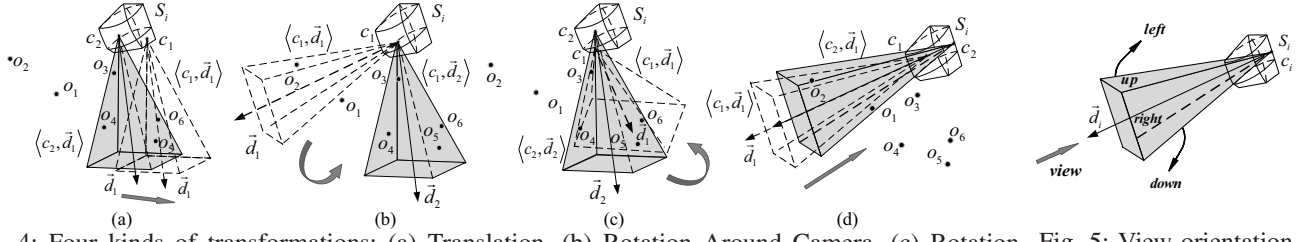


Fig. 4: Four kinds of transformations: (a) Translation, (b) Rotation Around Camera, (c) Rotation Around object(s), (d) Projection

Lemma 4.1: If $\langle c_2, \vec{d}_{c1} \rangle = f_{\perp}(\langle c_1, \vec{d}_{c1} \rangle)$, then $\langle c_2, \vec{d}_{c1} \rangle \geq \langle c_1, \vec{d}_{c1} \rangle$.

By Lemma 4.1 we can get the following corollary.

Corollary 4.1: Considering the case wherein UAVs lying on the boundaries of a subspace is equivalent to considering the whole subspace in terms of DCS extraction.

By Corollary 4.1, we only need to consider the strategies wherein cameras lying on the boundaries of subspace. We can perform the following transformation that begins with an arbitrary strategy $\langle c, \vec{d}_c \rangle$ where c lies on the boundary. First, we execute *RAC* until there is at least one object touches some side of the straight rectangle pyramid (note that an object will never fall out of the pyramid through its undersurface as we discussed before). Next, keeping c lying on the boundary and former touched objects lying on their former touching sides, execute *RAO* and *translation* such that there is at least another object touches some side of pyramid. Execute above transformation of *RAO* and *translation* under given constraints repeatedly, such that as many as possible objects touch sides of pyramid until there is no objects will touch any side, we call it final condition. Finally, the position and orientation of straight rectangle pyramid, namely strategy, can be either uniquely determined or not. For the former case, we can directly extract DCS of the unique strategy. For the latter, we can select an arbitrary strategy of final condition and extract DCS.

Because that during above transformation there is no object falling out of the pyramid, the set of covered objects of final condition dominates all sets under conditions of the process of transformation. Thus we only need to analyze the final condition which generate representative strategy. In particular, we can enumerate all possible cases of final conditions for which there are 1 to 7 objects touch sides of the pyramid. Then, we argue that there is NO performance loss and will prove it in Theorem 4.3.

In the following analysis, we use **(a, b, c)** to denote the case of final condition where $a(b, c)$ sides have three(two, one) objects touching each of them. We only analyze typical cases and omit the discussion of similar cases in this paper to save space. Figure 6 to 10 depict the typical cases of final conditions, whose view is from the inverse direction of $\vec{d}_{ci}$ as shown in Figure 5. The crossing dotted lines denote four edges of straight rectangle pyramid and their intersection point denotes the vertex of it.

To one- and two-object case, only need to choose a point c_i on the boundary of subspace S_i arbitrarily and extract DCSs greedily. In three-object case, there are three typical subcases.

(1) **(1, 0, 0)**. As **(1, 0, 0)** in Figure 6, o_1 , o_2 , and o_3 lie on the up side. Clearly, with the coordinates of three objects and expression of camera model, we have

$$\begin{cases} \vec{n}_{up} \cdot \vec{o_1 o_2} = 0, \vec{n}_{up} \cdot \vec{o_2 o_3} = 0, \\ \vec{d}_{ci} \cdot \vec{n}_{up} = \sin \beta, \vec{d}_{ci} \cdot \vec{n}_l = 0, \\ |\vec{n}_{up}| = 1, |\vec{d}_{ci}| = 1, |\vec{n}_l| = 1, \vec{n}_l / xOy. \end{cases} \quad (9)$$

where $\vec{n}_{up}$ is the normal vector of up side, $\vec{n}_l$ is the direction vector of the intersecting line of up side and the horizontal plane, and $\vec{d}_{ci} \cdot \vec{n}_l = 0$ describes camera can only rotate in the vertical plane we have discuss in Section III. Hence, we can obtain the orientation $\vec{d}_{ci}$ with Equation (9) and the candidate coordinates c_i can be expressed as follows:

$$\begin{cases} |\vec{c_i o_1}| \leq D, |\vec{c_i o_2}| \leq D, |\vec{c_i o_3}| \leq D, \\ \alpha(\vec{o_1 c_i}, \vec{d}_{o_j}) \leq \theta, \alpha(\vec{o_2 c_i}, \vec{d}_{o_j}) \leq \theta, \alpha(\vec{o_3 c_i}, \vec{d}_{o_j}) \leq \theta. \end{cases} \quad (10)$$

Then we only need to pick an arbitrary critical value of c_i that satisfies Inequality (10) to determine the strategy.

(2) **(0, 1, 1)**. First, as **(0, 1, 1)** shown in Figure 6, we can give the following equation:

$$\begin{cases} \vec{n}_{lf} \cdot \vec{o_1 o_2} = 0, \vec{n}_{lf} \cdot \vec{n}_{up} = \frac{\tan \alpha \tan \beta}{\sqrt{\sec^2 \alpha} \sqrt{\sec^2 \beta}}, \\ \vec{n}_{lf} \cdot \vec{n}_l = \cos \alpha, \vec{n}_{up} \cdot \vec{n}_l = 0, \\ |\vec{n}_{up}| = 1, |\vec{n}_{lf}| = 1, |\vec{n}_l| = 1, \vec{n}_l / xOy, \end{cases} \quad (11)$$

where $\vec{n}_{lf}$ is the normal vector of left side. With Equation (11), we can obtain an single-variable expression of the intersection line of left and up side. Then, combining Inequality (10) and the constraint of γ , we can determine the range of this parameter. Finally, selecting a legal parameter to determine intersection line, we can determine c_i easily with Inequality (10). Therefore, the strategy $\langle c_i, \vec{d}_{ci} \rangle$ can be determined. Subcases **(0, 1, 2)** and **(0, 1, 3)** can be solved by the same way. Here, we omit their analysis to save space.

(3) **(0, 0, 3)**. As **(0, 0, 3)** in Figure 6, select a point c_i on the boundary of subspace arbitrarily and connect $c_i o_1$, $c_i o_2$, and $c_i o_3$, respectively. Then, this subcase is transformed into **(0, 3, 0)** with two objects on each side. We can obtain the equation

$$\begin{cases} \vec{n}_{up} \cdot \vec{c_i o_1} = 0, \vec{n}_{rg} \cdot \vec{c_i o_2} = 0, \vec{n}_{bt} \cdot \vec{c_i o_3} = 0, \\ \vec{n}_{up} \cdot \vec{n}_{rg} = \frac{\tan \alpha \tan \beta}{\sqrt{\sec^2 \alpha} \sqrt{\sec^2 \beta}}, \vec{n}_{rg} \cdot \vec{n}_{bt} = \frac{\tan \alpha \tan \beta}{\sqrt{\sec^2 \alpha} \sqrt{\sec^2 \beta}}, \\ \vec{n}_{up} \cdot \vec{n}_{bt} = \cos 2\beta, |\vec{n}_{up}| = 1, |\vec{n}_{rg}| = 1, |\vec{n}_{bt}| = 1. \end{cases} \quad (12)$$

where $\vec{n}_{rg}$ is the normal vector of right side. Thus, selecting a feasible solution arbitrarily, we can get a strategy. Finally,

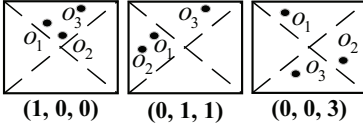


Fig. 6: Typical three-object cases

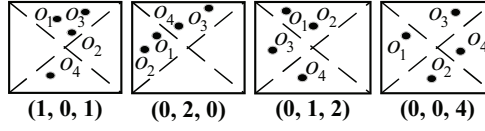


Fig. 7: Typical four-object cases

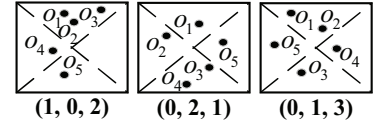


Fig. 8: Typical five-object cases

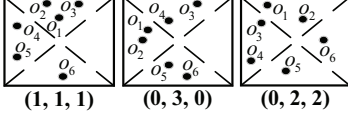


Fig. 9: Typical six-object cases

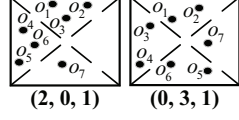


Fig. 10: Typical seven-object cases

execute *projection* until c_i reaching on the boundary of subspace to determine the final $\langle c_i, \vec{d}_{ci} \rangle$. Moreover, subcase $(0, 0, 4)$ can be solved by the same way.

In four-object case, we have two typical subcases.

(1) $(1, 0, 1)$. Similar to $(1, 0, 0)$, $\vec{d}_{ci}$ can be obtained, then normal vectors of four sides are easily to get. As $(1, 0, 1)$ in Figure 7, with the normal vector of down side and o_4 , we can obtain the intersection line expression of up side and down side. Then, selecting a point on this intersection line and execute *projection*, we can determine the strategy. Subcases $(1, 1, 0)$ in five-object and $(2, 0, 0)$ in six-object cases can be solved by the same way.

(2) $(0, 2, 0)$. As $(0, 2, 0)$ in Figure 7, combining $\vec{n}_{up} \cdot \vec{o_1 o_2} = 0$ and Equation (11), we can obtain the intersection line of up side and left side. Then, selecting c_i and $\vec{d}_{ci}$ by the same way as $(1, 0, 1)$, the strategy can be determined.

In five-object case, we have two typical cases.

(1) $(1, 0, 2)$. Similar to $(1, 0, 1)$, we can obtain the orientation $\vec{d}_{ci}$ and every normal vector of four sides. Thus, with the coordinates of o_4, o_5 and normal vectors of left and down sides as $(1, 0, 2)$ in Figure 8, we can determine c_i as well as a strategy. Then, execute *projection* until c_i reaching on the boundary of subspace to determine the final $\langle c_i, \vec{d}_{ci} \rangle$.

(2) $(0, 2, 1)$. Similar to $(0, 2, 0)$, we can obtain $\vec{d}_{ci}$ and normal vectors of four sides. With coordinates of o_1, o_3, o_5 and normal vectors of left, down, and right sides, we can obtain the intersection point of these three sides, saying c_i . Then, execute *projection* until c_i reaching on the boundary of subspace, we can obtain the final strategy.

Six-object case can be classified into two kinds of subcases. The first is $(2, 0, 0)$, it can be solved by the same way as $(1, 0, 1)$. The second kind includes $(1, 1, 1)$, $(0, 3, 0)$, and $(0, 2, 2)$, which has the only strategy.

In seven-object case, every subcase has the only strategy, no matter how distributed on four sides.

Based on the above analysis for all cases, we present Algorithm 1. Let Γ be the output set of DCSs in Algorithm 1, then we have the following theorem.

Theorem 4.2: Given any strategy $\langle c_i, \vec{d}_{ci} \rangle$, there exists $\langle c_k, \vec{d}_{ck} \rangle \in \Gamma$ such that $\langle c_k, \vec{d}_{ck} \rangle \succeq \langle c_i, \vec{d}_{ci} \rangle$.

Algorithm 1: DCS Extraction for the Area Case

Input: The subspace S_i , the candidate covered set of objects $\hat{O}_i$

Output: All DCSs

```

1 for  $i \leq 7$  (number of objects  $\leq$  the maximum of minimum
   number of objects on 4 sides to determine the only one
   strategy) do
2   for every combination  $o_{k_1}, \dots, o_{k_i}$  of all objects in  $\hat{O}_i$  do
3     if  $i \geq 3$  then
4       for every subcases  $a + b + c$  subject to
          $3a + 2b + c = i$  and  $a + b + c \leq 4$  do
5         for every possible arrangement of 4 sides
           taken  $a + b + c$  to arrange 3, 2, 1 objects
           respectively do
6           Execute the process following the
             corresponding subcase  $a + b + c$ .
7           if exists corresponding strategy  $\langle c_i, \vec{d}_{ci} \rangle$ 
             then
8             Add the results to the candidate DCS
               set. break
9       Select one point  $p$  on the boundary of  $S_i$  arbitrarily.
10      if  $i = 1$  then
11        Build strategy  $\langle p, \vec{d}_p \rangle$  with object  $o_k$  on the
          surface of straight rectangle pyramid.
12      if  $i = 2$  then
13        Decide if these objects can be in one camera
          coverage. If it is, build straight rectangle pyramid
          with line  $\vec{p o}_{k_1}$  and  $\vec{p o}_{k_2}$  on the surface.

```

Proof: Without loss of generality, we start from searching strategies for three-object cases. Assuming objects o_1, o_2 and o_3 touching one side of straight rectangle pyramid, we select an arbitrary feasible strategy $\langle c_1, \vec{d}_{c1} \rangle$. Then, keeping the three objects on this side and execute transformations, there exists numerous conditions which can be classified into three classes.

Class 1. There is the only one strategy for objects o_1, o_2 and o_3 . Obviously, the selected feasible strategy $\langle c_1, \vec{d}_{c1} \rangle$ is unique, and it must generate the only DCS such that $\langle c_1, \vec{d}_{c1} \rangle \in \Gamma$.

Class 2. There won't be any new object touch any side of pyramid. This condition implies $\langle c_1, \vec{d}_{c1} \rangle = \langle c_k, \vec{d}_k \rangle (k \neq 1, k \in U)$, where U is the universe of all strategies. Thus strategy $\langle c_1, \vec{d}_{c1} \rangle$ generates DCS such that $\langle c_1, \vec{d}_{c1} \rangle \in \Gamma$.

Class 3. Some new object(s) touch some side(s) of pyramid. Assuming object o_4 touches one side and we arbitrarily select a strategy $\langle c_2, \vec{d}_{c2} \rangle$ covering these four objects. Then, there exists two subclasses.

Subclass 3.1. $\langle c_2, \vec{d}_{c2} \rangle = \langle c_1, \vec{d}_{c1} \rangle$. This indicates that ob-

Algorithm 2: Strategy Selection

Input: The number of UAVs N , DCSs set Γ , object function $f(X)$

Output: Strategy set X

```

1  $X = \emptyset$ .
2 while  $|X| \leq N$  do
3    $e^* = \arg \max_{e \in \Gamma \setminus X} f(X \cup \{e\}) - f(X)$ .
4    $X = X \cup \{e^*\}$ .

```

ject o_4 has been covered by $\langle c_1, \vec{d}_{c1} \rangle$, then $\langle c_1, \vec{d}_{c1} \rangle$ generates the DCS such that $\langle c_1, \vec{d}_{c1} \rangle \in \Gamma$.

Subclass 3.2. $\langle c_2, \vec{d}_{c2} \rangle \succeq \langle c_1, \vec{d}_{c1} \rangle$. This indicates that object o_4 isn't covered by $\langle c_1, \vec{d}_{c1} \rangle$, thus $\langle c_1, \vec{d}_{c1} \rangle$ is not a DCS. However, as Algorithm 1, when searching strategy $\langle c_2, \vec{d}_{c2} \rangle$ for objects o_1, o_2, o_3 and o_4 in the next round of all combination of four objects on the sides, strategy $\langle c_1, \vec{d}_{c1} \rangle$ for objects o_1, o_2 and o_3 will be replaced by strategy $\langle c_2, \vec{d}_{c2} \rangle$. If no object will touch any side of pyramid during continuous transformation, strategy $\langle c_2, \vec{d}_{c2} \rangle$ generates the DCS such that $\langle c_1, \vec{d}_{c1} \rangle \preceq \langle c_2, \vec{d}_{c2} \rangle \in \Gamma$. Otherwise, similar to the above, strategy $\langle c_2, \vec{d}_{c2} \rangle$ will be replaced by $\langle c_3, \vec{d}_{c3} \rangle, \dots, \langle c_k, \vec{d}_{ck} \rangle$ iteratively until no object will touch any side or there is the only determined strategy. Strategy $\langle c_k, \vec{d}_{ck} \rangle$ generates the DCS for $o_1, \dots, o_k$, as well as, for o_1, o_2 and o_3 . Consequently, $\langle c_1, \vec{d}_{c1} \rangle \preceq \langle c_k, \vec{d}_{ck} \rangle \in \Gamma$. ■

D. Problem Reformulation and Solution

In this subsection, we discuss about how to select a given number of strategies from the obtained ones to maximize the number of coverage objects. We first reformulate the problem, then prove its submodularity, and finally present an effective algorithm to address this problem.

Let x_i be a binary indicator denoting whether the i_{th} strategy in the strategy set of DCSs Γ is select or not. For all DCSs from all subspaces in Γ , we can compute the coverage function with each object. The problem **P1** can be reformulated as

$$\begin{aligned}
 (\text{P2}) \quad & \max \sum_{j=1}^M \mathcal{U}_v \left(\sum_{\langle c_i, \vec{d}_{ci} \rangle \in \Gamma} x_i F_v(c_i, o_j, \vec{d}_{ci}, \vec{d}_{oj}) \right), \\
 \text{s.t.} \quad & \sum_{i=1}^{|\Gamma|} x_i = N (x_i \in \{0, 1\}).
 \end{aligned} \tag{13}$$

The problem is then transformed to a combinatorial optimization problem. Now, we give the following definitions to assist further analysis before addressing **P2**.

Definition 4.8: [37] Let S be a finite ground set. A real-valued set function $f : 2^S \rightarrow \mathbb{R}$ is normalized, monotonic, and submodular if and only if it satisfies the following conditions, respectively: (1) $f(\emptyset) = 0$; (2) $f(A \cup \{e\}) - f(A) \geq 0$ for any $A \subseteq S$ and $e \in S \setminus A$; (3) $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$ for any $A \subseteq B \subseteq S$ and $e \in S \setminus B$.

Definition 4.9: [37] A Matroid $\mathcal{M}$ is a strategy $\mathcal{M} = (S, L)$ where S is a finite ground set, $L \subseteq 2^S$ is a collection of independent sets, such that (1) $\emptyset \in L$; (2) if $X \subseteq Y \in L$, then

$X \in L$; (3) if $X, Y \in L$, and $|X| < |Y|$, then $\exists y \in Y \setminus X$, $X \cup \{y\} \in L$.

Definition 4.10: [37] Given a finite set S and an integer k . A uniform matroid $\mathcal{M} = (S, L)$ is a matroid where $L = \{X \subseteq S : |X| \leq k\}$.

Then, our problem can be reformulated as

$$\begin{aligned}
 (\text{P3}) \quad & \max f(X) = \sum_{j=1}^M \mathcal{U}_v \left(\sum_{\langle c_i, \vec{d}_{ci} \rangle \in X} F_v(c_i, o_j, \vec{d}_{ci}, \vec{d}_{oj}) \right), \\
 \text{s.t.} \quad & X \in L, \\
 & L = \{X \subseteq \Gamma : |X| \leq M\}.
 \end{aligned} \tag{14}$$

Lemma 4.2: The objective function $f(X)$ is a monotone submodular function, whose constraint is a uniform matroid.

Therefore, the reformulated problem falls into the scope of maximizing a monotone submodular function subject to matroid constraints, and we can use a greedy algorithm to achieve a good approximation [37]. The pseudo code of this strategy selecting algorithm is shown in Algorithm 2. In every round, Algorithm 2 greedily adds a strategy e^* to X to maximize the increment of function $f(X)$. We omit the proof to save space.

Theorem 4.3: Algorithm 2 achieves an approximation ratio of $1 - 1/e$, and its time complexity is $O(NM^9)$.

V. SIMULATION RESULTS

A. Evaluation setup

In our simulation, objects are uniformly distributed in a $100m \times 100m \times 50m$ cuboid space. If no otherwise stated, we set $\alpha = \pi/3$, $\beta = \pi/12$, $D = 25m$, $N = 10$, $\gamma_{min} = \pi/6$, $\gamma_{max} = \pi/3$, $\theta = \pi/6$, and $M = 20$, respectively. The orientations of objects are randomly selected from $[0, 2\pi]$ in horizontal plane and $[0^\circ, 90^\circ]$ in vertical plane. Each data point in evaluation figures is computed by averaging the results of 200 random topologies. As there are no existing approaches for PANDA, we present three compare algorithms. Randomized Coordinate with Orientation Discretization (RCOD) randomly generates coordinates of UAVs, and randomly selects orientation of UAVs from $\{0, \alpha, \dots, k\alpha, \dots, 2\pi\}$ in horizontal plane and $\{\gamma_{min}, \gamma_{min} + \beta, \dots, \gamma_{min} + \lfloor (\gamma_{max} - \gamma_{min}) / \beta \rfloor \beta, \dots, \gamma_{max}\}$ in vertical plane. Grid Coordinate with Orientation Discretization (GCOD) improves RCOD by placing UAVs at grid points. Grid Coordinate with Dominating Coverage Set (GCDACS) further improves GCOD by extracted DCSs greedily on each grids to generate candidate orientations and selects orientation greedily. Moreover, GCOD and GCDACS have the same square grid points whose length is $\sqrt{3}/3 \cdot D$.

B. Performance Comparison

1) *Impact of Number of UAVs N :* Our simulation results show that on average, PANDA outperforms RCOD, GCOD, and GCDACS by 12.35 times, 10.27 times, and 76.51%, respectively, in terms of N . Figure 11 shows that the coverage utility for all algorithms increase monotonically with N . In particular, the coverage utility of PANDA first fast increases and

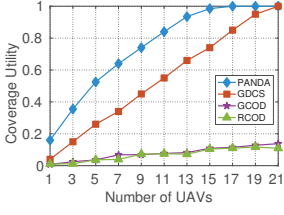


Fig. 11: N vs. utility

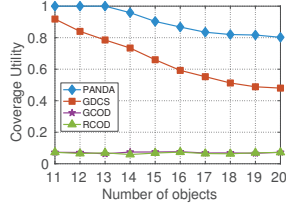


Fig. 12: M vs. utility

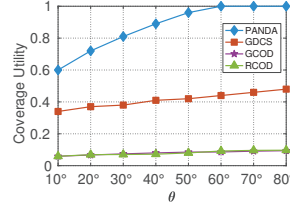


Fig. 13: θ vs. utility

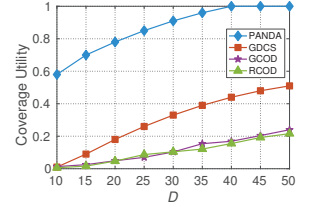


Fig. 14: D vs. utility

TABLE II: Coordinate and orientation of objects

Object	Coordinate	Orientation	Object	Coordinate	Orientation
o_1	(19.4, 0.7, 9.5, 8)	$(7\pi/4, \pi/2)$	o_9	(83.0, 2.7, 9.5, 0)	$(0, \pi/2)$
o_2	(21.0, 5.9, 3.3)	$(4\pi/5, 2\pi/6)$	o_{10}	(84.3, 3.2, 4.6)	$(\pi/4, \pi/3)$
o_3	(2.1, 0.9, 5.8)	$(0, \pi/6)$	o_{11}	(81.5, 19.3, 1.7)	$(3\pi/4, \pi/2)$
o_4	(9.6, 1.2, 5.4)	$(3\pi/4, 0)$	o_{12}	(16.2, 6.6, 9.1, 7)	$(\pi/2, 0)$
o_5	(11.4, 3.9, 4.2)	$(3\pi/4, 0)$	o_{13}	(9.94, 53.4, 0.5)	$(\pi/2, \pi/2)$
o_6	(18.7, 2.9, 4.6)	$(\pi/4, \pi/6)$	o_{14}	(83.2, 28.9, 0.5)	$(\pi, \pi/3)$
o_7	(3.0, 6.1, 3.3)	$(3\pi/2, \pi/2)$	o_{15}	(36.6, 63.8, 0.5)	$(3\pi/4, 0)$
o_8	(84.4, 3.0, 9.4, 6)	$(\pi, \pi/6)$			

approaches 1 when $N = 15$, and then becomes stable. GDCS increases relatively linearly because it can only choose among given grid coordinates for placing UAVs. In contrast, the coverage utility of RCOD and GCOD always remain low, because their candidate coordinates of UAVs are limited and orientations are predetermined or randomly generated.

2) *Impact of Number of Objects M* : Our simulation results show that on average, PANDA outperforms RCOD, GCOD, and GDCS by 12.37 times, 11.74 times, and 41.5%, respectively, in terms of M . From Figure 12, the coverage utility decreases monotonically with M . PANDA first performs well for no more than 13 objects but then decreases when M is larger than 13. The decreasing rate tends to be gentle and around 0.8. In contrast, GDCS invariably degrades while RCOD and GCOD always keep low performance.

3) *Impact of Efficient Angle θ* : Our simulation results show that on average, PANDA outperforms RCOD, GCOD, and GDCS by 10.01 times, 9.94 times, and 110.36%, respectively, in terms of θ . As shown in Figure 13, the coverage utility of four algorithms increases monotonically with θ . The coverage utility of PANDA first increases at a fast speed and approaches 1 when θ increases from 10° to 60° , and then keeps stable. However, the three comparison algorithms increase slowly.

4) *Impact of Farthest sight distance D* : Our simulation results show that on average, PANDA outperforms RCOD, GCOD, and GDCS by 21.20 times, 13.53 times, and 84.35%, respectively, in terms of D . Figure 14 shows that the coverage utility of PANDA invariably increases with D until it approaches 1, while that of RCOD, GCOD, and GDCS increase to about 0.2, 0.2, and 0.55, respectively, and then keeps relatively stable.

VI. FIELD EXPERIMENTS

As shown in Figure 15, our testbed consists of 7 DJI Phantom 4 advanced UAVs and 15 randomly distributed face figures as objects, and our experimental site is the playground of our school including its stands, whose size is $110\text{ m} \times 80\text{ m}$. Specifically, we set $\alpha = 35^\circ$, $\beta = 20^\circ$, $D = 10\text{ m}$, $\gamma_{\min} = 10^\circ$, $\gamma_{\max} = 70^\circ$, and $\theta = \pi/6$ based on real hardware

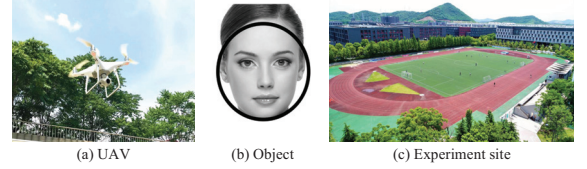


Fig. 15: Testbed

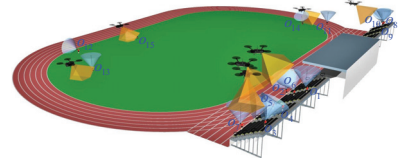


Fig. 16: Objects distribution and UAV placement by PANDA

parameters. The orientations of objects (θ, φ) are randomly generated where $\theta \in \{k\pi/4, k \in \{1, 2, 3, 4, 5, 6, 7, 8\}\}$ is the angle between orientation and xOz and $\varphi \in \{k\pi/6, k \in \{0, 1, 2, 3\}\}$ is the angle between orientation and xOy . Table II lists the obtained coordinates and orientations of all objects. Moreover, we draw a circle around the face on each face figure as shown in Figure 15(b) to help demonstrate the coverage result by observing the circle's distortion degree.

Figure 16 illustrates the placement results for PANDA. Note that the spherical base cones of objects are depicted in grey while the straight rectangle pyramids of UAVs are in yellow. Figure 17 shows the 7 pictures took by 7 UAVs for each of the three algorithms PANDA, GDCS, and GCOD. The rectangular enlarged views in each figure demonstrate the details of successfully efficient covered objects for the corresponding UAV. From Figure 17, we can see that PANDA covers the most objects among all the three algorithms. The coverage utilities of PANDA, GDCS, and GCOD are 0.93, 0.53, and 0.20, respectively, which means PANDA outperforms GDCS and GCOD by 75.4% and 3.65 times.

VII. CONCLUSION

In this paper, we solve the problem of 3D placement of UAVs to achieve directional coverage. The key novelty of this paper is on proposing the first algorithm for UAV placement with optimized directional coverage utility in 3D environment.

The key contribution of this paper is building the practical 3D directional coverage model, developing an approximation algorithm, and conducting simulation and field experiments for



Fig. 17: Experimental results of different algorithms

evaluation. The key technical depth of this paper is in reducing the infinite solution space of this optimization problem to a limited one by utilizing the techniques of space partition and Dominating Coverage Set extraction, and modeling the reformulated problem as maximizing a monotone submodular function subject to a matroid constraint. Our evaluation results show that our algorithm outperforms the other comparison algorithms by at least 75.4%.

ACKNOWLEDGMENT

This work was supported in part by the National Key R&D Program of China under Grant No. 2018YFB1004704, in part by the National Natural Science Foundation of China under Grant No. 61502229, 61872178, 61832005, 61672276, 61872173, 61472445, 61631020, 61702525, 61702545, and 61321491, in part by the Natural Science Foundation of Jiangsu Province under Grant No. BK20181251, in part by the Fundamental Research Funds for the Central Universities under Grant 021014380079, in part by the Natural Science Foundation of Jiangsu Province under Grant No. BK20140076.5, in part by the Nature Science Foundation of Jiangsu for Distinguished Young Scientist under Grant BK20170039.

REFERENCES

- [1] S. He *et al.*, "Full-view area coverage in camera sensor networks: Dimension reduction and near-optimal solutions," *IEEE TVT*, 2016.
- [2] X. Gao *et al.*, "Optimization of full-view barrier coverage with rotatable camera sensors," in *IEEE ICDCS*, 2017.
- [3] S. Hayat *et al.*, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys & Tutorials*, 2016.
- [4] L. Gupta *et al.*, "Survey of important issues in UAV communication networks," *IEEE Communications Surveys & Tutorials*, 2016.
- [5] "https://www.dji.com/phantom-4-adv/info."
- [6] V. Blanz *et al.*, "Face recognition based on frontal views generated from non-frontal images," in *IEEE CVPR*, 2005.
- [7] Y. C. Wang *et al.*, "Using Rotatable and Directional (R & D) Sensors to Achieve Temporal Coverage of Objects and Its Surveillance Application," *IEEE TMC*, 2012.
- [8] Z. Wang and others, "Achieving k-barrier coverage in hybrid directional sensor networks," *IEEE TMC*, 2014.
- [9] Y. Wang *et al.*, "On full-view coverage in camera sensor networks," in *IEEE INFOCOM*, 2011.
- [10] L. Li *et al.*, "Radiation constrained fair charging for wireless power transfer," in *ACM TOSN*, 2018.
- [11] H. Dai *et al.*, "Wireless charger placement for directional charging," in *IEEE TON*, 2018.
- [12] X. Wang *et al.*, "Robust scheduling for wireless charger networks," in *IEEE INFOCOM*, 2019.
- [13] —, "Heterogeneous wireless charger placement with obstacles," in *IEEE ICPP*, 2018.
- [14] N. Yu *et al.*, "Placement of connected wireless chargers," in *IEEE INFOCOM*, 2018.
- [15] H. Dai *et al.*, "Optimizing wireless charger placement for directional charging," in *IEEE INFOCOM*, 2017.
- [16] Z. Qin *et al.*, "Fair-energy trajectory plan for reconnaissance mission based on uavs cooperation," in *IEEE WCSP*, 2018.
- [17] K. Xie, L. Wang, X. Wang, G. Xie, and J. Wen, "Low cost and high accuracy data gathering in wsns with matrix completion," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1595–1608, 2018.
- [18] K. Xie, X. Ning, X. Wang, S. He, Z. Ning, X. Liu, J. Wen, and Z. Qin, "An efficient privacy-preserving compressive data gathering scheme in wsns," *Information Sciences*, vol. 390, pp. 82–94, 2017.
- [19] K. Xie, X. Li, X. Wang, G. Xie, D. Xie, W. J. Li, Zhenyu, and Z. Diao, "Quick and accurate false data detection in mobile crowd sensing," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 2019.
- [20] H. Ma *et al.*, "A coverage-enhancing method for 3d directional sensor networks," in *IEEE INFOCOM*, 2009.
- [21] L. Yupeng *et al.*, "A virtual potential field based coverage-enhancing algorithm for 3d directional sensor networks," in *ISSDM*, 2012.
- [22] J. Peng *et al.*, "A coverage-enhance scheduling algorithm for 3d directional sensor networks," in *CCDC*, 2013.
- [23] H.-J. Hsu *et al.*, "Face recognition on drones: Issues and limitations," in *ACM DroNet*, 2015.
- [24] "https://enterprise.dji.com."
- [25] G. Fusco *et al.*, "Selection and orientation of directional sensors for coverage maximization," in *IEEE SECON*, 2009.
- [26] T. Y. Lin *et al.*, "Enhanced deployment algorithms for heterogeneous directional mobile sensors in a bounded monitoring area," *IEEE TMC*, 2017.
- [27] Z. Yu *et al.*, "Local face-view barrier coverage in camera sensor networks," in *IEEE INFOCOM*, 2015.
- [28] Y. Hu *et al.*, "Critical sensing range for mobile heterogeneous camera sensor networks," in *IEEE INFOCOM*, 2014.
- [29] X. Yang and other, "3D visual correlation model for wireless visual sensor networks," in *IEEE ICIS*, 2017.
- [30] C. Han *et al.*, "An Energy Efficiency Node Scheduling Model for Spatial-Temporal Coverage Optimization in 3D Directional Sensor Networks," *IEEE Access*, 2016.
- [31] X. Yang and other, "3-D Application-Oriented Visual Correlation Model in Wireless Multimedia Sensor Networks," *IEEE Sensors Journal*, 2017.
- [32] P. Si *et al.*, "Barrier coverage for 3d camera sensor networks," *Sensors*, 2017.
- [33] M. Hosseini *et al.*, "Sensor selection and configuration in visual sensor networks," in *IST*, 2012.
- [34] J. Peng *et al.*, "A coverage detection and re-deployment algorithm in 3d directional sensor networks," in *CCDC*, 2015.
- [35] M. R. Gary *et al.*, "Computers and intractability: A guide to the theory of np-completeness," 1979.
- [36] J. D.S., "Euler's gem: The polyhedron formula and the birth of topology," *Elsevier*, vol. 6, 2008.
- [37] S. Fujishige, *Submodular functions and optimization*, 2005.