Greedy Perspectives: Multi-Drone View Planning for Collaborative Perception in Cluttered Environments

Krishna Suresh[1], Aditya Rauniyar[2], Micah Corah[3], and Sebastian Scherer[2]

***Abstract*— Deployment of teams of aerial robots could enable large-scale filming of dynamic groups of people (actors) in complex environments for applications in areas such as team sports and cinematography. Toward this end, methods for submodular maximization via sequential greedy planning can enable scalable optimization of camera views across teams of robots but face challenges with efficient coordination in cluttered environments. Obstacles can produce occlusions and increase chances of inter-robot collision which can violate requirements for near-optimality guarantees. To coordinate teams of aerial robots in filming groups of people in dense environments, a more general view-planning approach is required. We explore how collision and occlusion impact performance in filming applications through the development of a multi-robot multi-actor view planner with an occlusion-aware objective for filming groups of people and compare with a formation planner and a greedy planner that ignores inter-robot collisions. We evaluate our approach based on five test environments and complex multi-actor behaviors. Compared with a formation planner, our sequential planner generates 14% greater view reward for filming the actors in three scenarios and comparable performance to formation planning on two others. We also observe near identical view rewards for sequential planning both with and without inter-robot collision constraints which indicates that robots are able to avoid collisions without impairing performance in the perception task. Overall, we demonstrate effective coordination of teams of aerial robots in environments cluttered with obstacles that may cause collisions or occlusions and for filming groups that may split, merge, or spread apart. Our implementation and the data used to produce results for this paper are available via the companion website: https://greedyperspectives.github.io/**

I. Introduction

The capture of significant events via photos and video has become universal, and Unmanned aerial vehicles (UAVs) extend the capabilities of cameras by allowing for view placement in otherwise hard-to-reach places and tracking intricate trajectories. Multiple aerial cameras can be used to not only view an actor from multiple angles simultaneously but perform higher functions such as localization and tracking [1–5], environment exploration and mapping [3, 6], cinematic filming [7–10], and outdoor human pose reconstruction [11, 12]. These applications rely on effective collaboration between groups of UAVs whereas manual control may result in poor shot selection and view duplication while requiring many coordinated operators. Therefore, autonomous coordination of UAV teams may be necessary for tasks such as multi-robot filming or reconstruction. However, directly maximizing domain-specific metrics, such as reconstruction accuracy, can be difficult to perform online—this motivates development of proxy objectives that quantify coverage and detail for multiple views. For example, Bucker et al. [7] demonstrate cinematic filming through a joint objective combining collision and occlusion avoidance, shot diversity, and artistic principles in filming a *single actor*. We are interested in similar settings but where robots collaborate to obtain diverse views of *multiple actors*—in a cluttered environment, with occlusions, where robots may observe multiple actors at once.

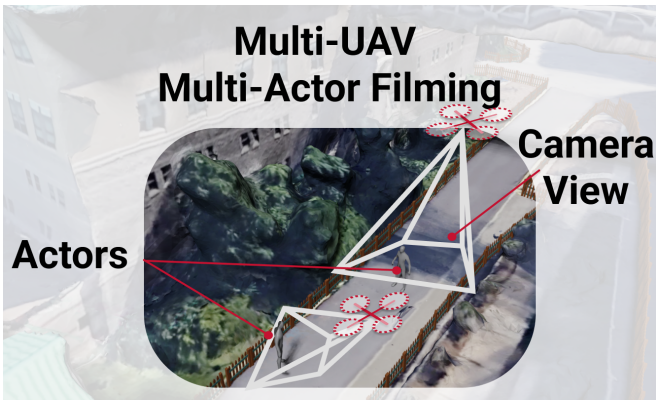

Fig. 1: Multi-Actor View Planning Scenario: Known actor and environment geometries as well as actor trajectories and robot start locations are input into the view-planning system. The planner aims to maximize coverage-like view reward for all actors for the duration of the planning horizon. Mesh of CMU campus from [13].

While defining an objective can be difficult, planning for multi-robot aerial systems also presents a challenge: the vast joint state space, non-convex environment, and non-linear view rewards make optimal planning intractable. Many applications exploit problem-specific structures to reduce the problem complexity such as by optimizing an actor-centric formation [4, 11, 14] or by altering the search procedure to generate single-robot trajectories sequentially [1–3, 6, 7]. In this work, we apply a planning approach much like Bucker et al. [7], and develop a system design and view rewards that enable application to a multi-actor setting.

Problem: The dynamic multi-actor view planning problem consists of generating sequences of camera views over a fixed planning horizon to maximize a collective view reward

[1]K. Suresh is with Olin College of Engineering, Needham, MA, USA ksuresh@olin.edu

[2]A. Rauniyar, and S. Scherer are with the Robotics Institute, School of Computer Science at Carnegie Mellon University, Pittsburgh, PA, USA {rauniyar, basti}@cmu.edu

[3]M. Corah is with the Department of Computer Science at the Colorado School of Mines, Golden, CO, USA micah.corah@mines.edu

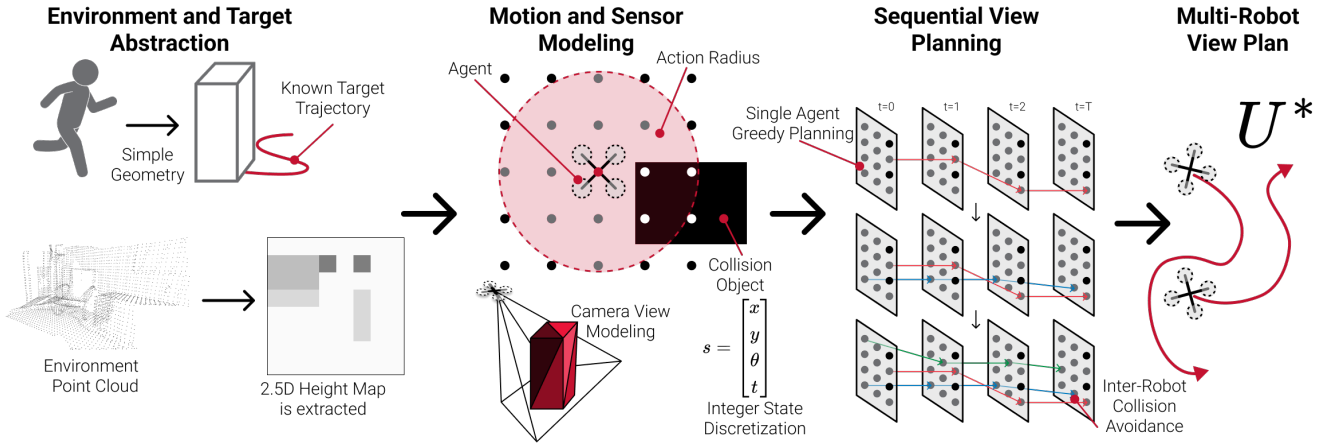

Fig. 2: View Planning System Overview The multi-actor scenario is translated to an internal planner representation. The Markov Decision Process with DAG structure encodes collision constraints and view rewards. The Multi-Robot View Plan is produced through sequential greedy planning.

that is a function of pixel densities over the surfaces of the actors. The primary assumptions for our approach are: *known static environment*, *known actor trajectories* (e.g. scripted scenarios), and *known robot start state*. An illustration of the problem setup is depicted in Fig. 1. These assumptions are fairly strict for the purpose of evaluating the planning approach. In practice, our approach could be applied in a receding-horizon setting based on scripted or predicted actor trajectories [7].

Contributions: The main contributions of this work are summarized as:

- An occlusion-aware objective based on rendered camera views for filming groups of people
- Implementation of a collision-aware multi-robot, multi-actor view planner
- Evaluation of the view planner in scenarios with challenging obstacles, occlusions, and group behaviors

Our contributions build on prior work developing perception objectives based on pixel densities by Jiang and Isler [15] and work by Hughes et al. [16] developing objectives for submodular multi-robot settings. This work presents a variant of such objectives that is occlusion-aware, and we present results for scenarios with a wide variety of obstacles and occlusions. These contributions toward objective design also enable us employ a planning approach similar to Bucker et al. [7] but for filming multiple actors.

II. Related Works

Aerial Filming: Aerial perception systems have grown to widespread use through their success in providing low-cost filming of conventionally challenging unscripted scenes. Consumer and commercial systems such as the Skydio S2+ [17] demonstrate single-drone filming capabilities and are starting to incorporate collaborative multi-drone behaviors for mapping. Research developing autonomous aerial filming systems for cinematography has also focused on developing systems that can perform parametrized actions such as tracking a subject, rule of thirds framing, or dolly zoom by methods such as model predictive control to optimize motion and camera controls [9, 10] and by scheduling multi-robot missions [18–21]. Another body of work focuses on learning artistic principles rather than implementing them by hand [22, 23]. The focus of these works on mission planning and learning artistic principles is complementary to our approach which develops a capability for filming complex group behaviors with occlusions.

A few works address this challenge of filming individual or group behavior from multiple perspectives [7, 24]. Xu et al. [24] present an actor-centered controller based on Voronoi coverage over a hemisphere, and Bucker et al. [7] describe an approach where robots plan trajectories based on a spherical discretization centered on the actor. A limitation of these approaches is that assigning robots to actors does not exploit the robots' capacity to observe multiple actors at once.

Motion capture and dynamic scenes: Aerial motion capture [11, 12, 14, 25] such as to reconstruct the motion of a moving person is closely related to filming. So far, these systems consist of a groups of robots that observe a single moving subject with various degrees of awareness of obstacles or occlusions. However, all specify either an actor-centric policy [25] or a formation [11, 12, 14]. Although actor-centric planning can reduce the search space to orienting the formation versus planning for robots individually, this choice limits relevance to multi-actor settings. The view planning approach we present could enable robot teams to reconstruct motions of complex group behaviors in environments with varied obstacles and occlusions.

Reconstruction of static scenes: Our view planning approach also bears similarity to methods for reconstruction of static scenes [15, 26, 27]. Like these works, our approach emphasizes design of a view reward and optimizing paths to maximize that reward. In particular, we draw on the approach by Jiang and Isler [15] which reasons about pixel densities.

Target tracking and localization: The filming scenario we study is similar in composition to target localization and tracking problems that focus on estimating position or motion of one or more targets [1–5]. While these works sometimes reason about field-of-view and occlusions [4] this reasoning

is often limited and secondary to the tracking task—a task that is often formulated in terms of minimizing uncertainty in target state via an information-theoretic objective [1–3, 5]. Target tracking may also be thought of as representing a subsystem of a hypothetical filming system that relaxes our requirement for known trajectories—from this perspective, we observe that several works in this area apply sequential and auction-based methods for coordination that are generally compatible with our approach [1–3, 5].

Sequential and Submodular Multi-Robot Planning: Typically multi-robot perception planning and information gathering problems, cannot tractably be solved optimally due to their combinatorial nature, but greedy methods for submodular optimization can often promise information gain or perception quality no worse than half of optimal in polynomial time [28, 29]. Submodular optimization and sequential greedy planning has been applied extensively to such multi-robot coordination problems [1–3, 6, 7, 30–32]. However, questions of occlusions and camera views have been explored primarily in the setting of mapping and exploration [3, 6]. Lauri et al. [31] present an exception in which eye-in-hand cameras inspect and reconstruct static scenes. Unlike exploration and mapping, applications involving filming moving actors can force persistent interaction between robots over the duration of a scenario or planning horizon, and our early work on this topic indicates that sequential planning is important for effective cooperation in settings involving observing moving subjects [33].

III. Preliminaries

We will begin with some background regarding submodular and greedy planning:

A. Submodularity and Monotonicity

The view reward we employ satisfies monotonicity properties that are useful in developing our approach to planning and coordination. Informally, submodularity expresses the principle of diminishing returns and monotonicity requiring functions to be always increasing. Given a set of actions Ω, a set function g maps subsets of actions (robots' plans) to a real value (the view reward). A set function is monotonic if adding elements to a set does not decrease its value; that is for any $A \subseteq B \subseteq \Omega$ then $g(A) \leq g(B)$. A set function is submodular if marginal gains decrease monotonically; specifically, given $A \subseteq B \subseteq \Omega$ and $C \subseteq \Omega \setminus B$, then $g(A \cup C) - g(A) \geq g(B \cup C) - g(B)$. Objectives related to perception planning [31, 34] and information gathering [30] are often submodular, and this corresponds to how marginal view rewards may diminish given by repeated views of the same actor from the same perspective.

B. Partition Matroids

A partition matroid can be used to represent product-spaces of actions or trajectories that arise in multi-robot planning problems [35, Sec. 39.4]. Consider a view planning problem involving a team of robots $\mathcal{R} = \{1, \ldots, N^{\mathrm{r}}\}$ where each robot $i \in \mathcal{R}$ has access to a set of actions $\mathscr{U}_i$. These actions can take many forms such as assignments, trajectories, or paths. The set of all actions for a robot is the ground set $\Omega = \bigcup_{i \in \mathcal{R}} \mathscr{U}_i$. Each robot is assigned one action from its corresponding set $\mathscr{U}_i$. If there are no collisions between robots the set of valid and partial assignments forms a *partition matroid*: $\mathscr{I} = \{X \subseteq \Omega \mid 1 \geq |X \cap \mathscr{U}_i| \ \forall i \in \mathcal{R}\}$. To satisfy this structure each robots' actions must be interchangeable to satisfy the *exchange property* of a matroid. Inter-robot collisions violate this property because swapping actions can cause conflicts with other robots.

IV. Problem Formulation

We aim to coordinate a team of UAVs to maximize coverage-like view rewards for observing a group of actors moving through an obstacle-dense environment. Consider a set of actors $\mathcal{A} = \{1, \ldots, N^{\mathrm{a}}\}$ each with a set of faces $\mathcal{F}_a = \{1, \ldots, N_a^{\mathrm{f}}\}$ where $a \in \mathcal{A}$ and a set of robots $\mathcal{R} = \{1, \ldots, N^{\mathrm{r}}\}$. Each robot $i \in \mathcal{R}$ can execute a control action $u_{i,t} \in \mathcal{U}_i \subseteq SE(2)$ at time $t \in \{0, \ldots, T\}$. The robots go on to select a finite-horizon sequence of viable control actions that form their plans. Additionally, robots have associated states $x_{i,t} \in \mathcal{X}$ which is a subset of $SE(2)$. Sequences of states form the robots' trajectories $\xi_i = [x_{i,0}, \ldots, x_{i,T}]$—we will occasionally index trajectories to obtain $\xi_{i,t} = x_{i,t}$. Each trajectory, once fixed, produces non-collision constraints for all other robots. Given the trajectories of all actors in $SE(3)$, start states $x_{i,0}$, and environment geometry, we aim to find joint collision-free control sequences $U^* = \bigcup_{i \in \mathcal{R}} [u_{i,0}, \ldots, u_{i,T}]$ that maximize our objective and fit our motion model.

A. Motion Model

State transitions for each robot are specified by the following motion model

$$x_{i,t+1} = f_i(x_{i,t}, u_{i,t}) \tag{1}$$

where f_i is defined to only allow collision-free motions to positions and orientations within a constant velocity constraint. Given the time step duration, maximum translational and rotational velocities are converted to bounds on rotation and Euclidean distance as illustrated by Fig. 2.

B. Non-Collision Constraints

Robots are considered in collision with the environment when the discretized state location is occupied by an element of the environment map that exceeds the robot's height. Similarly, a pair of robots is in collision when both occupy the same discretized cell at the same time. We implicitly assume a conservative model of the environment (the height-map and discretization of the state space) to ensure safety of states that satisfy the obstacle and inter-robot collision constraints.

C. Camera and View Reward Model

We use a coverage-like reward based on pixel densities as a proxy for effectively observing an actor. Inspired by [15], we compute rewards based on cumulative pixel densities ($\frac{px}{m^2}$)

for observations of faces from polyhedral representations of each actor j. We define a function $\texttt{pixels}(x_{i,t}, t, j, f) \to \mathbb{R}$ which returns the pixel density for actor-j's face f when observed from a robot's state at time t.

In order to encourage robots to assume views that uniformly cover the actors and their corresponding faces, we apply a square root to introduce diminishing returns on increasing pixel densities from multiple views [16]. Finally, given robot trajectories according to the dynamics (1) and selected control inputs, the robots obtain the following view reward for the given face and time:

$$R_{\text{v}}(t, j, f) = \sqrt{\sum_{i \in \mathcal{R}} \texttt{pixels}(x_{i,t}, t, j, f)}. \tag{2}$$

The formal statement of monotonicity and submodularity properties for rewards of this form and the relationship to coverage are the subject of [16]. Intuitively, (2) obtains these desirable properties because the square-root is one of many real functions that increases monotonically but at ever-decreasing rates.[1] If not constrained through the selection of camera views, the submodularity of our objective based on (2) would ensure that rewards would be maximized by distributing all pixels approximately uniformly over the faces of the actor models. Likewise, submodularity due to the square-root encourages robots to distribute their views evenly across the actors and their surfaces. By contrast, summing pixel densities without the square-root would assign the same reward for distributing all pixels on one face or for distributing pixels uniformly. Our approach also allows for more variation in rewards than for simply thresholding on range or pixel density.

D. Objective

In addition to maximizing the view reward, we add a reward for stationary behavior $R_s(u_{i,t})$ to reduce unnecessary movement whereas

$$R_s(u) = \begin{cases} \epsilon & \text{if } u \text{ is stationary} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

So, robot $i \in \mathcal{R}$ obtains $\sum_{t \in \{0,\dots,T\}} R_s(u_{i,t})$ reward for time-steps it remains stationary. The joint objective is then as follows:

$$\mathcal{J}(X_{\texttt{init}}, U) = \sum_{t \in \{0,\dots,T\}} \left(\sum_{i \in \mathcal{R}} R_{\text{s}}(u_{i,t}) + \sum_{j \in \mathcal{A}} \sum_{f \in \mathcal{F}_j} R_{\text{v}}(t, j, f)) \right) \tag{4}$$

where $X_{\texttt{init}} = [x_0, \dots, x_{N^{\text{r}}}]$ is an array of initial robot states and U represents the robots' sequences of control actions. Since we aim to find the control sequence that maximizes this objective, our optimal control sequence can be defined as:

$$U^* = \arg\max_U \mathcal{J}(X_{\texttt{init}}, U) \tag{5}$$

[1]In fact, any other real function with similar monotonicity properties other than the square-root would satisfy the requirements of submodularity and monotonicity. However, we will not focus on the choice amongst such functions in this work.

V. Multi-Robot Multi-Actor View Planning

We now present our multi-drone view planning approach. The planner aims not only to produce sufficient coverage over the actors but also to exploit problem structure to efficiently find single-robot trajectories by greedy planning.

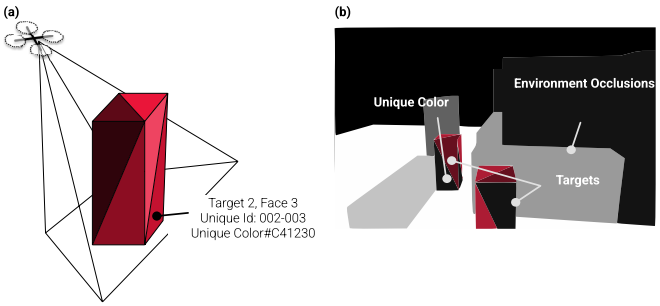

Fig. 3: Actor Coverage (a) UAV camera model frustum observing a simplified actor geometry. Actor faces are colored slightly differently based on a face identification system to allow for pixel density computation. **(b)** Example camera output from OpenGL internal rendering system.

A. Evaluation of View Reward

To produce an occlusion-aware implementation of our view reward (2) we compute `pixels` by implementing an OpenGL rasterization renderer based on a 2.5D height map of the environment and simplified actor geometry—we use polygonal cylinders. We then use a perspective camera based on specified camera intrinsics to capture an occlusion-aware representation of the scene from a given robot state. The system renders the environment via the GPU with a geometry shader to draw the height-map. To determine how many pixels the cameras observe for each face, we render the faces with unique colors associated with the actor and face IDs. Counting pixels of each color and dividing by the areas of the faces yields the corresponding pixel densities. Figure 3 illustrates this process and provides an example of a rendered view.

B. Single-Robot Planning

With the robot state in $SE(2)$ we aim to represent the single-robot planning problem as a Markov Decision Process (MDP) which has an underlying Directed Acyclic Graph (DAG) structure. We use the AI-ToolBox library to represent and solve the MDP [36]. The MDP state s is represented as an integer vector:

$$s = \begin{bmatrix} x & y & \theta & t \end{bmatrix}$$

Each MDP action a is in the same discrete space, encoding the next state and incrementing the time by 1. This forces the MDP structure to be directed since states can never go back in time. The MDP is constructed with a transition matrix associating each (s, a, s') tuple with a transition probability and a reward matrix associating each (s, a) pair with a reward according to (4) with knowledge other robots' actions (to be introduced in Section V-C). We perform a breadth-first search over the state space by branching on feasible actions to populate the transition and reward matrices. As depicted

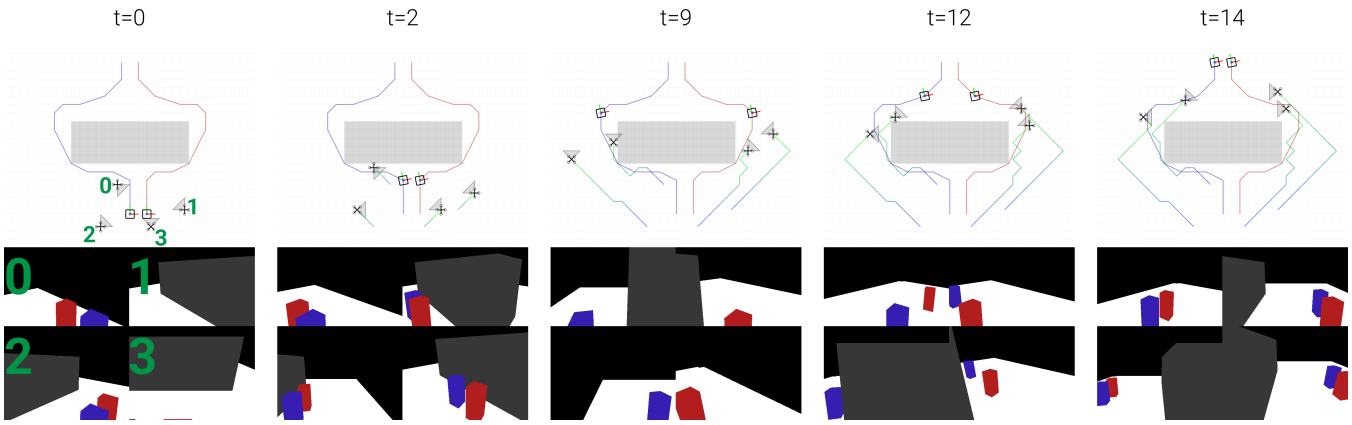

Fig. 4: Example robot views and joint trajectories from sequential plan in `Split` test case. Robot first-person views at each time step display viewing of the actors over the planning horizon.

in Fig. 2, the set of available actions is pruned based on environmental and inter-robot collisions. This directed MDP can be solved with one backward pass of value iteration to find the optimal greedy policy,[2] similar to the approach by Bucker et al. [7]. Following this policy from the initial state produces the optimal single-robot control sequence.

C. Sequential Planning

Now, we are able to generate the joint view plans for the multi-robot team. We do so by sequentially planning greedy single-robot trajectories in an arbitrary order as is common for methods based on submodular optimization [1–3, 6, 7, 30–32]. With some abuse of notation, each robot maximizes the objective for itself with access to prior decisions in the sequence:

$$U_i = \arg\max_{U} \mathcal{J}(X_{\mathrm{init},1:i}, U_{1:i-1} \cup U) \tag{6}$$

This forms a series of single-robot sub-problems that we solve with the value iteration planner (Sec. V-B). Through the course of this process, we accumulate pixel densities per each face to evaluate the view reward (2) and filter out states that would produce collisions with other robots (Sec. IV-B).

1) Suboptimality guarantees and inter-robot collisions: If we ignore inter-robot collisions, (5) has the form of a submodular maximization problem with a partition matroid constraint. Thanks to the famous result by Fisher et al. [29], sequential greedy planning via (6) is guaranteed to produce solutions to (5) with objective values no worse than 50% of optimal. However, inter-robot collisions violate the form of a partition matroid [6] so solutions that incorporate inter-robot non-collision constraints will not satisfy this guarantee. However, if the operating environment is not congested with robots, these non-collision constraints may not significantly influence the view rewards in practice. Our simulation results in Section VI-C support this conclusion that inter-robot constraints have negligible impacts on solution quality while *averting the serious consequences of collision*.

[2]Our current implementation converges in 5 passes without exploiting this structure.

D. Time Scaling Analysis

This section seeks to clarify the computational cost of Algorithm 1. After instantiating the MDP, value iteration for a single robot runs (ideally) with a single backwards pass over the reachable states. For planar motion the number of reachable states at step t is $O(t^2)$, and the total number of states over a horizon of T steps is $O(T^3)$. Inter-robot collision checking involves computation for each prior robot at every state, and for a single robot requires $O(T^3 N^{\mathrm{r}})$ time. The total time for the entire sequential planning process is then $O(T^3 N^{\mathrm{r}2})$. This addresses number of robots but not necessarily increasing problem scale in terms of the number of actors or environment complexity. Incorporating larger environments or more actors introduces more nuance. For example, evaluating the objective (4) is at least proportional to N^{a} but may be larger depending on the cost of rendering scenes with different numbers of actors. So, we can also say that the computation time scales as $\Omega(T^3 N^{\mathrm{r}2} N^{\mathrm{a}})$.

E. Considerations for application to real systems

Now, let us discuss how the proposed approach could be adapted for implementation on physical robots. First, our approach relies on known or predicted actor trajectories. This is reasonable for a scripted sequence such as when filming a movie—in this case, offline planning may also be reasonable. However, to account for uncertainty in actor or robot motions, systems should employ receding-horizon planning; though practical application would require substantial improvement in planning time. Then, for outdoor operation, GPS is often sufficient for localizing robots [11, 12, 23], particularly with RTK systems. Although prior works have implemented visual tracking for a single actor [37], additional instrumentation on the actors (such as additional GPS units) may be preferable for multi-actor settings. Then, while long-horizon prediction may be challenging, a Kalman filter can provide predictions over a short horizon [37] based on velocity and orientation. Regarding the map, if robots operate frequently in the same location (like a sports arena), a pre-existing map along with local collision avoidance may be

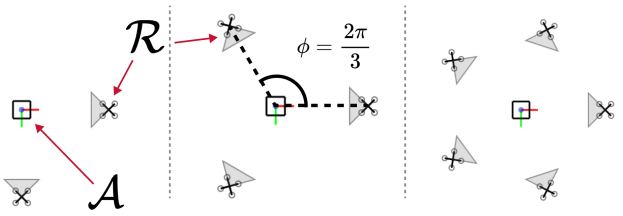

Fig. 5: Multi-robot formations for $N^{\mathrm{r}} = 2, 3, 5$

appropriate.

Algorithm 1: Sequential Greedy View Planning

Initialize $U_{\text{seq}} \leftarrow \{\}$
Initialize collisionMap $\leftarrow \{\}$
foreach i *in* $\mathcal{R}$ **do**
  $S_i \leftarrow$ DiscretizeStateSpace(envHeightMap)
  $A_i \leftarrow$ DiscretizeActionSpace(robotMaxMotion)
  MDP $\leftarrow$ BreadthFirstSearch($x_{i,0}$, S_i, A_i)
  // In BFS, R_{v} is computed at each explored state. Branching through availableActions removes actions that lead to collision states.
  $\pi_i \leftarrow$ ValueIteration(MDP)
  $\{u_{i,0}, \ldots, u_{i,T}\} \leftarrow$ ExtractTrajectory(π_i)
  Append $\{u_{i,0}, \ldots, u_{i,T}\}$ to U_{seq}
  $\xi_i \leftarrow$ applyActions($x_{i,0}$,$\{u_{i,0}, \ldots, u_{i,T}\}$)
  addCollisions(ξ_i)
end
return U_{seq}

VI. Experiments

We evaluate the performance of the sequential view planner in five test scenarios that aim to demonstrate view planning under a variety of conditions, and we compare to a formation planning baseline.

A. Formation Planning

We compare our sequential view planner against an assignment and formation based planner which we model off the multi-view formations applied in [11, 14] following analysis by Bishop et al. [38]. We assign equal numbers of robots to each actor, and groups assume formations as follows. The formation has a constant radius around an actor and a separation angle ϕ; for $N^{\mathrm{r}} > 2$ then $\phi = \frac{2\pi}{N^{\mathrm{r}}}$ and when $N^{\mathrm{r}} = 2$ then $\phi = \frac{\pi}{2}$ (see Fig. 5). We directly orient the formation about the actor to maximize the view reward R_{v} (including all actors) at each time-step. The formation planner has a fixed radius, ignores the motion model (Sec. IV-A), and does not consider environment and robot collisions; so view rewards for formation planning are generally optimistic and require no-extra computation.

Test Name	# Robot	# Actors	Timesteps	Env Collision
`Merge`	4	2	17	Yes
`Corridor`	2	2	17	Yes
`Forest`	2*	3	20	Yes
`Large`	18	6	10	No
`Split`	4	2	15	Yes

TABLE I: Test scenarios and parameters. The formation planner obtains an extra robot in the `Forest` scenario "for free" to match the number of actors

Test	Formation	Seq. w/o Inter-Robot		*Sequential*
`Split`	**1380**	1352 ± 34	$r_{\mathrm{c}} = 0.4$	1351 ± 35
`Large`	**1413**	1390 ± 27	$r_{\mathrm{c}} = 10.7$	1381 ± 27
`Merge`	1149	**1275 ± 26**	$r_{\mathrm{c}} = 0.2$	1274 ± 28
`Corridor`	1612	1808 ± 86	$r_{\mathrm{c}} = 0.8$	**1812 ± 85**
`Forest`	2114	**2534 ± 73**	$r_{\mathrm{c}} = 0.2$	2505 ± 116

TABLE II: Average and standard deviation of view reward (R_{v}) per robot for all test cases from 10 robot start configurations, comparing baselines to our approach (sequential). For sequential planning without inter-robot constraints, we also report the collision rate r_{c} (robots collided per trial).

B. Test Scenarios

Test scenarios are detailed in Fig. 6 and Table I. We use robots with camera intrinsic parameters of 2500px, 4000px, and 3000px (focal length, image width, image height). All drones are placed at 5 meters high with a camera tilt of 10 degrees from the horizon. For each test scenario, we also specify 10 unique robot starting configurations to introduce further variation. The scenarios are as follows:

`Split`: This test case is a simple group split and merge of 2 actors around an obstacle. A full view sequence from an example trajectory is displayed in Fig. 4.

`Large`: Focuses on scaling to larger teams and features 18 robots. Actors move through a series of short walls that produce occlusions but not collision constraints since they are below the navigation plane.

`Merge`: Contains actors moving around a corner in opposite directions. This test case investigates implicit actor assignment with actors being "handed off" at the corner.

`Corridor`: Tests robots moving through a narrow corridor. This focuses on the collision-aware aspect of the planner.

`Forest`: This is a dense occlusion/collision environment. For this scenario we limit sequential planning to two robots, fewer than the number of actors (three). This test aims to demonstrate the capacity to adapt to scenarios where assignments are not possible by evaluating if fewer robots can achieve similar or better coverage compared with the formation planner which requires 3 robots due to the minimum of 1 robot per formation. For the purpose of evaluation, we treat both planners as featuring only two robots when we report *per-robot* results.

C. Sequential Planner Performance

Fig. 7 and Table II summarize planner performance across each of the scenarios in terms of the view reward for formation planning and sequential planning both with and without inter-robot collision constraints. We observe that

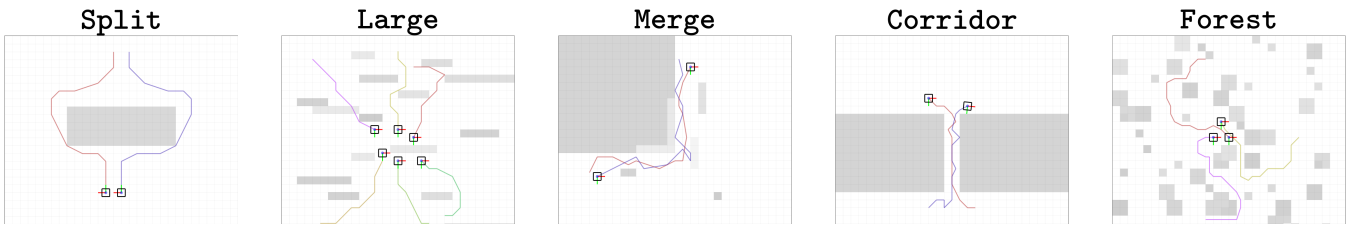

Fig. 6: Scenarios to evaluate specific aspects of multi-actor view planning. Actors are illustrated as boxes with uniquely colored trajectories. The darkness of elements in the height map indicates their occupied height. `Large` is the only test case with no collision obstacles as all elements are below the robot operating plane.

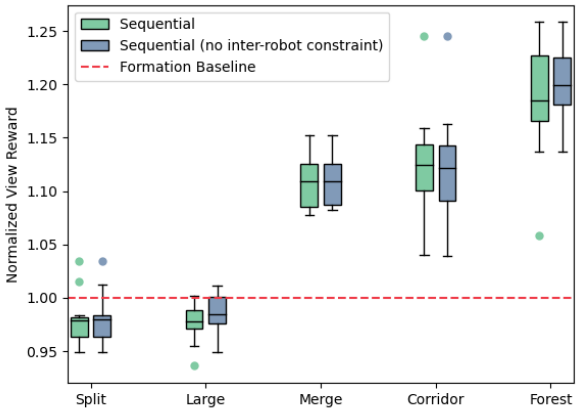

Fig. 7: Average view reward normalized by the baseline formation planner performance. 10 unique robot start configurations were specified for the sequential planners and are compared with the unique output from the formation planner. Both sequential planners outperform the formation planner baseline in the `Merge`, `Corridor`, and `Forest` scenarios, or else perform similarly. Sequential planners (with and without collision avoidance) also behave similarly in terms of view rewards: view rewards for the collision-aware planner are not impaired, though that planner no longer satisfies guarantees on solution quality.

sequential planning[3] outperforms formation planning in three of five scenarios—by an average of 13.9% in the `Merge`, `Corridor` and `Forest` scenarios. Notably, sequential planning also outperforms formation planning by 18% in the `Forest` scenario despite having one fewer robot. We also observe that inter-robot non-collision constraints do not significantly impair the performance of sequential planning. In the `Split` and `Large` test cases, all planners perform similarly. This may be because these scenarios provide more favorable conditions for the formation planner (and because the formation planner does respect starting positions or robot dynamics). `Split` provides ample space for formations of two robots to view the actors, and `Large` has shorter obstacles that produce occlusions but not collisions.

The sequential planner that ignores inter-robot collisions guarantees solutions within half of optimal but may allow robots to collide. Since both versions of the sequential planner obtain similar solution quality (Fig. 7), we conclude inter-robot collision constraints do not significantly impair performance in this setting. We also report collision rates r_c in terms of *robots collided per trial* for the sequential planner that ignores collisions Table II. In most scenarios, we observe less than one collision per trial except for `Large` where ten robots (more than half) collide on average, but we still do not see a significant difference in objective values. The `Corridor` scenario is also highly-constrained, and we see similar objective values despite increased collisions.

Figure 4 displays the joint view plans and internal view planner renderings for the `Split` test case. This figure illustrates the capacity of sequential planning to optimize views of one or more actors and to implicitly reconfigure or "hand-off" assignments over the course of a trial. The robots all view both actors at $t = 2$; the pairs split off and transition to each viewing a single actor by $t = 9$; and they go back to jointly viewing actors by $t = 14$. This behavior is not manually specified and arises only from optimizing trajectories and views.

D. Scaling number of robots

Since R_v is a square-root sum of pixel coverage over the actors, we would expect scaling the number of robots to follow a similar trend of diminishing returns. In Fig. 8, we observe this trend with the `Large` test case for 1–18 robots. These diminishing returns would correspond—likely simultaneously—to increasing coverage over viewing angles of the actors' faces and gradual (and ideally uniform) increase in pixel densities. Furthermore, scaling the number of robots produces nearly linear growth in computation time even though our approach requires quadratic time asymptotically (Sec. V-D). This is likely because the cost of inter-robot collision checks is inconsequential compared to the cost of solving the single-robot MDPs. Additionally, growth in planning time slows substantially following the first robot—because robots share the same camera models we benefit significantly from caching evaluation of camera views.

VII. Conclusion and Future Work

In this work, we presented a novel system for multi-robot view planning based on sequential greedy planning with an occlusion-aware objective. Through evaluation in five different scenarios, we observe sequential planning outperforming formation-based planning and specifically excelling in obstacle-dense environments. Additionally, we observe similar perception performance for sequential planning with and

[3]Referring to the collision-free version, but both obtain similar performance.

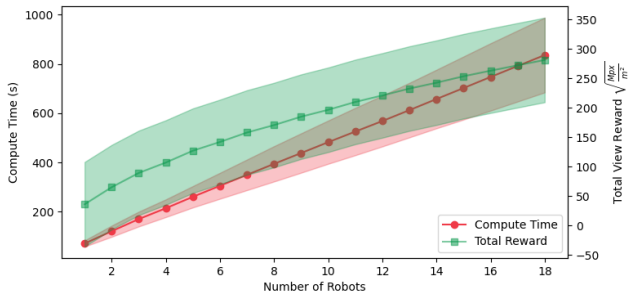

Fig. 8: Computation time and reward accumulated with scaling the number of robots in the `Large` test case. The means and standard deviations are computed across 10 unique start locations.

without inter-robot collision constraints. This demonstrates that sequential planning is able to find good solutions when accounting for possible collisions between robots (despite no longer satisfying requirements for bounded suboptimality). In future work, we aim to extend this view planner to a 3D human pose reconstruction task, and optimize our implementation to run at real-time rates.

Acknowledgment

This work was supported by the National Science Foundation (NSF) under Grant No. 2024173. Krishna was sponsored by the NSF REU program, as a part of the Robotics Institute Summer Scholars (RISS) program. Micah contributed to this work primarily while a postdoc at CMU.

References

[1] M. Corah and N. Michael, "Scalable distributed planning for multi-robot, multi-target tracking," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Prague, Czech Republic, Sep. 2021.

[2] X. Cai, B. Schlotfeldt, K. Khosoussi, N. Atanasov, G. J. Pappas, and J. P. How, "Energy-aware, collision-free information gathering for heterogeneous robot teams," *IEEE Trans. Robotics*, vol. 39, pp. 2585–2602, 2023.

[3] B. Schlotfeldt, V. Tzoumas, and G. J. Pappas, "Resilient active information acquisition with teams of robots," *IEEE Trans. Robotics*, vol. 38, no. 1, pp. 244–261, 2021.

[4] H.-A. Hung, H.-H. Hsu, and T.-H. Cheng, "Image-based multi-UAV tracking system in a cluttered environment," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 4, pp. 1863–1874, 2022.

[5] Y. Zhao, X. Wang, C. Wang, Y. Cong, and L. Shen, "Systemic design of distributed multi-UAV cooperative decision-making for multi-target tracking," *Autonomous Agents and Multi-Agent Systems*, vol. 33, pp. 132–158, 2019.

[6] M. Corah and N. Michael, "Distributed matroid-constrained submodular maximization for multi-robot exploration: theory and practice," *Auton. Robots*, vol. 43, no. 2, pp. 485–501, 2019.

[7] A. Bucker, R. Bonatti, and S. Scherer, "Do you see what I see? Coordinating multiple aerial cameras for robot cinematography," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Xi'an, China, May 2021.

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

[9] P. Pueyo, J. Dendarieta, E. Montijano, A. C. Murillo, and M. Schwager, "CineMPC: A fully autonomous drone cinematography system incorporating zoom, focus, pose, and scene composition," *IEEE Trans. Robotics*, vol. 40, pp. 1740–1757, 2024.

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

[11] C. Ho, A. Jong, H. Freeman, R. Rao, R. Bonatti, and S. Scherer, "3D human reconstruction in the wild with collaborative aerial cameras," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Prague, Czech Republic, Sep. 2021.

[12] N. Saini, E. Price, R. Tallamraju, R. Enficiaud, R. Ludwig, I. Martinovic, A. Ahmad, and M. J. Black, "Markerless outdoor human motion capture using multiple autonomous micro aerial vehicles," in *Proc. of the IEEE/CVF Intl. Conf. on Comp. Vis.*, Seoul, South Korea, 2019.

[13] C. Cao, H. Zhu, F. Yang, Y. Xia, H. Choset, J. Oh, and J. Zhang, "Autonomous exploration development environment and the planning algorithms," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Philadelphia, PA, May 2022.

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

[15] Q. Jiang and V. Isler, "Onboard view planning of a flying camera for high fidelity 3D reconstruction of a moving actor," Jul. 2023. [Online]. Available: http://arxiv.org/abs/2308.00134

[16] S. Hughes, R. Martin, M. Corah, and S. Scherer, "Multi-robot planning for filming groups of moving actors leveraging submodularity and pixel density," in *Proc. of the IEEE Conf. on Decision and Control*, Milan, Italy, Dec. 2024, to appear.

[17] "Drone That Follows You - Skydio 2+ | Skydio." [Online]. Available: https://www.skydio.com/skydio-2-plus/

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

[19] I. Mademlis, A. Torres-González, J. Capitán, M. Montagnuolo, A. Messina, F. Negro, C. Le Barz, T. Gonçalves, R. Cunha, B. Guerreiro *et al.*, "A multiple-UAV architecture for autonomous media production," *Multimedia Tools and Applications*, vol. 82, no. 2, pp. 1905–1934, 2023.

[20] L.-E. Caraballo, Á. Montes-Romero, J.-M. Díaz-Báñez, J. Capitán, A. Torres-González, and A. Ollero, "Autonomous planning for multiple aerial cinematographers," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Las Vegas, Nevada, Sep. 2020.

[21] A. Ray, A. Pierson, H. Zhu, J. Alonso-Mora, and D. Rus, "Multi-robot task assignment for aerial tracking with viewpoint constraints," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Prague, Czech Republic, May 2021.

[22] P. Pueyo, E. Montijano, A. C. Murillo, and M. Schwager, "CineTransfer: Controlling a robot to imitate cinematographic style from a single example," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Detroit, Michigan, Oct. 2023.

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

[24] X. Xu, G. Shi, P. Tokekar, and Y. Diaz-Mercado, "Interactive multi-robot aerial cinematography through hemispherical manifold coverage," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Kyoto, Japan, Oct. 2022.

[25] R. Tallamraju, N. Saini, E. Bonetto, M. Pabst, Y. Liu, M. Black, and A. Ahmad, "AirCapRL: Autonomous aerial human motion capture using deep reinforcement learning," *IEEE Robot. Autom. Letters*, vol. 5, no. 4, pp. 6678–6685, 2020.

[26] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3D object reconstruction," *Auton. Robots*, vol. 42, no. 2, pp. 197–208, 2018.

[27] M. Roberts, S. Shah, D. Dey, A. Truong, S. Sinha, A. Kapoor, P. Hanrahan, and N. Joshi, "Submodular trajectory optimization for aerial 3D scanning," in *Proc. of the IEEE/CVF Intl. Conf. on Comp. Vis.*, Venice, Italy, Oct. 2017, pp. 5334–5343.

[28] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.

[29] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions-II," *Polyhedral Combinatorics*, vol. 8, pp. 73–87, 1978.

[30] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *J. Artif. Intell. Res.*, vol. 34, pp. 707–755, 2009.

[31] M. Lauri, J. Pajarinen, J. Peters, and S. Frintrop, "Multi-sensor next-best-view planning as matroid-constrained submodular maximization," *IEEE Robot. Autom. Letters*, vol. 5, no. 4, pp. 5323–5330, 2020.

[32] S. McCammon, G. Marcon dos Santos, M. Frantz, T. P. Welch, G. Best, R. K. Shearman, J. D. Nash, J. A. Barth, J. A. Adams, and G. A. Hollinger, "Ocean front detection and tracking using a team of heterogeneous marine vehicles," *J. Field Robot.*, vol. 38, no. 6, pp. 854–881, 2021.

[33] M. Corah, "On performance impacts of coordination via submodular maximization for multi-robot perception planning and the dynamics of target coverage and cinematography," in *RSS Workshop on Envisioning an Infrastructure for Multi-robot and Collaborative Autonomy Testing and Evaluation*, 2022.

[34] M. Corah and N. Michael, "Volumetric objectives for multi-robot exploration of three-dimensional environments," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Xi'an, China, May 2021.

[35] A. Schrijver, *Combinatorial optimization: polyhedra and efficiency*. Springer Science & Business Media, 2003, vol. 24.

[36] E. Bargiacchi, D. M. Roijers, and A. Nowé, "AI-Toolbox: A C++ library for reinforcement learning and planning (with Python bindings)," *Journal of Machine Learning Research*, vol. 21, no. 102, pp. 1–12, 2020.

[37] R. Bonatti, C. Ho, W. Wang, S. Choudhury, and S. Scherer, "Towards a robust aerial cinematography platform: Localizing and tracking moving targets in unstructured environments," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Macau, China, Nov. 2019.

[38] A. N. Bishop, B. Fidan, B. D. Anderson, K. Doğançay, and P. N. Pathirana, "Optimality analysis of sensor-target localization geometries," *Automatica*, vol. 46, no. 3, pp. 479–492, 2010.