

Deploying Adjustable Multi-camera UAVs for Monitoring Tasks

Weijun Wang^{1,2} Haipeng Dai¹ Yue Zhao¹ Chao Dong³ Bangbang Ren² Guihai Chen¹ Xiaoming Fu²

¹Nanjing University ²University of Goettingen ³Nanjing University of Aeronautics and Astronautics

ABSTRACT

With the development of Unmanned Aerial Vehicles (UAVs), more and more industrial UAVs are equipped with multiple cameras for high work efficiency. However, there are few work consider this novel multi-camera UAV model so far. This paper studies a fundamental problem of Deploying adjustable Multi-camera UAVs for monitoring tasks (DoMino), that is, given a set of objects with known locations, deploy a set of multi-camera UAVs by adjusting UAVs' locations and each camera's direction as well as focal length such that the overall monitoring utility for all objects is maximized. Unfortunately, this problem involves two coupling NP-hard problems leading it much more complicated than the existing UAV placement problem. To address DoMino, we first propose an area discretization and Maximal Set Extraction (MSE) scheme to transform it to a combinatorial optimization problem. Next, in tackling the challenge of two tightly-coupled NP-hard problems, we construct this transformed problem as a two-level optimization problem, named DoMino-T, with specific theoretical analysis of submodularity. Then, we propose a two-level greedy algorithm to address DoMino-T and prove its approximation ratio to the original DoMino problem is $\frac{1}{2} + \frac{1}{2e^2} - \frac{1}{e} - \epsilon$. Extensive simulations are conducted, and the results show that our algorithm outperforms comparison algorithms at least 2.03×. Moreover, outdoor field experiments with 23 objects and 7 two-camera UAVs shows our algorithm achieve 65% of ground truth.

CCS CONCEPTS

• Networks → Mobile control network; • Theory of computation → Deployment algorithms;

KEYWORDS

UAV network; Multi-directional coverage; Deployment

1 INTRODUCTION

1.1 Motivation and Problem Statement

In past decades, the fast development of UAV has brought dramatic changes in emergency response, construction projects management, oil refinery management, and electricity grids and plant inspection [5]. From 2016, more and more companies, including DJI, Skydio, and Parrot, publish multi-camera UAVs, e.g., DJI M200 [2], DJI M300 RTK [2], DJI Guidance [1], Skydio R1 [3], and Waldo XCAM [6]. Although multiple cameras improve the work efficiency, they also pose significant challenges to UAV pilots. They need to control UAVs' flying attitude, determine UAVs' monitoring locations, and adjust every camera's direction and focal length. Therefore, an efficient

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobiHoc '21, July, Shanghai, China

© 2020 Association for Computing Machinery.

ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

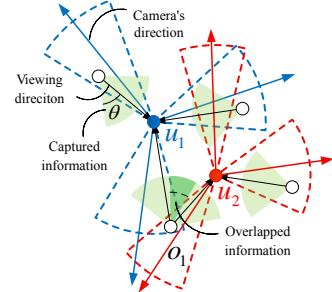


Figure 1: Deployment with Adjustable Multi-camera Model

deployment algorithm with guaranteed monitoring performance is highly needed to free UAV pilots from complicated operations.

Fig. 1 depicts a classic case of deploying a set of multi-camera UAVs to execute a monitoring task. The black hollow circles denote a set of objects with known locations to be monitored. The red and blue solid circles, saying u_1 and u_2 , denote two three-camera UAVs whose locations, cameras' directions and focal lengths to be determined. The red and blue dotted-line sectors and solid-line arrows denote the cameras' monitoring regions and directions, respectively. As the captured objects' information varies with the camera's focal length and distance between UAVs and objects [18], we define *Quality of Monitoring (QoM)* to quantify it. Besides, one camera can only capture partial information about an object because of the viewing direction, as shown by the black arrows in Fig. 1. Thus, we utilize aspect coverage, a new model for practical problems proposed by [42, 46], to distinguish whether an aspect of objects is monitored or not. For example, as shown by the light green sectors in Fig. 1, angles in the θ interval surrounding viewing direction can be efficiently monitored (Please refer to Section 3 for details). Moreover, multiple UAVs may monitor the same object from nearby viewing directions, saying o_1 in Fig. 1, and hence, they may capture overlapped information, as shown by the dark green sector with center o_1 in Fig. 1. To fuse the overlapped information from multiple UAVs, we propose *monitoring utility* (Please refer to Section 3 for details). During the deployment, cameras' focal lengths and directions can be respectively continuously adjusted from a minimum value to a maximum value and zero to 2π , and UAVs' monitoring locations can be deployed at any point in monitoring area. To sum up, we study a fundamental problem of Deploying adjustable Multi-camera UAVs for monitoring tasks (DoMino), that is, given a set of objects with known locations, deploy a set of multi-camera UAVs by adjusting UAVs' locations and each camera's direction and focal length such that the overall monitoring utility for all objects is maximized.

1.2 Limitations of Prior Art

Although UAV or camera deployment problems for monitoring tasks have been studied with plenty of work, surprisingly, there is no existing work considering the multi-camera model. Fig. 2 illustrates the problem space of camera model in deployment problem. We categorize the existing work, including UAV networks, camera sensor networks, and computer vision applications, by the camera model. Most work study the deployment problem with an unadjustable camera model to monitor different targets [16, 21, 27, 29, 36, 37, 40, 41], e.g., area,

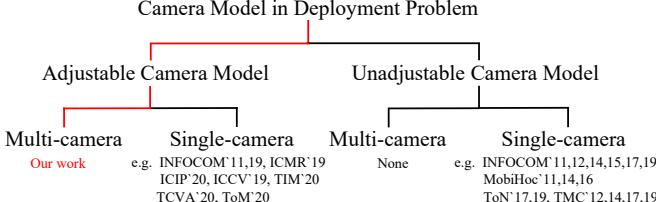


Figure 2: Problem Space

barrier, objects. Some work consider the adjustable camera model [8, 10, 19, 20, 22, 25, 28, 38, 48], but they use the single-camera model, which is not our consideration. In some computer vision applications, two unique camera models, *i.e.*, omnidirectional cameras and fisheye cameras, are considered [7, 34, 49]. Still, the disk model of omnidirectional cameras and the wide-angle camera model of fisheye cameras also differ significantly from ours. To the best of our knowledge, we are the first to consider the adjustable multi-camera model. Moreover, our method with a slight modification can also address the case with unadjustable multi-camera model (Please refer to Section 4.6 for details).

1.3 Key Technical Challenges

The first challenge is that the DoMino problem is nonlinear. The evaluation function of obtained information, *i.e.*, QoM, is nonlinear; the monitored objects' angles, *i.e.*, aspect coverage, are a number of trigonometric functions, which is nonlinear; the information fusion for one object, *i.e.*, monitoring utility, which is also nonlinear (Please refer to Section 3 for details). To sum up, DoMino is nonlinear.

The second challenge is that the solution space of the DoMino problem is continuous. UAVs can be deployed at any point in the monitoring area, and thus the location variable is infinite. Cameras' focal lengths and directions can be continuously adjusted, and hence, these two variables are continuous. In conclusion, the solution space of DoMino is continuous.

The last but most important is that the DoMino problem involves two tightly-coupled NP-hard problems. This challenge comes from the multi-camera model, and thus, there is not existing work encountering this challenge. One of these two NP-hard problems is the UAVs' location determination problem. Ignore cameras' directions and fix their focal lengths, namely, all UAVs are equipped with unadjustable 360° cameras. The DoMino problem is then slacked to deploy a set of unit disks to maximize the monitoring utility for all objects, which is NP-hard [33]. The other one NP-hard problem is the cameras' directions and focal lengths scheduling problem. Suppose all UAVs locate at fixed locations. Then DoMino is slacked to the problem of scheduling cameras' focal lengths and directions to maximize the monitoring utility for all objects, which is also NP-hard [20]. Further, the DoMino problem is not trivial putting two problems together. The decision variables, *i.e.*, UAVs' locations, cameras' directions, and focal lengths, impact each other, which means that the two NP-hard problems are tightly-coupled.

1.4 Proposed Approaches

To address the first challenge, we present two methods, *i.e.*, a piecewise constant function and an aspect combining technique, to approximate the nonlinear QoM function and the monitoring utility with performance bound, respectively. By doing so, the DoMino problem is linearized.

To address the second challenge, we present a Maximal Set Extraction (MSE) method to extract the set of representative arrangements, *i.e.*, candidate locations and subarrangements, rather than repeatedly

searching infinite equivalent ones. This method reduces the continuous solution space to a discrete one without performance loss.

To address the third challenge, we present a two-level greedy algorithm with a constant approximation ratio. After the first two challenges are addressed, the DoMino problem is transformed into a combinatorial optimization problem: select a fixed number of arrangements from a set of candidate ones to maximize the overall monitoring utility. To solve this problem, we propose a greedy algorithm that adds arrangements in order of non-increasing marginal benefit into the final arrangements set. However, the multi-camera model leads to computing the marginal benefit of each candidate arrangement, *i.e.*, evaluate each candidate arrangement's marginal benefit by adding it into the current selected arrangements set, itself is a hard problem. We thus relax the algorithm to use a greedy algorithm with polynomial-time and constant approximation ratio for marginal benefit computing. Finally, we prove the properties of the transformed combinatorial optimization problem and bound the approximation ratio of our two-level greedy algorithm.

1.5 Our Contributions

We summarize our contributions as follow.

- First, we propose a general information fusion method in monitoring utility function, in which any functions following submodularity are appropriate to it.
- Second, we propose a scheme including two mathematics tools, *i.e.*, discretization to tackle complex integral objective function and maximal set extraction to tackle continuous solution space, to transform nonlinear continuous DoMino problem to a combinatorial optimization problem DoMino-T.
- Third and the most important, we design a two-level greedy algorithm to solve the DoMino-T problem, which involves two coupling NP-hard problems. We prove the properties of the DoMino-T problem and bound the approximation ratio with a tricky proof technique. Further, the two-level greedy algorithm and the proof technique also apply to other two-level NP-hard problems, provided both of them in the scope of maximizing submodular problem.
- At last, we conduct both simulations and field experiments to validate our algorithm. The simulation results show that our algorithm outperforms comparison algorithms at least $2.03\times$. The field experiment results show that our algorithm achieve 65% of ground truth and outperforms comparison algorithms $4.24\times$.

1.6 Overview

The rest of this paper is organized as follows. Section 2 briefly reviews the related work, and Section 3 formally states the DoMino problem. Section 4 first summarizes the overview of the solution, then introduces each step and theoretically analyzes the solution. Section 5 and Section 6 respectively present the simulation results and experimental results, and Section 7 concludes the paper.

2 RELATED WORK

Unadjustable Camera Model. Most of the existing work [16, 17, 21, 27, 30, 31, 36, 37, 40–45, 50] study the deployment problem with an unadjustable camera model. [17, 30, 31, 37, 40, 41, 44, 50] focus on object (point) coverage, while [16, 43, 45] focus on barrier or area coverage. In particular, [16, 17, 30, 31, 40, 50] further consider impact of objects' facing directions to quality of monitoring. [16, 17, 30, 31, 43, 50] consider a full-view coverage model and utilize this model to monitor different targets. [40] consider the object's facing direction in 3-D space and solve the problem of maximizing the coverage number of objects achieving directional coverage. However,

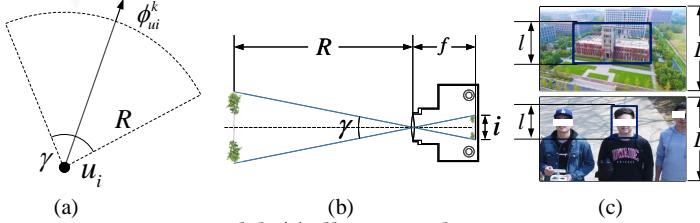


Figure 3: Camera Model. (a) illustrates the monitoring region of camera. (b) illustrates the relationship between f , R , and γ . (c) illustrates the parameters l and L in applications on large-scale building monitoring and human face recognition.

all above work try to change the object's monitoring model in various applications, while none consider different camera models.

Adjustable Camera Model. Some work [8, 10, 15, 19, 20, 22, 25, 28, 35, 38, 48] study deployment problem with adjustable camera model. Some of them consider the direction only adjustable camera model, e.g., [15, 35], which assume that sensors have been deployed in advance and design various scheduling mechanisms on cameras' directions. Some of them consider both focal length and direction adjustable camera [8, 10, 19, 20, 22, 25, 28, 38, 48]. Specifically, [20] assumes the cameras are pre-deployed and focuses on adjusting the direction and focal length to maximize the number of covered objects. [38] utilizes the adjustable camera model to maximize the covered portion of a given area. [28] and [8] both build monitoring model based on the pixel requirement of computer vision application, e.g., face detection, and design the photo selection and coverage algorithm based on their model, respectively. [19] and [22] both study the deployment in 3-D space. [19] focuses on the large-scale scenario and design an efficient distributed algorithm, while [22] focuses on covering heterogeneous objects under budget constraint. Due to the space constraint, we briefly review the related work in this paper, but you can get more details of camera models in these two comprehensive surveys [32] and [11].

Unique Camera Model. There are also some work in computer vision or cinematography applications [7, 21, 34, 36, 49]. [21] presents an end-to-end system to address the issue of optimizing the locations of two subjects captured in the photos by adjusting the trajectory of UAV and camera direction. [7, 34, 49] consider two unique camera model, i.e., omnidirectional cameras and fisheye cameras, to address computer vision related issues.

However, all the above work consider the single-camera model, and thus, none of them can solve our problem.

Multi-antenna Wireless Communication. We also review the multi-antenna model in the wireless communication domain [9, 13, 26, 39, 47, 51]. Some work focus on improving performance in Multi-input Multi-output (MIMO) system and simultaneous wireless information and power transfer (SWIPT) system [9, 26, 47, 51]. [47] and [9] survey the beamforming technique, aiming to improve the power transfer efficiency and communication robust, in the MIMO system and the SWIPT system. [26] bounds the performance gain of distributed antennas in one cell and proposes a deployment scheme to maximize the average users' communication rates. Some work study the connectivity between communication nodes by scheduling the antennas' directions and power [13, 39]. However, the antenna model is often modeled as a beam, which is quite different from camera models, and their deployment schemes focus on searching optimal one in discrete solution space, which also differ significantly from ours. Therefore, none of them can solve our problem.

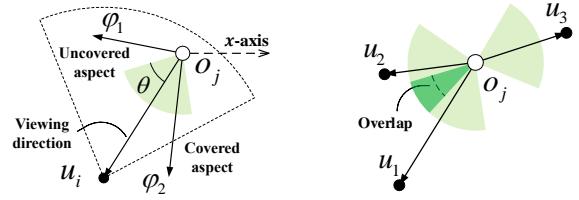


Figure 4: Aspect Coverage. **Figure 5: Monitoring Utility.** ϕ_2 of o_j is a covered aspect by Both u_1 and u_2 monitor the u_i while ϕ_1 is not.

Table 1: Notations and Symbols

Symbol	Meaning
u_i	UAV i , or its location
o_j	Object j to be monitored, or its location
\mathcal{U}	Set of all UAVs
O	Set of all objects
ϕ_{ui}^k	Direction of k -th camera on UAV i
f_{ui}^k	Focal length of k -th camera on UAV i
Φ_{ui}	Set of cameras' direction on UAV i
\mathcal{F}_{ui}	Set of cameras' focal length on UAV i
$\langle u_i, \phi_{ui}^k, f_{ui}^k \rangle$	Subarrangement of camera k on UAV i
$\langle u_i, \Phi_{ui}, \mathcal{F}_{ui} \rangle$	Arrangement of UAV i
γ	Angle of view (AOV) of camera's field of view
R	Range of distance of camera's field of view
ΔA	Angle of aspect combining
Λ	Set of extracted candidate locations
Γ_l	Set of location l 's candidate subarrangements

3 PROBLEM STATEMENT

Network Model. Suppose N objects $O = \{o_1, o_2, \dots, o_N\}$ are distributed in 3D space but with similar altitudes. We have M UAVs $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$, each equipped with K cameras, to be deployed to monitor these objects. Each UAV's state is described with a tuple $\langle u_i, \Phi_{ui}, \mathcal{F}_{ui} \rangle$ (called *arrangement*), in which u_i denotes the location of UAV i (as well as the location of its cameras), Φ_{ui} and \mathcal{F}_{ui} denote the directions set and focal lengths set of its cameras, respectively. The k -th camera on UAV i is described with $\langle u_i, \phi_{ui}^k, f_{ui}^k \rangle$ (called *subarrangement*), in which ϕ_{ui}^k and f_{ui}^k denote its direction and focal length. Table 1 lists the notations we used in this paper.

Camera Model. We adopt the camera model used in many literature [18, 28, 42, 46]. As shown in Fig. 3(a), the monitoring region of the camera model is a sector, which is determined by the angle of view (AOV) γ and the range of distance R . Both γ and R are determined by the camera's focal length f . Specifically, γ depends on f and the dimension of image i (Fig. 3(b)), while R depends on γ and the requirement of applications (Fig. 3(c)). Formally,

$$\gamma \triangleq 2 \arctan\left(\frac{i}{2f}\right), R \triangleq \cot\left(\frac{\gamma}{2}\right) \cdot \frac{P \cdot r}{2p} \cdot z, \quad (1)$$

where the digital zoom ratio z , the total pixels of the entire image L are the hardware determined, l and r are the required pixels and required ratio predefined by different applications. For example, in Fig. 3(c), UAVs can capture useful image a hundred meters away in monitoring large-scale buildings application, while can take useful images within a few meters in person identification application.

Quality of Monitoring (QoM). In most cases, the QoM in the sector-shape monitoring region is not uniform. On one hand, QoM varies with the distance between the camera and the object d and the focal length f [12, 20], and their relation satisfies $QoM \propto f^2/d^2$. On the other hand, QoM varies with the angle between the object's facing direction and the camera's viewing direction. Specifically, along with this angle increase, the QoM, e.g., face recognition accuracy, drops dramatically [23]. To sum up, the best way is to monitor objects

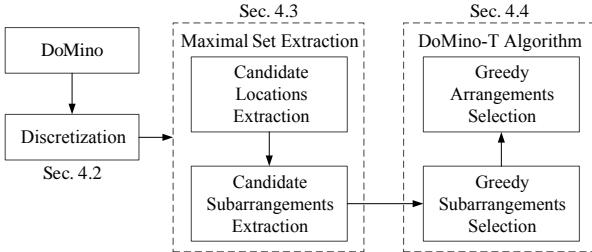


Figure 6: DoMino Workflow

with high QoM from as many angles of view as possible. To quantify the monitored angle of view, we utilize the *aspect coverage* concept proposed by [42, 46]. An aspect φ of an object is a direction that can be represented by an angle in $[0, 2\pi]$ with 0 degree indicating the one pointing to the positive direction of x -axis, as shown in Fig. 4. An aspect φ is covered if there is a camera on UAV u_i guaranteeing two conditions: 1. o_j is covered by u_i 's camera; 2. $\alpha(\varphi, \overrightarrow{o_j u_i}) < \theta$ ($\overrightarrow{o_j u_i}$ is the viewing direction, $\alpha(\cdot)$ is the angle between two vectors, and θ is the efficient angle). For example, in Fig. 4, φ_2 of o_j is a covered aspect by u_i while φ_1 is not. Formally, the QoM of o_j 's aspect φ is

$$Q(u_i, o_j, f_{ui}^k, \phi_{ui}^k, \varphi) \triangleq \begin{cases} \frac{a(f_{ui}^k)^2}{(\|u_i o_j\| + b)^2}, & 0 \leq \|u_i o_j\| \leq R, \\ \overrightarrow{u_i o_j} \cdot \phi_{ui}^k - \|u_i o_j\| \cos(\frac{\gamma}{2}) \geq 0, \\ \text{and } \overrightarrow{o_j u_i} \cdot \text{vec}(\varphi) - \|o_j u_i\| \cos \theta > 0, \\ 0, & \text{otherwise}. \end{cases}$$

where a and b are two constants determined by environment and hardware of devices, $\|u_i o_j\|$ is the distance between u_i and o_j , $\text{vec}(\varphi)$ is the vector with the angle of φ and length of $\|o_j u_i\|$, and γ and R can be calculated with Eq. (1) setting $f = f_{ui}^k$.

Monitoring Utility. By the QoM function, an object's monitoring utility is an integration of aspects' QoM value from 0 to 2π . However, according to the aspect coverage's conditions, we observe that if an aspect φ is covered, then its surrounding aspects in the θ interval, i.e., $[\varphi - \theta, \varphi + \theta]$, are all covered by the same one camera. Meanwhile, multiple cameras may cover the overlapped aspects with different QoMs, as shown in Fig. 5. Thus, quantifying the obtained information in the overlapped aspects, i.e., information fusion, is an issue to be addressed. We prove that any functions following submodularity applies to our algorithm (Please refer to Section 4.5 for details), e.g., trivial linear addition, cross-entropy, mutual information, even training a function with empirical experiment results. Here, we use $\max()$ function as an example, because when multiple UAVs monitor the same aspect, the image with maximum QoM contains complete information of all other images. Thus, the monitoring utility of an object o_j is

$$U(\mathcal{U}, o_j) \triangleq \int_0^{2\pi} \max_{\mathcal{A}} Q(u_i, o_j, f_{ui}^k, \phi_{ui}^k, \varphi) d\varphi \quad (2)$$

where \mathcal{A} is the arrangements set of \mathcal{U} .

Problem Formulation. Then, the problem of Deploying adjustable Multi-camera UAVs for monitoring tasks (**DoMino**) is, given M UAVs and the distribution of objects, determine the arrangements for all M UAVs equipped with K cameras to optimize overall monitoring utility for all N objects. Here, we define the overall monitoring utility as the normalized sum of N objects' monitoring utility.

DoMino Problem (P1):

$$\begin{aligned} \max & \quad \frac{1}{2\pi N} \sum_{j=1}^N \int_0^{2\pi} \max_{\mathcal{A}} Q(u_i, o_j, f_{ui}^k, \phi_{ui}^k, \varphi) d\varphi \\ \text{s.t.} & \quad |\mathcal{A}| = M, 0 \leq \phi_{ui}^k < 2\pi, 0 \leq \varphi < 2\pi. \end{aligned}$$

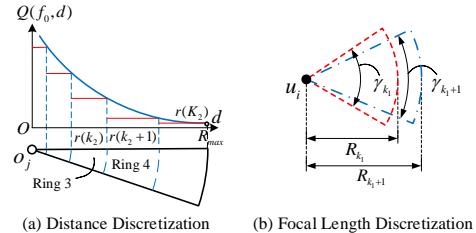


Figure 7: QoM Approximation

The following theorem indicates the hardness of DoMino.

THEOREM 3.1. *The DoMino problem is NP-hard.*

PROOF SKETCH. Consider a simple case in which focal length f is fixed, $\gamma = \theta = 2\pi$, $R = 1$, and QoM is set to be a $0 - 1$ function. In this case, as long as an object monitored by an UAV, all its aspects have been captured and other UAVs will not give any contribution even they also monitor this object. In this case, DoMino changes to the well-known Unit Disk Coverage problem [33], which is known to be NP-hard. Thus, we claim the DoMino problem is also NP-hard. \square

4 SOLUTION

4.1 Solution Overview

Fig. 6 shows the workflow of our solution. First, we present piecewise constant function and aspect combining technique to approximate the nonlinear QoM function and monitoring utility. By doing so, the whole solution space is discretized into many cells. Then, we present a Maximal Set Extraction (MSE) method, including Candidate Locations Extraction and Candidate Subarrangements Extraction, to extract the representative locations and subarrangements. Thus, the DoMino problem is transformed a combinatorial optimization problem DoMino-T. Finally, we design a two-level greedy algorithm to address DoMino-T as well as prove the properties of it and bound the algorithm's approximation ratio.

4.2 Discretization

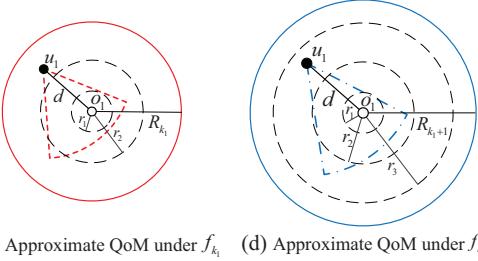
4.2.1 Piecewise Constant Approximation of QoM. Let $Q(f, d)$ denote the QoM of an aspect monitored by a camera respect to focal length f and distance d , i.e., $Q(f, d) = \frac{af^2}{(d+b)^2}$, where $0 \leq d \leq R_{max}$, $F_{min} \leq f \leq F_{max}$ ¹. We use multiple piecewise constant segments $\tilde{Q}(f, d)$ to approximate the QoM $Q(f, d)$, respectively. The core idea here is that utilize the minimum QoM in each interval to approximate the real QoM. Our goal is to bound the approximation error and the computational overhead.

In the distance dimension, let $r(0), r(1), \dots, r(K_2)$ be the endpoints of K_2 constant segments in an increasing order. Then, R_{max} is divided into little segments, and these segments divide the monitored region of an object into K_2 rings. As shown in Fig. 7(a), the monitored region of o_j is divided into 5 rings, and the QoM in each ring is approximated to a constant value when $f = f_0$.

In the focal length dimension, let $l(0), l(1), \dots, l(K_1)$ be the endpoints of K_1 constant segments in an increasing order. Each focal length l_{k_1} ($0 \leq k_1 \leq K_1$) corresponds to one type of camera model (sector) whose AOV γ and range R can be obtained via Eq. (1). Fig. 7(b) illustrates two types of sectors responding to $f = l_{k_1}$ and $f = l_{k_1+1}$. γ_{k_1} is larger than γ_{k_1+1} while R_{k_1} is shorter than R_{k_1+1} . By doing so, there are total K_1 different focal lengths, i.e., K_1 types of sectors. Then, any point in each ring monitored by one same type of sector,

¹ F_{min} and F_{max} are minimum and maximum focal length, and R_{max} is calculated by Eq. (1) at $f = F_{max}$.

² Without ambiguity, we may interchangeable use cameras and sectors.



(c) Approximate QoM under f_{k_1} (d) Approximate QoM under f_{k_1+1}

Figure 8: Toy Example

the QoM is a same constant. For example, Fig. 8(a) and Fig. 8(b) illustrate the approximate QoMs of o_1 under two focal lengths. The approximate QoMs of o_1 are respectively $Q(l_{k_1}, r_3)$ and $Q(l_{k_1+1}, r_3)$ since $R_{k_1} \leq r_3$.

Formally, the piecewise constant function is defined as follows.

DEFINITION 4.1. Setting $l(0) = F_{min}$, $l(K_1) = F_{max}$, $r(0) = 0$, and $r(K_2) = R_{max}$ ³, the piecewise constant QoM function $Q(f, d)$ is defined as follows:

$$\tilde{Q}(f, d) \triangleq \begin{cases} Q(l(1), r(1)), & f = l(0), d = r(0) \\ Q(l(k_1 - 1), r(k_2)), & l(k_1 - 1) < f \leq l(k_1), \\ & r(k_2 - 1) < d \leq r(k_2) \\ 0, & f > l(K_1), d > r(K_2). \end{cases} \quad (3)$$

Now, we provide a sufficient condition to guarantee the approximation error in the following two theorems. Theo. 4.1 first regards f as a constant and bounds the approximation error of $\tilde{Q}(f, d)$ in the distance dimension.

THEOREM 4.1. Holding fixed $f = f_0$, setting $r(0) = 0$, $r(K_2) = R_{max}$, and $r(k_2) = b((1 + \epsilon_1)^{k_2/2} - 1)$, ($k_2 = 1, \dots, K_2 - 1$, and $K_2 = \lceil \frac{\ln(Q(f, 0)/Q(f, R_{max}))}{\ln(1 + \epsilon_1)} \rceil$), the approximation error is

$$1 \leq \frac{Q(f_0, d)}{\tilde{Q}(f_0, d)} \leq 1 + \epsilon_1.$$

Based on Theo. 4.1, we then bound the complete approximation error of $\tilde{Q}(f, d)$. Due to space constraint, we provide both proofs in the technical report [4].

THEOREM 4.2. Setting $l(0) = F_{min}$, $l(K_1) = F_{max}$, and $l(k_1) = F_{min} \cdot \left(\frac{1 + \epsilon_2}{1 + \epsilon_1}\right)^{k_1/2}$, ($k_1 = 1, \dots, K_1 - 1$, and $K_1 = \lceil \frac{\ln(F_{max}/F_{min})}{\ln((1 + \epsilon_2)/(1 + \epsilon_1))} \rceil$), we have the approximation error:

$$1 \leq \frac{Q(f, d)}{\tilde{Q}(f, d)} \leq 1 + \epsilon_2.$$

4.2.2 Monitoring Utility Approximation for an Object. We present the *aspect combining* technique to approximate the monitoring utility of an object. Fig. 9 illustrates the basic idea of it. We use $\frac{2\pi}{\Delta A}$ directions centered at o_j with even space ΔA to combine o_j 's 2π aspects. Efficient angle θ in aspect coverage model is approximated as the completely covered integer multiple ΔA interval. For example, in Fig. 9, θ is approximated as the interval $[\Delta A, 4\Delta A]$ because $[0, \Delta A]$ and $[4\Delta A, 5\Delta A]$ are not completely covered.

Now, we provide a sufficient condition to guarantee the approximation error in the following theorem. Due to space constraint, we provide the proof in the technique report [4].

THEOREM 4.3. Setting $\Delta A = cd(360, \theta)$ ($cd(\cdot, \cdot)$ is the Common Divisor of two integers) and $\epsilon_\Delta = \frac{2\Delta A}{\theta - 2\Delta A}$, then, u_i monitors o_j with identical efficient interval provided $\overline{o_j u_i} \in (t\Delta A, (t + 1)\Delta A)$. The approximation error is

$$1 \leq \frac{U(\tilde{Q})}{U(Q)} \leq 1 + \epsilon_\Delta.$$

where $U(\tilde{Q}) = \int_0^{2\pi} \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, \varphi) d\varphi$.

³ K_1 and K_2 control the approximation error, and obviously, they incur less approximation error but more computational overhead with larger K_1 or K_2 .

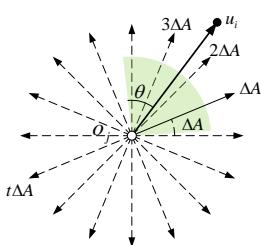


Figure 9: Aspect Combining

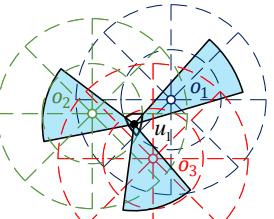


Figure 10: Cell Discretization

After the aspect combining, the calculation of monitoring utility of o_j with U is simplified from an integral to a weighted sum:

$$\tilde{U}(\tilde{Q}) = \tilde{U}(\mathcal{U}, o_j) = \Delta A \cdot \max_{t=1}^{\frac{2\pi}{\Delta A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A). \quad (4)$$

4.2.3 Discretizing to Cells. Following above two approximation techniques, we discretize the entire area into *cells*. Fig. 10 illustrates the basic idea of it, which contains two steps. First, following the distance-dimension discretization, we draw concentric circles with radius $r(0), r(1), \dots, r(K_2)$ at each object, respectively. Second, following the aspect combining, we draw aspects with $\Delta A, 2\Delta A, \dots, 2\pi$ angles centered at each object. By doing above two steps, the whole area is divided into a number of cells. For example, in Fig. 10, $\Delta A = \pi/4$, thus the entire area is divided into 148 cells. UAV u_1 lies in the ring 1 of o_1, o_2 , and o_3 , and its three cameras respectively monitor o_1 with discretized focal length $l(k_1 + 1)$, and o_2 and o_3 with $l(k_1)$. The approximated QoM of o_1 from u_1 is $Q(l(k_1 + 1), r(1))$, and the one of o_2 and o_3 is $Q(l(k_1), r(2))$.

Now, we state the bound of cells number enforced by Discretizing to Cell scheme in the following theorem. We provide the proof in the technical report [4] due to space constraint.

THEOREM 4.4. Following the cell discretization scheme, N objects partition the whole space into $O(N^2 \epsilon_1^{-2} \epsilon_\Delta^{-2})$ cells.

4.3 Maximal Set Extraction

After discretization, the monitoring utility at any point in each cell is approximated as a constant, provided the discretized focal length value f_0 is given. Hence, we focus on the relationship between objects and UAVs, which depends on the locations between UAVs and objects, and the cameras' focal lengths and directions. In this subsection, we show that the final results of decision variables only can be a limited number of representative arrangements who generate Maximal Sets, rather than other infinite equivalent arrangements. Based on this, our problem is transformed to extract all the representative arrangements and their corresponding maximal sets. This procedure includes two steps, candidate locations extraction and candidate subarrangements extraction. The first step extracts the locations for all representative arrangements, while the second step extracts all candidate subarrangements. Our goal is to convert the original problem into a combinatorial optimization problem.

4.3.1 Preliminaries. We first give the following definitions to assist analysis.

DEFINITION 4.2. Dominance Relation: Given two arrangements $\langle u_1, \Phi_{u1}, \mathcal{F}_{u1} \rangle$, $\langle u_2, \Phi_{u2}, \mathcal{F}_{u2} \rangle$ and their monitored object sets O_1, O_2 . If $O_1 = O_2$, we say $\langle u_1, \Phi_{u1}, \mathcal{F}_{u1} \rangle$ is equivalent to $\langle u_2, \Phi_{u2}, \mathcal{F}_{u2} \rangle$; If $O_1 \supseteq O_2$, we say $\langle u_1, \Phi_{u1}, \mathcal{F}_{u1} \rangle$ dominates $\langle u_2, \Phi_{u2}, \mathcal{F}_{u2} \rangle$.

DEFINITION 4.3. Maximal Set (MS): Given a set of objects O_1 monitored by an arrangement $\langle u_1, \Phi_{u1}, \mathcal{F}_{u1} \rangle$, if there is no arrangement $\langle u_1, \Phi_{u1}, \mathcal{F}_{u1} \rangle$ such that $\langle u_1, \Phi_{u1}, \mathcal{F}_{u1} \rangle$ dominates $\langle u_1, \Phi_{u1}, \mathcal{F}_{u1} \rangle$, then O_1 is a Maximal Set.

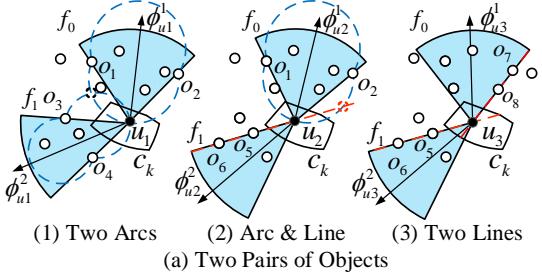


Figure 11: Candidate Locations Extraction

By Def. 4.3, we realize that deploying UAVs monitoring MSs is always better than monitoring its subsets. Thus, we focus on finding all MSs and their corresponding arrangements.

To this end, we define the candidate location as follows.

DEFINITION 4.4. Candidate Location: Given an arrangement $\langle u_i, \Phi_{ui}, \mathcal{F}_{ui} \rangle$, u_i is a candidate location iff there is at least one camera with $\langle u_i, \phi_{ui}^k, f_{ui}^k \rangle$ monitoring a maximal set.

4.3.2 Candidate Locations Extraction. In this subsection, we first present the candidate locations extraction algorithm, then prove this algorithm extracts all candidate locations without left.

The details of the candidate locations extraction algorithm is presented in Alg. 1. Let Λ be the extracted candidate locations set. Fig. 11 shows an example of how algorithm operates in one loop of focal length f_0 . Fig. 11(a), (b), and (c) correspond to Step 2-7, Step 8-10, and Step 11-13, respectively. We omit the figure of Step 14 because it is easy to understand.

Now, we state the guarantees enforced by Candidate Locations Extraction in the following theorem. Due to the space constraint, we provide the proof in technical report [4] but only sketch here.

THEOREM 4.5. Λ contains all the candidate locations that can possibly generate Maximal Sets.

PROOF SKETCH. Def. 4.4 defines that the candidate locations satisfy at least one of its cameras monitoring a maximal set. Thus we first find the critical conditions of one camera monitoring a maximal set. Starting from an arbitrary subarrangement and executing a series of transformations, we obtain the following lemma.

LEMMA 4.1. *The two necessary conditions for one camera monitoring a maximal set are: 1. two objects respectively located on two radii; 2. two objects located on the same one radius.*

Based on the Lem. 4.1, under the necessary condition 1 as shown in Fig. 11(c)(1), any points on the arc inside c_k are possible to be the candidate location. Similarly, under the necessary condition 2 as shown in Fig. 11(c)(2), any points on the line inside c_k are possible to be the candidate location. Moreover, any pair of arcs or lines intersect each other inside c_k can determine the only one candidate location as shown in Fig. 11(a)(b). Therefore, we only need to enumerate all possible combinations of arc and line pairs rather than consider cases more than two arcs or lines.

We use the proof by cases to prove the necessary and sufficient conditions for candidate locations. Case 1: only one camera monitoring maximal set, i.e., no more than one arc or line inside cell. There exists three subcases, only one arc (Fig. 11(c)(1)), only one line (Fig. 11(c)(2)), and no arc nor line inside cell, which respectively corresponds to Step 12, Step 13, and Step 14 in Alg. 1. Case 2: more than one camera monitoring maximal sets, i.e., one pair of arcs or lines inside cell. There also exists three subcases, two arcs (Fig. 11(a)(1)), two lines (Fig. 11(a)(2)), and one arc and one line inside cell (Fig. 11(a)(3)) and Fig. 11(b)), which respectively corresponds to Step 4, Step 7, and Step 6,10 in Alg. 1. \square

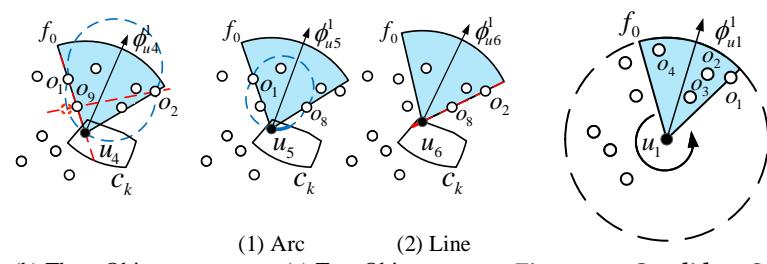


Figure 12: Candidate Subarrangements Extraction

Algorithm 1: Candidate Locations Extraction

```

Input: Cell  $c_k$  and corresponding  $O_{max}$ .  $O_{max}$  for cell  $c_k$  is the
set of objects can be monitored by a UAV  $u_i$  with  $2\pi$  AOV
and  $R_{max}$  range at arbitrary points inside  $c_k$ .
Output:  $\Lambda$  and corresponding fixed subarrangements.
1 for all types of discretized focal length, say  $f_0$  do
2   for all two pairs of objects  $o_p, o_q$  and  $o_s, o_t$  in  $O_{max}$  do
3     for all types of discretized focal length  $f_1$  do
4       Draw two arcs crossing  $o_p, o_q$  with inscribed angle  $\gamma_{f_0}$ ,
        and two arcs crossing  $o_s, o_t$  with  $\gamma_{f_1}$ , respectively; Go
        to Step 15.
5     for each pair of objects in the two pairs do
6       Draw two arcs crossing this pair of objects with
       inscribed angle  $\gamma_{f_0}$ , and a straight line crossing the
       rest pair, respectively; Go to Step 15.
7     Draw straight line crossing  $o_p, o_q$  and  $o_s, o_t$ , respectively;
     Go to Step 15.
8   for all three objects  $o_p, o_q$ , and  $o_s$  in  $O_{max}$  do
9     for all pairs  $o_1, o_2$  from three objects do
10      Draw two arcs crossing  $o_1, o_2$  with inscribed angle  $\gamma_{f_0}$ ,
         and two straight lines crossing  $o_1, o_3$  as well as  $o_2, o_3$ ,
         respectively; Go to Step 15.
11   for all pairs of  $o_p, o_q$  in  $O_{max}$  do
12     Draw two arcs crossing  $o_p$  and  $o_q$  with  $\gamma_{f_0}$  to get the
       intersecting curve with cell  $c_k$ ; Randomly select a point on
       intersecting curve, add it to  $\Lambda$  and record corresponding
       fixed subarrangements;
13     Draw a straight line crossing  $o_p$  and  $o_q$  to get the
       intersecting curve with cell  $c_k$ ; Randomly select a point on
       intersecting line, add it to  $\Lambda$  and record corresponding
       fixed subarrangements.
14   Randomly select a point inside cell  $c_k$  and add it to  $\Lambda$ .
15 If the intersecting point is in  $c_k$ , then we add it to  $\Lambda$  and record the
   corresponding fixed subarrangements.

```

4.3.3 Candidate Subarrangements Extraction. After candidate locations extraction, we obtain all the locations of representative arrangements. Now, we present the candidate subarrangements extraction algorithm to extract all directions and focal lengths that may monitor maximal sets for each candidate location. The details of the candidate subarrangements extraction procedure are presented in Alg. 2. Fig. 12 shows an example of extracting candidate subarrangements with focal length f_0 at candidate location u_i . It rotates the sector at location u_i such that its direction varies from 0° to 360° . During this progress, it tracks the current monitored objects while identifies MSs and records corresponding subarrangements. The extracted candidate subarrangements set is denoted as Γ , and Γ_l denotes the candidate subarrangements set of location l . After execute extraction loop at all candidate locations with all types of focal lengths, we obtain all subarrangements for all locations.

Algorithm 2: Candidate Subarrangements Extraction

Input: Cell c_k , O_{max} , Λ , and candidate location u_i .
Output: All candidate arrangements.

```

1 for all candidate locations in  $\Lambda$  do
2   for all types of discretized focal length  $l_{k1}$  do
3     Initialize the direction of sector to  $0^\circ$ .
4     while rotated angle is less than  $360^\circ$  do
5       while no object will fall out of sector do
6         Rotate the UAV anticlockwise.
7         Add current direction as well as focal length to the
8         candidate arrangements set, and record the
9         corresponding monitored objects.
  while no object will enter the sector do
    Rotate the UAV anticlockwise.

```

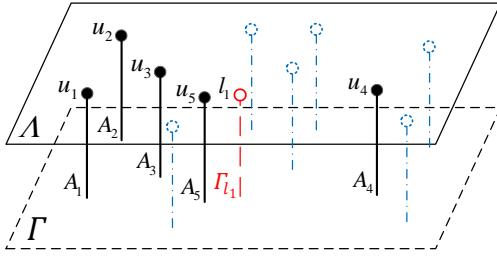


Figure 13: DoMino-T Problem

4.4 Problem Reformulation and Solution

After discretization and maximal set extraction, DoMino is transformed to a combinatorial optimization problem **DoMino-T**, that is, select a set L with M locations from Λ , and for each location l in L select K subarrangements from Γ_l to construct arrangement A_l , such that the monitoring utility for N objects is maximized. Yet, the challenge is that both selecting M locations from Λ and selecting K subarrangements from Γ_l are variants of NP-hard Set Cover problem [24]. Further, locations selection and subarrangements selection are tightly coupled because selected locations and their arrangements impact the following locations and subarrangements selection, resulting in a much more significant challenge.

To address this challenge, we present a two-level greedy algorithm in Alg. 3. In each iteration, Alg. 3 first evaluates the monitoring utility $H(\cdot)$ (defined in Section 4.5) of each location $l \in \Lambda \setminus L$ by greedily selecting $K - i$ subarrangements from Γ_l under current selected arrangements A_L (Step 3-7). For example, Fig. 13 illustrates the 6-th iteration which is evaluating l_1 under current selected arrangements set $\{A_1, A_2, A_3, A_4, A_5\}$. After evaluating all the locations in $\Lambda \setminus L$, Alg. 3 selects arrangement A_l^* of location l^* who contributes the maximum local marginal benefit of $f(\cdot)$ (defined in (P3) in Section 4.5) and adds A_l^* and l^* to A_L and L respectively, while deletes l^* from Λ (Step 8-9).

4.5 Theoretical Analysis

In this section, we give a series of analysis about the submodularity property of DoMino-T problem. On this basis, we prove the approximation ratio of our algorithm.

4.5.1 Properties of DoMino-T Problem. As shown in Fig. 13, DoMino-T can be regarded as a two-level optimization problem. Before analyzing DoMino-T's properties, we first formulate the upper level and lower level problem.

Algorithm 3: DoMino-T Algorithm

Input: Locations set Λ , candidate subarrangements set Γ , objects set O .
Output: Selected arrangements A_L , selected locations L .

```

1 Initialization:  $\Delta_H^a := H_{L \cup \{l\}}(A_L \cup \{a\}) - H_{L \cup \{l\}}(A_L)$ ;
   $\Delta_f^{AL} := f(A_L \cup \{A_l\}) - f(A_L)$ ;  $A_L := \emptyset$ ;  $L := \emptyset$ .
2 while  $|A_L| \leq M$  do
3   foreach  $l \in \Lambda$  do
4      $i :=$  number of fixed subarrangements in Alg. 1;  $A_l := \emptyset$ .
5     while  $|A_l| \leq K - i$  do
6        $a^* = \arg \max \{\Delta_H^a | a \in \Gamma_l\}$ ;
7        $A_l := A_l \cup \{a^*\}$ .
8    $A_{l^*} = \arg \max \{\Delta_f^{AL} | l \in \Lambda\}$ ;
9    $A_L := A_L \cup \{A_{l^*}\}$ ;  $L := L \cup \{l^*\}$ ;  $\Lambda := \Lambda \setminus L$ .

```

Let Π be the set of all arrangements constructed by lower level (Please refer to the DoMino-T problem box for the details of construct method), x_i be the indicator variable denoting whether the i -th location is selected or not. Then, DoMino problem **P1** can be reformulated as:

$$(P2) \quad \max \frac{\Delta A}{2\pi N} \sum_{j=1}^N \sum_{t=1}^{\frac{2\pi}{\Delta A}} (\max_{(u_i, \phi_{ui}, \mathcal{F}_{ui}) \in \Pi} x_i \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A)) \\ s.t. \quad \sum_{i=1}^{|A|} x_i = M \quad (x_i \in \{0, 1\}).$$

We first give following definitions to assist its properties analysis.

DEFINITION 4.5. [14] Let S be a finite ground set. A real-valued set function $f : 2^S \rightarrow \mathbb{R}$ is normalized, monotonic, and submodular iff. it satisfies: (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 4.6. [14] 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 4.7. [14] Given a finite set S and an integer k . A uniform matroid $\mathcal{M} = (S, L)$ subjects to $L = \{X \subseteq S : |X| \leq k\}$.

By these definitions, problem **P2** can be rewritten and served as the two-level DoMino-T problem,

DoMino-T Problem (P3):

Lower Level Subproblem: Given L selected locations with arrangements A_L , and an additional location $e \in \Lambda \setminus L$, select K subarrangements from Γ_e forming arrangements A_e to maximize the overall monitoring utility for N objects.

$$\max H_{L \cup \{e\}}(Y) = \frac{\Delta A}{2\pi N} \sum_{j=1}^N \sum_{t=1}^{\frac{2\pi}{\Delta A}} (\max_{(u_i, \phi_{ui}, \mathcal{F}_{ui}) \in A_L \cup Y} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A)) \\ s.t. \quad Y \in \Gamma_e, A_e = \{Y \subseteq \Gamma_e : |Y| \leq K\}.$$

Upper Level Subproblem: Based on the constructed arrangements on the lower level, select M arrangements to maximize the overall monitoring utility of N objects.

$$\max f(X) = \frac{\Delta A}{2\pi N} \sum_{j=1}^N \sum_{t=1}^{\frac{2\pi}{\Delta A}} (\max_{(u_i, \phi_{ui}, \mathcal{F}_{ui}) \in X} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A)) \\ s.t. \quad X \in L, L = \{X \subseteq A : |X| \leq M\}.$$

Now, we state the property of the upper level subproblem in the following lemma. Due to the space constraint, we omit their proofs but provide them in technical report [4].

LEMMA 4.2. *The objective function $f(X)$ is monotone and submodular, and its constraint is a uniform matroid.*

The property of lower level subproblem, as shown in following lemma, can be proved with the similar proof technique of Lem. 4.2, thus we omit here to save space.

LEMMA 4.3. *The objective function $H_{L \cup \{e\}}(X)$ is monotone and submodular, and its constraint is a uniform matroid.*

By Lem. 4.2 and Lem. 4.3, both upper and lower level optimizations fall into the scope of the problem that maximizing a monotone submodular function subject to uniform matroid constraint, which can be solved by an approximate algorithm with greedy policy [14].

4.5.2 Approximation Ratio and Time Complexity Analysis. To save space, we only prove the most tricky part, *i.e.*, the approximation ratio between Alg. 3 and problem **P2** (also **P3**). Then, combining Theo. 4.2, and Theo. 4.3, the ultimate approximation ratio can be bounded as follows.

THEOREM 4.6. *The DoMino algorithm achieves an approximation ratio of $\frac{1}{2} + \frac{1}{2e^2} - \frac{1}{e} - \epsilon$, where $\epsilon = \frac{e^2\tau + \tau - 2e\tau}{2e^2 + 2e^2\tau}$ and $\tau = \epsilon_\Delta + \epsilon_2 + e\epsilon_\Delta\epsilon_2$, and its time complexity is $O(MKN^6\epsilon^{-4})$.*

PROOF. Let **OPT** be the optimal solution of **P2** and **P3**.

Let $\{l_1, l_2, \dots, l_M\}$ denote the locations set of selected arrangements, where l_i is the location of selected i -th arrangement in a fixed order generated by scheme L . Here, $\{l_1^{++}, l_2^{++}, \dots, l_M^{++}\}$ denotes the locations of selected arrangements set generated by scheme L^{++} that greedily selects subarrangements and greedily selects location, *i.e.*, Alg. 3. $\{l_1^{*+}, l_2^{*+}, \dots, l_M^{*+}\}$ denotes the locations generated by scheme L^{*+} that selects *optimal* subarrangements for each location and greedily selects locations. Note that the orders of l_i in the two sets are not the same since the different schemes.

Next, we introduce three auxiliaries to assist our proof. 1. $\Delta L_{i-1}(\tilde{l}_i)$ denotes the marginal utility increment of adding arrangement \tilde{l}_i to arrangements set L_{i-1} . For example, $\Delta L_{i-1}^{++}(l_i^{++})$ denotes the marginal utility increment of adding l_i^{++} in the i -th iteration of Alg. 3. 2. $\Delta L_{i-1}^{++}(l_i^{*+})$ denotes the marginal utility increment of adding an arrangement associated with scheme L^{++} at location l_i^{*+} rather than arrangement l_i^{++} . In other words, it is a created arrangement that greedily selects subarrangements at location l_i^{*+} . 3. $L^{++} \bowtie L^{*+} \triangleq \{l_1^{++}, l_2^{++}, \dots, l_M^{++}, l_1^{*+}, l_2^{*+}, \dots, l_M^{*+}\}$ which concatenates two locations set L^{++} and L^{*+} . Its physical meaning is the locations generated by scheme that *deploy 2M of UAVs, the first M UAVs deploy by scheme L^{++} and the second M UAVs deploy by scheme L^{*+}* . Then, $\Delta L^{++} \bowtie L_{i-1}^{*+}(l_i^{*+})$ denotes the increment of marginal utility by greedily selecting subarrangements at location l_i^{*+} on $M+i-1$ locations including M UAVs by scheme L^{++} and $i-1$ UAVs by L^{*+} .

Now, let us bound the approximation ratio. First, $\Delta L_{i-1}^{++}(l_i^{++}) \geq \Delta L_{i-1}^{++}(l_i^{*+})$ since the greedy property, namely, each selected location with scheme L^{++} contribute the most marginal utility increment, of course, its marginal utility is larger than any other locations following scheme L^{++} . Second, $\Delta L_{i-1}^{++}(l_i^{*+}) \geq \Delta L^{++} \bowtie L_{i-1}^{*+}(l_i^{*+})$ since the submodularity of $f(X)$ and the fact $L_{i-1}^{++} \subseteq L^{++} \bowtie L_{i-1}^{*+}$. Third, because of the definition $\Delta L^{++} \bowtie L_{i-1}^{*+}(l_i^{*+}) = H_{L^{++} \bowtie L_{i-1}^{*+} \cup \{l_i^{*+}\}}(X)$, Lemma 4.3, and greedily subarrangements selection, we have $\Delta L^{++} \bowtie L_{i-1}^{*+}(l_i^{*+}) \geq (1 - 1/e) \cdot \Delta L^{++} \bowtie L_{i-1}^{*+}(l_i^{*+})$ [14], where l_i^{*+} is the optimal subarrangements selection at location l_i^{*+} . Combine above three inequalities, we have

$$\Delta L_{i-1}^{++}(l_i^{++}) \geq (1 - \frac{1}{e}) \cdot \Delta L^{++} \bowtie L_{i-1}^{*+}(l_i^{*+}). \quad (5)$$

Then, summate all marginal utility and by Inequality (5) we have

$$\begin{aligned} \sum_{i=1}^M \Delta L_{i-1}^{++}(l_i^{++}) &\geq \sum_{i=1}^M (1 - \frac{1}{e}) \cdot \Delta L^{++} \bowtie L_{i-1}^{*+}(l_i^{*+}) \\ 2 \cdot \sum_{i=1}^M \Delta L_{i-1}^{++}(l_i^{++}) &\geq \sum_{i=1}^M \Delta L_{i-1}^{++}(l_i^{++}) + (1 - \frac{1}{e}) \cdot \sum_{i=1}^M \Delta L^{++} \bowtie L_{i-1}^{*+}(l_i^{*+}) \\ 2 \cdot f(L^{++}) &\geq (1 - \frac{1}{e}) \cdot f(L^{++} \bowtie L^{*+}) \\ f(L^{++}) &\geq (\frac{1}{2} - \frac{1}{2e}) \cdot f(L^{++} \bowtie L^{*+}). \end{aligned}$$

The third inequality holds since $f(L^{++}) = \sum_{i=1}^M \Delta L_{i-1}^{++}(l_i^{++})$ and $f(L^{++} \bowtie L^{*+}) = \sum_{i=1}^M \Delta L_{i-1}^{++}(l_i^{++}) + \sum_{i=1}^M \Delta L^{++} \bowtie L_{i-1}^{*+}(l_i^{*+})$.

By Lemma 4.2 and [14], we have $f(L^{*+}) \geq (1 - \frac{1}{e}) \cdot \text{OPT}$. Then,

$$\begin{aligned} f(L^{++}) &\geq (\frac{1}{2} - \frac{1}{2e}) \cdot f(L^{++} \bowtie L^{*+}) \\ &\geq (\frac{1}{2} - \frac{1}{2e}) \cdot f(L^{*+}) \geq (\frac{1}{2} - \frac{1}{2e}) \cdot (1 - \frac{1}{e}) \cdot \text{OPT}. \end{aligned}$$

Hence, we bound the approximation ratio between Alg. 3 and **P2**. We provide the time complexity analysis in technique report [4] due to the space constraint. \square

4.6 Discussion

Deploying Unadjustable Multi-camera for Monitoring Tasks. Specifically, the problem is that, given N object with know locations, given M UAVs and each of them equipped with k unadjustable cameras, determine the the locations of UAVs and directions of each camera to optimize overall monitoring utility for all object. The solution is almost the same as DoMino except two modifications. The first modification is that, we only need to execute the discretization in distance domain because unadjustable camera's focal length is constant. The second modification is that, Alg. 1 and Alg. 2 only need to execute the extraction loop for one type of focal length. Then, utilizing the Alg. 3, DoMino-T algorithm, we can solve this problem with performance guarantee in polynomial time.

5 SIMULATION RESULTS

5.1 Evaluation Setup

Parameters Setup. In our simulation, objects are uniformly distributed in a $400m \times 400m$ square area. If no otherwise stated, $M = 7, N = 15, K = 3, \theta = 30^\circ, \epsilon_1 = \frac{1}{6}, \epsilon_2 = \frac{1}{4}, \epsilon_\Delta = \frac{1}{3}, i = \sqrt{3}, f \in [30, \frac{\sqrt{3} \times 10}{4-2\sqrt{3}}], \frac{P \cdot r \cdot z}{P} = \frac{10}{\sqrt{3}}$ in Equ. (1), respectively. The above parameters are set up based on our experimental results with real hardware [2], which let the monitoring region of one camera vary in $\gamma \in [30^\circ, 60^\circ]$ and $R \in [5m, 10.2m]$. Each data point in the figures is computed by averaging the results of 200 random topologies.

Baseline Setup. Since there are no existing approaches for DoMino, we present three comparison algorithms. (1) *Random Location and Random Subarrangement (RLRS)* randomly generates locations of UAVs, and randomly selects K directions from $\{0, \alpha, \dots, k\alpha, \dots, 2\pi\}$ and focal lengths from $[30, \frac{\sqrt{3} \times 10}{4-2\sqrt{3}}]$ for UAVs, respectively. (2)

Constraint Location and Random Subarrangement (CLRS) discretizes entire monitoring area into square grids whose length is $\sqrt{2}/2 \cdot R_{max}$. Then, randomly select M locations from grid points and guarantee all selected locations within at least one circle centering at an object and radius with R_{max} . The subarrangements are generated with the same method of RLRS. (3) *Constraint Location and Greedy Subarrangement (CLGS)* improves CLRS by placing UAVs with the same way of CLRS but greedily selecting subarrangement for each location.

5.2 Performance Comparison

5.2.1 Impact of Number of UAVs M . Our simulation results show that on average, DoMino outperforms CLGS, CLRS, and RLRS by 2.13×,

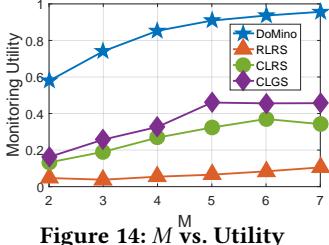
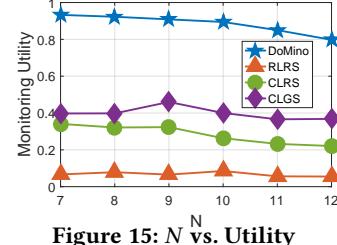
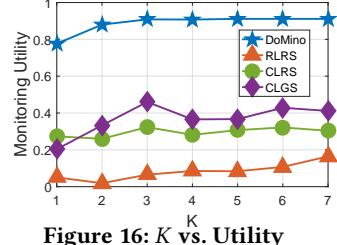
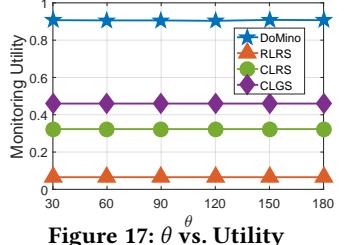
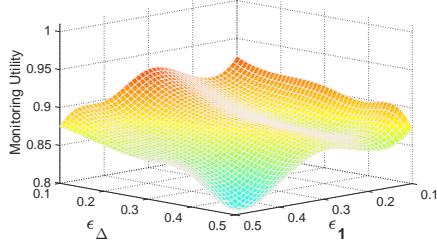
Figure 14: M vs. UtilityFigure 15: N vs. UtilityFigure 16: K vs. UtilityFigure 17: θ vs. Utility

Figure 18: Insight

$2.83\times$, and $10.05\times$, respectively, in terms of M . Fig. 14 shows that the monitoring utility of all algorithms increases monotonically. In particular, the monitoring utility of DoMino starts at a higher value than comparison algorithms because it has a performance guarantee, *i.e.*, approximation ratio, then it increases until very close to 1. The monitoring utility of comparison algorithms also increases but relatively limited and fluctuant because of their random directions and locations selection mechanism.

5.2.2 Impact of Number of Objects N . Our simulation results show that on average, DoMino outperforms CLGS, CLRS, and RLRS by $2.03\times$, $3.28\times$, and $9.31\times$, respectively, in terms of N . Fig. 15 shows that DoMino achieves high monitoring utility and performs consistently better than comparison algorithms. However, the monitoring utility of DoMino has a slight trend of decreasing, and it seems the decrease is more considerable than comparison algorithms. But actually, the DoMino only drops 12.3% while the three comparison algorithms drop 19.2% on average.

5.2.3 Impact of Number of Cameras K . Our simulation results show that on average, DoMino outperforms CLGS, CLRS, and RLRS by $2.93\times$, $3.16\times$ and $9.13\times$, respectively, in terms of K . Fig. 16 shows that DoMino’s monitoring utility increases first and then stays at a high value. Comparison algorithms have a slight increase but unstably because of their random selection mechanism.

5.2.4 Impact of Efficient Angle θ . Our simulation results show that on average, DoMino outperforms CLGS, CLRS, and RLRS by $2.26\times$, $2.93\times$, and $11.05\times$, respectively, in terms of θ . Fig. 17 shows that DoMino consistently achieves high monitoring utility because its aspect combining technique floats the change from various θ . This simulation results indicate that DoMino can satisfy the requirement of anisotropic monitoring tasks.

5.2.5 Insight. Here we study the impact of approximation errors ϵ_Δ and ϵ_1 and both of them vary from 0.1 to 0.5. Fig. 18 depicts the results, and each point on the surface denotes an average value of 200 experiment results of different topologies. We observe that the monitoring utility always stays at a high value (≥ 0.85) when ϵ_1 and ϵ_Δ are both ≤ 0.4 . Thus, DoMino can guarantee a high approximation ratio with relatively low computational overhead.

6 FIELD EXPERIMENTS

Experiment Setup. As shown in Fig. 19, our testbed consists of a teaching building with 25 gates and 7 two-camera UAVs. UAVs fly in the $7.5m$ sky, and the angle of pitch is set to 20° . The experiment



Figure 19: Gates Distribution and Images

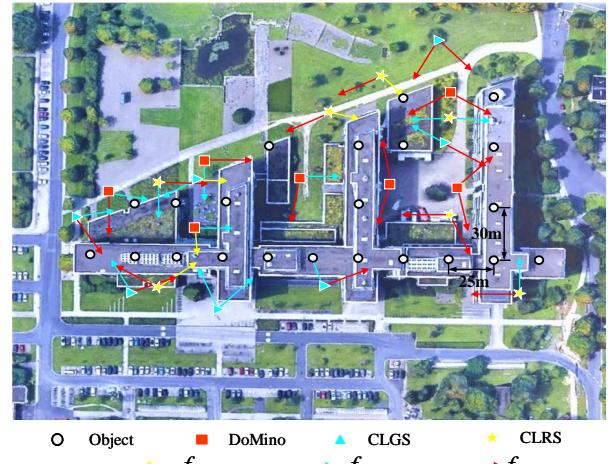


Figure 20: Objects Distribution and Arrangements Distribution of Three Algorithms

field is $150m \times 200m$, and the parameters of the camera, $\gamma \in [30^\circ, 60^\circ]$ and $R \in [20m, 45m]$, are set up based on our experimental results with real hardware [2]. The approximate errors are set to $\epsilon_1 = 1/3$, $\epsilon_2 = 4/11$, and $\epsilon_\Delta = 2/3$, which leads the range of focal length value discretized into 3 intervals, *i.e.*, 3 different camera models. As shown in Fig. 20, 3 arrows named f_0 , f_1 , and f_2 with 3 different colors and lengths denote 3 types of camera models. Their AoVs γ and ranges R are 60° and $20m$, 45° and $30m$, and, 30° and $45m$. Since the teaching building is too large, we discretize it into 22 parts and use 22 objects to denote them, as shown in Fig. 20. The distances between two



Figure 21: Captured Images of Three Algorithms

adjacent objects are $25m$ in the east-west direction and $30m$ in the south-north direction. To avoid collision with the building, UAVs fly in the outside air and keep $15m$ away from the objects.

Experimental Results. The locations of 22 objects, the UAV number 7 and each UAV's camera number 2, and 3 types of cameras' γ, R input three algorithms, *DoMino*, *CLGS*, and *CLRS*. The output arrangements of three algorithms are depicted in Fig. 20 with different labels. We use the number of monitored gates as a metric to compare the performance of three algorithms, and thus, the ground truth is 25. The captured images of each UAV following the outputted arrangements by three algorithms are shown in Fig. 21. In each image, we enlarge the captured gates bounded in rectangle boxes for clearness. The experimental results shows that *DoMino* monitors 15 gates while *CLGS* and *CLRS* only monitor 4 and 1, respectively. Therefore, *DoMino* achieves 65% of ground truth and outperforms *CLGS* and *CLRS* $4.25\times$ and $15\times$, respectively.

7 CONCLUSION

In this paper, we solve the problem of deployment of adjustable multi-camera UAVs for monitoring tasks. This is the first work for deployment problem with multi-camera model. The key technical depth of this paper is reducing the infinite decision space to a limited one with discretization and Maximal Set extraction technique, propose a polynomial time algorithm to tackle a two tightly-coupled NP-hard problem, and bound the constant approximation ratio with a tricky proof technique. Our evaluation results show that our algorithm achieve 65% of ground truth and outperforms comparison algorithms by at least $2.03\times$.

REFERENCES

- [1] 2016. DJI GUIDANCE Kit. <https://www.dji.com/cn/guidance/features>.
- [2] 2018. Djı Matrice 200 and 300 RTK. <https://www.dji.com/products/enterprise?site=brandsite&from=nav#drones>.
- [3] 2018. Skydio R1. <https://www.diyphotography.net/skydio-r1-4k-self-flying-drone-12-cameras-will-follow-anywhere/>.
- [4] 2021. Deploying Adjustable Multi-camera UAVs for Monitoring Tasks (Technical Report). <https://weijunalexwang.github.io/files/MOBIHOC21.pdf>
- [5] N/A. Inspection. <https://enterprise.dji.com>.
- [6] N/A. Waldo XCAM System. <http://www.waldoair.com/xcam-ultra.html>.
- [7] Fakhreddine Ababsa et al. 2019. 3D Human Tracking with Catadioptric Omnidirectional Camera. In *ACM ICMR*.
- [8] S. Aghajanzadeh et al. 2020. Camera Placement Meeting Restrictions of Computer Vision. In *IEEE ICIP*.
- [9] Y. Alsaba et al. 2018. Beamforming in Wireless Energy Harvesting Communications Systems: A Survey. *IEEE Communications Surveys Tutorials* (2018).
- [10] A. A. Altahir et al. 2020. Visual Sensor Placement Based on Risk Maps. *IEEE Transactions on Instrumentation and Measurement* (2020).
- [11] M. Amac Guvensan et al. 2011. On coverage issues in directional sensor networks: A survey. *Ad Hoc Networks* (2011).
- [12] V. Blanz et al. 2005. Face recognition based on frontal views generated from non-frontal images. In *IEEE CVPR*.
- [13] Stefan Dobrev et al. 2012. strong connectivity in sensor networks with given number of directional antenna of bounded angle. (2012).
- [14] Satoru Fujishige. 2005. Submodular functions and optimization. *Elsevier* (2005).
- [15] Giordano Fusco et al. 2010. Placement and Orientation of Rotating Directional Sensors. In *IEEE SECON*.
- [16] S. He et al. 2016. Full-View Area Coverage in Camera Sensor Networks: Dimension Reduction and Near-Optimal Solutions. *IEEE Transactions on vehicular technology* (2016).
- [17] Yitao Hu et al. 2014. Critical sensing range for mobile heterogeneous camera sensor networks. In *IEEE INFOCOM*.
- [18] R. Jacobson. 2000. The Manual of Photography 9th edition. *Focal Press* (2000).
- [19] F. Jiang et al. 2020. Distributed Optimization of Visual Sensor Networks for Coverage of a Large-scale 3-D Scene. *IEEE Transactions on Mechatronics* (2020).
- [20] Matthew P. Johnson et al. 2011. Pan and scan: Configuring cameras for coverage. In *IEEE INFOCOM*.
- [21] Niels Joubert et al. 2016. Towards a Drone Cinematographer: Guiding Quadrotor Cameras using Visual Composition Principles. *CoRR* (2016).
- [22] S. Jun et al. 2017. Camera Placement in Smart Cities for Maximizing Weighted Coverage With Budget Limit. *IEEE Sensors Journal* (2017).
- [23] M. Kan et al. 2016. Multi-view Deep Network for Cross-View Classification. In *IEEE CVPR*.
- [24] R. Karp. 1972. Reducibility Among Combinatorial Problems. In *Complexity of Computer Computations*.
- [25] Ajay Kaushik et al. 2019. A Grey Wolf Optimization Based Algorithm for Optimum Camera Placement. *Wireless Personal Communications* (2019).
- [26] E. Koyuncu. 2018. Performance Gains of Optimal Antenna Deployment in Massive MIMO Systems. *IEEE Transactions on Wireless Communications* (2018).
- [27] Chaoyang Li et al. 2016. On k-full-view-coverage-algorithms in camera sensor networks. In *IEEE ICCC*.
- [28] Fenghua Li et al. 2019. HideMe: Privacy-Preserving Photo Sharing on Social Networks. In *IEEE INFOCOM*.
- [29] Kin Sun Liu et al. 2016. Poster: Combinatorics, Algorithms and Systems for Sensor Deployment With Line-Of-Sight Constraints. In *ACM MobiHoc*.
- [30] Xiaolan Liu et al. 2016. Achieving Full-View Barrier Coverage with Mobile Camera Sensors. In *IEEE NaNA*.
- [31] Huan Ma et al. 2012. Minimum camera barrier coverage in wireless camera sensor networks. In *IEEE INFOCOM*.
- [32] Aaron Mavrinac et al. 2013. Modeling Coverage in Camera Networks: A Survey. *Springer International Journal of Computer Vision* (2013).
- [33] R. Garey Michael et al. 1979. Computers and intractability: a guide to the theory of NP-completeness. *WH Free. Co., San Fr* (1979).
- [34] Miura Miura et al. 2020. 3D human pose estimation model using location-maps for distorted and disconnected images by a wearable omnidirectional camera. *Transactions on Computer Vision and Applications* 12 (2020).
- [35] Vikram P. Munishwar et al. 2012. Coverage management for mobile targets in visual sensor networks. In *ACM MSWiM*.
- [36] Tobias Nageli et al. 2017. Real-time Planning for Automated Multi-view Drone Cinematography. *ACM Transactions on Graphics* (2017).
- [37] P. Rahimian et al. 2017. Optimal Camera Placement for Motion Capture Systems. *IEEE Transactions on Visualization and Computer Graphics* (2017).
- [38] Mac Schwager et al. 2011. Eyes in the Sky: Decentralized Control for the Deployment of Robotic Camera Networks. *Proc. IEEE* (2011).
- [39] T. Tran et al. 2018. Symmetric Connectivity Algotirthms in Multiple Directional Antennas Wireless Sensor Networks. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*.
- [40] W. Wang et al. 2019. PANDA: Placement of Unmanned Aerial Vehicles Achieving 3D Directional Coverage. In *IEEE INFOCOM*.
- [41] W. Wang et al. 2019. VISIT: Placement of Unmanned Aerial Vehicles for Anisotropic Monitoring Tasks. In *IEEE SECON*.
- [42] Yi Wang et al. 2014. SmartPhoto: a resource-aware crowdsourcing approach for image sensing with smartphones. In *ACM MobiHoc*.
- [43] Yi Wang and Guohong Cao. 2011. Barrier Coverage in Camera Sensor Networks. In *ACM MobiHoc*.
- [44] You-Chiun Wang et al. 2012. Using Rotatable and Directional (R & D) Sensors to Achieve Temporal Coverage of Objects and Its Surveillance Application. *IEEE Transactions on Monile Computing* (2012).
- [45] Zhibo Wang et al. 2014. Achieving k-Barrier Coverage in Hybrid Directional Sensor Networks. *IEEE Transactions on Monile Computing* (2014).
- [46] Y. Wu et al. 2017. Photo crowdsourcing for area coverage in resource constrained environments. In *IEEE INFOCOM*.
- [47] Q. Xu et al. 2018. Waveforming: An Overview With Beamforming. *IEEE Communications Surveys Tutorials* (2018).
- [48] Xincong Yang et al. 2018. Computer-Aided Optimization of Surveillance Cameras Placement on Construction Sites. *Wiley CACIE* (2018).
- [49] Senthil Yogamani et al. 2019. WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving. In *IEEE ICCV*.
- [50] Zuoming Yu et al. 2015. Local face-view barrier coverage in camera sensor networks. In *IEEE INFOCOM*.
- [51] R. Zhang et al. 2013. MIMO Broadcasting for Simultaneous Wireless Information and Power Transfer. *IEEE Transactions on Wireless Communications* (2013).

8 APPENDIX

8.1 Proof of Theorem 4.1

PROOF. Fix $f = f_0$. Without loss of generality, suppose that we have $r(k_2 - 1) < d \leq r(k_2)$ for a given distance d . As $Q(f_0, d)$ monotonically decrease with distance d , on one hand, $\frac{Q(f_0, d)}{Q(f_0, r(k_2))} =$

$$\frac{Q(f_0, d)}{Q(f_0, r(k_2))} \geq \frac{Q(f_0, r(k_2))}{Q(f_0, r(k_2))} = 1; \text{ on the other hand,}$$

$$\begin{aligned} \frac{Q(f_0, d)}{Q(f_0, r(k_2))} &= \frac{Q(f_0, d)}{Q(f_0, r(k_2))} \\ &\leq \frac{Q(f_0, r(k_2 - 1))}{Q(f_0, r(k_2))} \\ &\leq \frac{(b((1 + \epsilon_1)^{k_2/2} - 1) + b)^2}{(b((1 + \epsilon_1)^{(k_2-1)/2} - 1) + b)^2} \\ &= 1 + \epsilon_1, \end{aligned}$$

the second inequality hold because substitute QoM (Equ. (2)) and the expression of $r(k_2)$. Thus, the result follows. \square

8.2 Proof of Theorem 4.2

PROOF. Without loss of generality, suppose $r(k_2 - 1) < d \leq r(k_2)$ for a given distance d and $l(k_1 - 1) < f \leq l(k_1)$ for a given focal length f . As $Q(f, d)$ monotonically increase with focal length f and decrease with distance d , on one hand, $\frac{Q(f, d)}{Q(f, d)} = \frac{Q(f, d)}{Q(l(k_1 - 1), r(k_2))} \geq$

$$\frac{Q(f, r(k_2))}{Q(l(k_1 - 1), r(k_2))} \geq \frac{Q(l(k_1 - 1), r(k_2))}{Q(l(k_1 - 1), r(k_2))} = 1; \text{ on the other hand,}$$

$$\begin{aligned} \frac{Q(f, d)}{Q(f, d)} &= \frac{Q(f, d)}{Q(l(k_1 - 1), r(k_2))} \\ &\leq \frac{Q(l(k_1), r(k_2 - 1))}{Q(l(k_1 - 1), r(k_2))} \\ &= \frac{Q(l(k_1), r(k_2 - 1))}{Q(l(k_1 - 1), r(k_2 - 1))} \cdot \frac{Q(l(k_1 - 1), r(k_2 - 1))}{Q(l(k_1 - 1), r(k_2))} \\ &\leq \frac{a \cdot (((1 + \epsilon_2)/(1 + \epsilon_1))^{k_1/2})^2}{a \cdot (((1 + \epsilon_2)/(1 + \epsilon_1))^{(k_1-1)/2})^2} \cdot (1 + \epsilon_1) \\ &= \left(\frac{1 + \epsilon_2}{1 + \epsilon_1}\right) \cdot (1 + \epsilon_1) = 1 + \epsilon_2. \end{aligned}$$

The second inequality holds because substitute QoM (Equ. 2), the expression of $l(k_1)$, and the conclusion of Theo. 4.1. Thus, the result follows. \square

8.3 Proof of Theorem 4.3

PROOF. We prove this theorem by constructing the extreme cases. Fig. 22 and Fig. 23 illustrates two examples of extreme cases. Without loss of generality, given \tilde{Q} , we have $U(\tilde{Q}) = \int_0^{2\pi} \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, \varphi) d\varphi$ and $\tilde{U}(\tilde{Q}) = \Delta A \cdot \sum_{t=1}^{\frac{2\pi}{\Delta A}} \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A)$. In the best case, as a toy example shown by Fig. 22, $\tilde{U}(\tilde{Q})$ is at most the value of $U(\tilde{Q})$, i.e., every covered θ exactly covers the integral multiple of ΔA . In such case, $\frac{U(\tilde{Q})}{\tilde{U}(\tilde{Q})} = 1$. In the worst case, in which θ is very close to the integral multiple of ΔA and there is no overlapped covered aspects. Formally, $\forall i \leq K, i \in \mathbb{Z}^+, K \leq 2 \times \frac{2\pi}{\theta}, \beta_i \rightarrow 0$, e.g., $K = 6$ in Fig. 23. Let $\Phi(x) \triangleq \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, x)$, then in the worst case,

$$U(\tilde{Q}) = \int_0^{2\pi} \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, \varphi) d\varphi = \sum_{t=1}^{\frac{2\pi}{\theta}} \frac{\theta}{\Delta A} \cdot \Delta A \cdot \Phi(t\Delta A)$$

because the limit subjects to $\beta_i \rightarrow 0$ and there are at most K (constant) β_i s, namely, the $\frac{K}{2}$ intervals from $(t - 1)\Delta A + \beta_i$ to $t\Delta A$ and $\frac{K}{2}$ intervals from $(t - 1)\Delta A$ to $t\Delta A - \beta_i$ all cover complete aspects in ΔA . Additionally, $\tilde{U}(\tilde{Q}) = \Delta A \cdot \sum_{t=1}^{\frac{2\pi}{\Delta A}} \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A) = \sum_{t=1}^{\frac{2\pi}{\theta}} (\frac{\theta}{\Delta A} - 2) \cdot \Delta A \cdot \Phi(t\Delta A)$ because the $\frac{K}{2}$ intervals from $(t - 1)\Delta A + \beta_i$

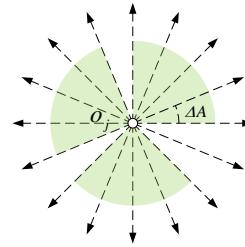


Figure 22: Best Case

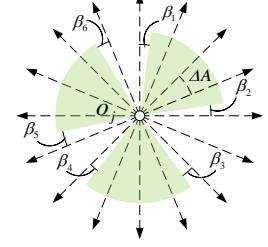


Figure 23: Worst Case

to $t\Delta A$ and $\frac{K}{2}$ intervals from $(t - 1)\Delta A$ to $t\Delta A - \beta_i$ all cover zero aspects in ΔA . Thus, we have

$$\frac{U(\tilde{Q})}{\tilde{U}(\tilde{Q})} = \frac{\sum_{t=1}^{\frac{2\pi}{\theta}} \frac{\theta}{\Delta A} \cdot \Delta A \cdot \Phi(t\Delta A)}{\sum_{t=1}^{\frac{2\pi}{\theta}} (\frac{\theta}{\Delta A} - 2) \cdot \Delta A \cdot \Phi(t\Delta A)} = 1 + \frac{2\Delta A}{\theta - 2\Delta A} = 1 + \epsilon_\Delta.$$

\square

8.4 Proof of Theorem 4.4

PROOF. We first prove the upper bound of the number of discretized area on 2D plane by M uniform sectors intersecting with each other in the following lemma.

LEMMA 8.1. *The number of discretized area on 2D plane by n uniform sectors intersecting with each other is at most $5n^2 - 5n + 2$.*

PROOF. First, we analyze the relationship between the number of cells and that of intersection points. Obviously, if there are three or more edges or arcs intersecting at same point, the number of cells must not be maximized. Thus, consider the condition that there are only two edges or arcs intersecting at one point, then one intersection point divides each edge into two parts, i.e., the total number of added edges is 2 times that of intersection point. Let e denote the initial total number of edges, v denote the initial total number of vertices, f denote the initial total number of cells, (i.e., faces in Graph Theory), and x denote the added intersection point. Due to the Euler characteristic⁴, we have $f = e - v + 2 = (e + 2x) - (v + x) + 2 = x + 2$.

Furthermore, we observe that when the radian of sector is in $(\pi/2, \pi)$, there are the most intersection points for two sectors intersects with each other, i.e., 10 intersection points. Thus, any pair among n sectors intersect at 10 different points, and there are at most $10 \cdot \binom{n}{2} = 5n^2 - 5n$ intersection points. By $f = x + 2$, the total number of cell is at most $5n^2 - 5n + 2$. \square

Given N objects, there are at most $\frac{2\pi}{\Delta A} \cdot NK_2$ number of sectors. By Lem. 8.1, Theo. 4.1, and Theo. 4.3, we have $\frac{2\pi}{\Delta A} = O(\epsilon_\Delta^{-1})$ and $K_2 = O(\epsilon_1^{-1})$. Substitute $O(\epsilon_\Delta^{-1})$ and $O(\epsilon_1^{-1})$ into upper bound in Lem. 8.1, the number of cells is at most $5 \times \left(\frac{2\pi}{\Delta A} \cdot NK_2\right)^2 - 5 \times \left(\frac{2\pi}{\Delta A} \cdot NK_2\right) + 2 = O(N^2 \epsilon_1^{-2} \epsilon_\Delta^{-2})$. Therefore, the number of cells is $O(N^2 \epsilon_1^{-2} \epsilon_\Delta^{-2})$. \square

8.5 Proof of Theorem 4.5

PROOF. Def. 4.4 defines that the candidate locations satisfy at least one of its cameras monitoring a maximal set. Thus, We first prove the necessary conditions for one camera monitoring a maximal set.

⁴D.S. Jeanine. Euler's Gem: The Polyhedron Formula and the Birth of Topology, Elsevier, Volume 6, 2008

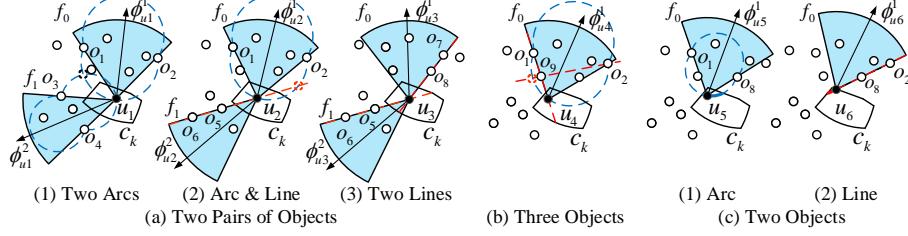


Figure 24: Candidate Locations Extraction (Copy Figure 11 to here for convenient illustration in proof)

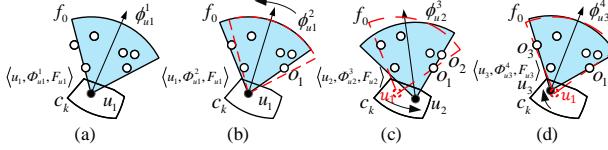


Figure 25: Auxiliary Figure for Theorem 4.5

LEMMA 8.2. *The two necessary conditions for one camera monitoring a maximal set are: 1. two objects located on the same one radius; 2. two objects respectively located on two radii.*

PROOF. Start from an arbitrary subarrangement $\langle u_1, \phi_{u1}^1, f_0 \rangle$ of u_1 as shown in Fig. 25(a). First, keep the location u_1 fixed and rotate its direction ϕ_{u1}^1 anticlockwise until there is at least one object, e.g., o_1 in Fig. 25(b), hitting on the right radius. In this state, the direction of u_1 changes to ϕ_{u1}^2 . Suppose there is no object hitting on the radii of other sectors of u_1 during the transformation, then current arrangement $\langle u_1, \phi_{u1}^2, f_{u1} \rangle$ dominates the previous one $\langle u_1, \phi_{u1}^1, f_{u1} \rangle$. Second, keep o_1 sliding on the right radius and move u_2 until at least another object, e.g., o_2 in Fig. 25(c) and o_3 in Fig. 25(d), hitting on one radius. After this step, u_1, ϕ_{u1}^2 changes to u_2, ϕ_{u2}^3 or u_3, ϕ_{u3}^4 , respectively. During the whole transformation, objects won't fall out but only be added into sector because the transformation keeps the critical objects sliding on the radius. Therefore, after the transformation, both $\langle u_2, \phi_{u2}^3, f_{u2} \rangle$ and $\langle u_3, \phi_{u3}^4, f_{u3} \rangle$ dominate $\langle u_1, \phi_{u1}^2, f_{u1} \rangle$. Thus, we obtain the two necessary conditions for one camera monitoring a maximal set. \square

Apparently, we shall stay at one of following three statuses after above transformation. We use *Case a-b* to denote the case *b* under *a* sector(s) status. **Case 1-1:** Three objects hit on the radii. In this case, there is the only candidate location, such as Fig. 24(b), corresponding to Step 8-10 in Alg. 1. **Case 1-2:** Two objects hit on the radii. In this case, the center of the sector can slide on the arc or straight line, such as Fig. 24(c), corresponding to Step 11-13 in Alg. 1. If during the sliding there are other objects hitting on radius, it falls into the scope of Case 1-1. **Case 1-3:** One object hits on the radius. In this case, the center of the sector can be located at any point in the cell, corresponding to Step 14 in Alg. 1.

Now, we consider that *there are object(s) hitting on the radii of other sectors during above transformation*. By one-sector analysis, the total cases of two sectors are the combinations of above one-sector condition, i.e., 2^3 cases. Yet, they can combine to 4 cases. **Case 2-1:** As long as three objects hit on the radii of one sector. This sector can determine the only location, then it falls into the scope of Case 1-1. **Case 2-2:** Two pairs of objects hit on the radii of two sectors. There are three subcases, but each of them can determine the only location, such as Fig. 24(a), which corresponds to Step 2-7 in Alg. 1. **Case 2-3:** One pair objects hit on the radius of one sector and an other object hits on the radius of the other. In this case, the center

of the sector can slide on the arc or straight line while keeping the individual object sliding on the radius, so this case falls into the scope of Case 1-2. If during the sliding, there are other objects hitting on radius, it falls into the scope of Case 2-1 or Case 2-2. **Case 2-4:** Two objects hit on the radius of each sector, respectively. This case falls into the scope of Case 1-3. All the cases of more than two sectors fall into aforementioned cases. Consequently, Λ contains all the candidate locations. \square

8.6 Proof of Lemma 4.2

PROOF. According to the Definition 4.5, we need to check the three listed requirements of $f(X)$ to prove it is monotone and submodular. For ease of analysis, define

$$\tau(X, j) = \sum_{t=1}^{\frac{2\pi}{\Delta A}} (\max_{\langle u_i, \phi_{ui}^k, f_{ui}^k \rangle \in X} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A)). \quad (6)$$

First, when the number of UAVs is 0, obviously $\tau(X, j) = 0$, thus, we have $f(\emptyset) = 0$.

Second, let A be a set of arrangements in Γ , $e \in \Gamma \setminus A$. Then, $\tau(X \cup e, j) - \tau(X, j) \geq 0$ since: 1. the Σ function is non-decreasing for $\cdot \geq 0$; 2. max function is also non-decreasing, i.e., $\max_{\langle u_i, \phi_{ui}, f_{ui} \rangle \in X \cup \{e\}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A) - \max_{\langle u_i, \phi_{ui}, f_{ui} \rangle \in X} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A) \geq 0$ to any specific t . Therefore,

$$f(A \cup \{e\}) - f(A) = \frac{1}{N} \sum_{j=1}^N (\tau(A \cup \{e\}, j) - \tau(A, j)) \geq 0.$$

Third, let A and B be two sets of arrangements in Γ where $A \subseteq B \subseteq \Gamma$, and $e \in \Gamma \setminus B$. Recall the definition of $\tau(X, j)$ in Eq. (6), we have

$$\tau(X \cup \{e\}, j) - \tau(X, j) =$$

$$\begin{cases} 0, & \tilde{Q}(u_e, o_e, f_{ue}^k, \phi_{ue}^k, t_e \Delta A) \leq \tilde{Q}_{\max}(X), \\ \tilde{Q}(u_e, o_e, f_{ue}^k, \phi_{ue}^k, t_e \Delta A) - \tilde{Q}_{\max}(X), & \text{otherwise}. \end{cases} \quad (7)$$

where $\tilde{Q}_{\max}(X) = \max_{\langle u_i, \phi_{ui}, f_{ui} \rangle \in X} \tilde{Q}(u_i, o_e, f_{ui}^k, \phi_{ui}^k, t_e \Delta A)$, i.e., the maximum QoM provided by some arrangements in X who monitoring the same aspect $t_e \Delta A$ of the same object o_e as arrangement e . We observe that: (1) for given e , its QoM $\tilde{Q}(u_e, o_e, \phi_{ue}^k, t_e \Delta A)$ is a constant; (2) $\tilde{Q}_{\max}(A) \leq \tilde{Q}_{\max}(B)$ since $A \subseteq B$. Then, we use the proof by cases.

Case 1: $\tilde{Q}(u_e, o_e, f_{ue}^k, \phi_{ue}^k, t_e \Delta A) \leq \tilde{Q}_{\max}(A)$. By Eq. (7), we have $\tau(A \cup \{e\}, j) - \tau(A, j) = 0$. The observation (2) implies that $\tilde{Q}(u_e, o_e, f_{ue}^k, \phi_{ue}^k, t_e \Delta A) \leq \tilde{Q}_{\max}(A) \leq \tilde{Q}_{\max}(B)$. Thus, by Eq. (7), $\tau(B \cup \{e\}, j) - \tau(B, j) = 0$. Therefore, $\tau(A \cup \{e\}, j) - \tau(A, j) = \tau(B \cup \{e\}, j) - \tau(B, j)$.

Case 2: $\tilde{Q}(u_e, o_e, f_{ue}^k, \phi_{ue}^k, t_e \Delta A) > \tilde{Q}_{\max}(A)$. According to Eq. (7), we have $\tau(A \cup \{e\}, j) - \tau(A, j) = \tilde{Q}(u_e, o_e, f_{ue}^k, \phi_{ue}^k, t_e \Delta A) - \tilde{Q}_{\max}(A)$. By observation (2), we have $\tilde{Q}_{\max}(A) \leq \tilde{Q}_{\max}(B)$, which implies

$\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) \leq \tilde{Q}_{max}(B)$ or $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) > \tilde{Q}_{max}(B)$.

Case 2.1: $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) \leq \tilde{Q}_{max}(B)$. Based on Eq. (7), $\tau(B \cup \{e\}, j) - \tau(B, j) = 0$. Thus, $\tau(A \cup \{e\}, j) - \tau(A, j) \geq \tau(B \cup \{e\}, j) - \tau(B, j)$ since $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) > \tilde{Q}_{max}(A) > 0$.

Case 2.2: $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) > \tilde{Q}_{max}(B)$. Based on Eq. (7), $\tau(B \cup \{e\}, j) - \tau(B, j) = \tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) - \tilde{Q}_{max}(B)$. Furthermore, according to the observation (1) and (2), we have $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) - \tilde{Q}_{max}(A) \geq \tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) - \tilde{Q}_{max}(B)$, i.e., $\tau(A \cup \{e\}, j) - \tau(A, j) \geq \tau(B \cup \{e\}, j) - \tau(B, j)$.

Because the $\tau(A \cup \{e\}, j) - \tau(A, j) \geq \tau(B \cup \{e\}, j) - \tau(B, j)$ holds in all cases, we have

$$\begin{aligned} & (f(A \cup \{e\}) - f(A)) - (f(B \cup \{e\}) - f(B)) \\ &= \frac{1}{N} \sum_{j=1}^N ((\tau(A \cup \{e\}, j) - \tau(A, j)) \\ &\quad - (\tau(B \cup \{e\}, j) - \tau(B, j))) \geq 0. \end{aligned}$$

To sum up, we can conclude that $f(X)$ in **P3** is monotone and submodular. In addition, the constraint of **P3** is clear a uniform matroid constraint. Thus, proof completes. \square

8.7 Analysis of Time Complexity

PROOF. First, according to Theo. 4.4, DoMino discretizes the entire space into $O(N^2 \epsilon_1^{-2} \epsilon_\Delta^{-2})$ cells. To generate cells, it takes $O(1)$ time to computer all intersecting points among NK_2 circles and $N\Delta A$ line segments, and thus, it totally takes $O(N^2 \epsilon_1^{-1} \epsilon_\Delta^{-1})$ time (Please refer to Theo 4.1 for the relation between K_2 and ϵ_1 , and Theo 4.3 for the relation between ΔA and ϵ_Δ^{-1}). Then, Alg. 1 extracts all candidate locations and maximal set in each cell. In each cell, Alg. 1 traverses all combinations of four objects with all combinations of two types of focal lengths (Step 2-7), traverses all combinations of three objects with all types of focal lengths, and traverses two objects with all types of focal lengths (Step 8-13), and randomly select a point (Step 14). Thus, Alg. 1 takes no more than $O(N^2 \epsilon_1^{-2} \epsilon_\Delta^{-2}) \times ((\frac{N}{4}) \cdot (\frac{K_1}{2}) + (\frac{N}{3}) \cdot K_1 + (\frac{N}{2}) \cdot K_1) = O(N^6 \epsilon_1^{-2} \epsilon_\Delta^{-4})$ time (Please refer to Theo 4.2 for the relation between K_1 and ϵ_2). Next, Alg. 2 extracts subarrangements at each candidate location. At each candidate location, it takes $O(1)$ time to extract constant subarrangements, and hence, it totally takes $O(N^6 \epsilon_1^{-2} \epsilon_\Delta^{-4})$ time. At last, Alg. 3 greedily selects M locations from $O(N^6 \epsilon_1^{-2} \epsilon_\Delta^{-4})$ locations, and at each of M locations greedily selects K subarrangements from constant ones. Thus, Alg. 3 takes $O(KMN^6 \epsilon_1^{-2} \epsilon_\Delta^{-4})$. According to Theo. 4.6, we have $\epsilon = \Theta(\epsilon_2 \epsilon_\Delta)$, and by substitution, the time complexity of DoMino algorithm is $O(KMN^6 \epsilon^{-4})$, \square