Hypergame-based Adaptive Behavior Path Planning for Combined Exploration and Visual Search

Mihir Dharmadhikari[1], Harshal Deshpande[2], Tung Dang[1], and Kostas Alexis[1,3]

***Abstract*— In this work, we present an adaptive behavior path planning method for autonomous exploration and visual search of unknown environments. As volumetric exploration and visual coverage of unknown environments, with possibly different sensors, are non-identical objectives, a principled combination of the two is proposed. In particular, the method involves three distinct planning policies, namely exploration, and sparse or dense visual coverage. A hypergame formulation is proposed which allows the robot to select for the next-best planning behavior in response to the currently encountered environment challenges in terms of geometry and visual conditions, alongside a self-assessment of its performance. The proposed planner is evaluated in a collection of experimental and simulation studies in diverse environments, while comparative results against a state-of-the-art exploration method are also presented.**

I. INTRODUCTION

Robotic systems are being deployed in an ever-widening set of applications. Among others, autonomous robots, be it flying, ground, or underwater, are tasked to explore, map, search and characterize diverse environments of increasing size and complexity. In response to the needs of such applications, the research community has contributed a set of methods for exploration path planning, visual search, and broadly, information sampling [1–16]. Despite the progress, a limiting factor of the majority of such works relates to the fact that they are optimizing a single objective - for example relating to volumetric exploration [10] or mutual information maximization [14]. However, the fact that possible diverse challenges relating to a) the size of the environment, b) its geometric complexity, c) degradation of sensor data quality (e.g., due to darkness), and d) the lack of prior information with respect to where entities of interest may lie, can render such monolithic strategies inefficient. Although the planning method may continue to locally maximize its objective, the decisions taken may not best reflect the specific challenges encountered. Essentially, there is a lack of adaptive behavior-based path planning that would modify the information sampling policy in response to challenges of topology, size, visual conditions, and how the robot may self-evaluate its performance in its task as it explores a new environment.

In response to the above, this work contributes a hypergame-based adaptive behavior path planner that exploits a library of possible planning behaviors to best adjust how the robot explores and searches its environment with no prior knowledge. The method proposes the modeling of the diverse - and at any time - unknown future challenges the environment may pose to the robot (e.g., different topologies requiring different planning behaviors or darker zones requiring close-up search) through the concept of hypergames [17]. In this context, the environment and the robot participate in a game in which the robot is equipped with a set of possible path planning strategies yet it is not aware which of them best suits the future challenges the environment it explores will present to it. To best explore volumetrically, and visually search environments of arbitrary size, the robot makes local decisions for when to engage three types of policies, namely a) volumetric exploration, b) visual coverage for good visual conditions, and c) dense visual coverage for dark settings. It selects the best policy given an imperfect understanding of the environment challenges, and relies on a) an online uncertain topological representation encoding zones of distinct geometries, b) evaluation of the visual conditions, and c) self-assessment of the performance of its recently engaged planning modes. The result of this self-assessment-aware evaluation is probabilistic in nature and returns the next-best-behavior and the associated path for the robot to execute.

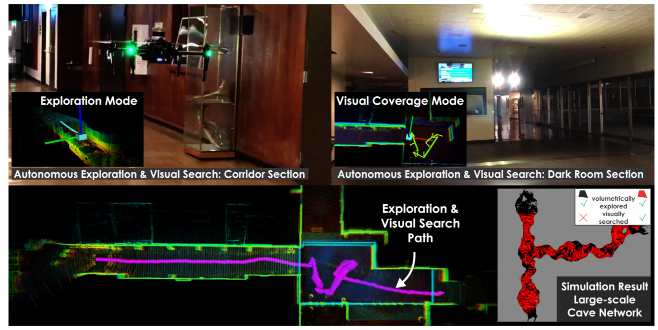

Fig. 1. Instance of an autonomous exploration and visual search mission.

To evaluate this new path planning strategy, we present experimental studies using an aerial robot equipped with a 3D LiDAR and a visual camera, performing an autonomous exploration and search mission, alongside a set of simulation studies inside complex buildings and cave networks. To demonstrate the benefits of this hypergame-based adaptive behavior planning, we further present comparisons against a state-of-the-art exploration planner.

The remainder of this paper is organized as follows. Section II outlines related work, followed by the problem statement in Section III. The proposed approach is detailed in Section IV. Evaluation studies are detailed in Section V, followed by conclusions drawn in Section VI.

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111820045. The presented content and ideas are solely those of the authors.

This work was conducted at the University of Nevada, Reno, NV, USA

[1]University of Nevada, Reno, 1664 N. Virginia, 89557, Reno, NV, USA mdharmadhikari@nevada.unr.edu

[2]BITS-Pilani Goa Campus, India

[3]NTNU, O. S. Bragstads Plass 2D, 7034, Trondheim, Norway

II. RELATED WORK

A rich body of work has focused on the problems of exploration, mapping and visual search [1–16, 18, 19]. Early work includes the sampling of "next–best–views" [8], and frontiers–based exploration [9]. More recent efforts have proposed receding horizon multi-objective planning [5, 6, 20, 21], graph-based and motion primitives-based methods [10, 11], visual search [15, 16], information-theoretic schemes [14], and multi-robot strategies [22, 23]. Existing planning methods have performed well in simple missions yet cannot provide resilient performance in environments of very large scale, diverse and challenging geometries, and with visually-degraded conditions. Although general planning solutions can - in principle - be modeled with the use of Partially Observable Markov Decision Process (POMDP) formulations, it is well known that the associated large state/action spaces, data complexity in the acquired observations, and the complexity of the transition, observation and reward functions render this problem extremely hard and often intractable in its full form. Motivated by the above and considering the limited endurance of Micro Aerial Vehicles, this work contributes a new hypergame-based path planner to explore and visually search unknown environments by adapting to the specific environmental challenges encountered as the robot proceeds in its mission. It delivers a strategy to sample information that transitions from pure volumetric exploration to sparse or dense visual coverage depending on the complexities the environment presents and the robot's self-assessment for its performance.

III. PROBLEM FORMULATION

The overall problem considered in this work is that of sampling information over a bounded 3D volume $V \subset \mathbb{R}^3$ given a robot with two frustum-constrained sensors, namely a depth sensor $\mathcal{S}^D$ and a visual search sensor $\mathcal{S}^V$ with the latter having dynamic range in response to possible degradation in its observations. This information sampling problem may be cast globally as that of starting from an initial collision-free configuration and deriving viewpoints that lead to determine which parts of the initially unexplored volume $V_{une} \overset{init.}{=} V$ are free $V_{free} \subseteq V$ or occupied $V_{occ} \subseteq V$, alongside covering the associated viewable surfaces of V_{occ}, S_{occ}, with each of the $\mathcal{S}^V$ observations remaining informative. A visual camera observation is informative if it facilitates inference for scene understanding and in this work relates to the effective range given bright or dark conditions. From an environment representation standpoint, the volume is discretized in an occupancy map $\mathbb{M}$ consisting of cubical voxels $m \in \mathbb{M}$ with edge length r_V, paired with an associated surface model that is being reconstructed $\mathbb{M}_S$. The operation is subject to the dynamic constraints of the vehicle. As for most sensors the perception stops at surfaces, hollow spaces or narrow pockets cannot be fully explored leading to residual volume and surface we have the following definition and problem definition cast globally.

Definition 1 (Residual Volume and Surface) Let Ξ be the simply connected set of collision free configurations and $\bar{\mathcal{V}}_m \subseteq \Xi$ the set of all configurations from which the voxel m can be perceived by both $\mathcal{S}^D$ and $\mathcal{S}^V$. Then the residual volume is given as $V_{res} = \bigcup_{m \in \mathbb{M}} (m | \ \bar{\mathcal{V}}_m = \emptyset)$. A residual surface is also denoted as S_{res} considering $\mathcal{S}^V$.

Problem 1 (Volumetric Exploration and Visual Search Problem) Given a bounded volume V, find a collision free path σ starting at an initial configuration $\xi_{init} \in \Xi$ that a) leads to identifying the free and occupied parts V_{free} and V_{occ} based on $\mathcal{S}^D$ and b) performs complete visual search (coverage) with $\mathcal{S}^V$ when being executed, such that there does not exist any collision free configuration from which any piece of $V \backslash \{V_{free}, V_{occ}\}$, or any subset of the associated surface S_{occ} related to the outer observable surfaces of V_{occ}, could be perceived. Thus, $V_{free} \cup V_{occ} = V \setminus V_{res}$ and $S_{free} \cup S_{occ} = V \setminus S_{res}$.

IV. PROPOSED APPROACH

This section details the proposed Hypergame-based Adaptive Behavior path Planner (HABPlanner) for combined exploration and visual search. The description focuses on robots with true 3D navigation capabilities (e.g., aerial systems) although the method is generalizable to different configurations. The proposed approach is motivated by the need to decide in real-time the best information sampling behavior, and associated paths, the robot has to employ to achieve both exploration of the initially unknown environment and visual coverage over its surface. The first core behavior is driven by a depth sensor $\mathcal{S}^D$, while the latter by a visual camera $\mathcal{S}^V$. As the environment is initially unknown - including its complexity and visual conditions - and the robot is subject to endurance limitations, the proposed method is focused on delivering efficient mapping and search behaviors simultaneously as the system iteratively executes its mission. Given that any decision at an instance of the mission is only informed for the map explored and searched up to this moment, the decisions of the robot are subject to uncertainty with respect to the challenges it will face subsequently.

A. Hypergame Modeling

A hypergame occurs when one of the players does not know - or fully understand - all the game strategies. Hypergame theory extends game theory concepts by allowing a player to outdo an opponent and obtain a preferable outcome. The ability to outdo an opponent occurs in the hypergame because the different views (perception or deception) of opponents are captured in the model by incorporating information unknown to other players (misperception or intentional deception). In the context of this work, the robot and the environment are thought of as two "players" in a game in which the robot tries to best explore and visually search the environment but the latter presents a set of possible challenges (relating to geometric complexity, size or visual conditions) to the robot that are not known beforehand and for which the robot can only perform uncertain inference. To best perform in this game, HABPlanner equips the robot with two sets of planning behaviors, namely for a) exploration and

b) sparse or dense visual search (Figure 2). The decision for the next-best-behavior to employ at any given time is based on the robot's imperfect perception of the environment encoded through a utility function that involves a) the geometric classification of a local subset of the map identifying if it is room- or corridor-like, b) the percentage of the explored map seen by $\mathcal{S}^V$, c) detecting if the local environment is dark, and d) a self-assessment for the recent performance of the planner behaviors. As the inference of the robot for the environment is subject to uncertainty, hypergames correspond to the appropriate modeling language.

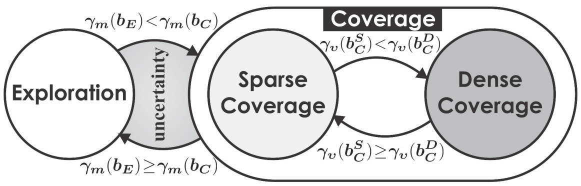

Fig. 2. Illustration of the behavior modes of the proposed hypergame-based planner for exploration and visual search. First, the best planning mode, exploration (b_E) or visual coverage (b_C), is decided based on the scores γ_m. When coverage is selected, the scores γ_v are used to decide for sparse (b_C^S) or dense (b_C^D) search. As the decisions depend on uncertain inference for the map topology encountered and self-assessment of the robot's performance, a hypergame formulation is most suitable for modeling the adaptive planning.

B. Mapping and Topology Identification

The planner relies on a dual representation of the environment, namely a volumetric map and a lightweight topological graph. The volumetric map facilitates autonomous collision-free navigation, volumetric calculations, and evaluations of areas covered. Specifically, we use Voxblox [24], a volumetric mapping pipeline based on Truncated Signed Distance Fields (TSDFs). Voxblox uses voxel hashing resulting in a constant time lookup in the volumetric map. The map is generated using the pointcloud data obtained from the onboard depth sensor $\mathcal{S}^D$. The mesh representation of the occupied part of the map generated by voxblox is used as the reconstructed surface model. In this map, we further annotate the occupied voxels within the visual camera $\mathcal{S}^V$ frustum as voxels seen by $\mathcal{S}^V$. The camera range r used to annotate the map is dynamically adjusted based on the light conditions as:

$$r = r_{\min} + (1 - i_d)(r_{\max} - r_{\min}) \quad (1)$$

where i_d is the fraction of camera image pixels having intensity below a threshold value λ_{th}.

Alongside this volumetric map, a higher-level representation of the environment topology is needed to select the optimal planning behavior. Hence a topological map is also maintained and represented as a lightweight graph the vertices of which correspond to points in space and store the beliefs of the robot about the geometry of the local environment. In this work, we consider two topologies, namely tunnel/corridor-like sections and room-like areas.

To identify which of the modeled topologies best describes a local subset of the environment geometry, the instantaneous pointcloud data from $\mathcal{S}^D$ is utilized in a two-step detection. First, we evaluate the distribution of the pointcloud around the robot. The pointcloud is divided into eight parts along the eight octants in the x-y plane centered around the robot's current location. A value v is assigned to every octant and is equal to the eigenvalue along the direction of that octant obtained through Principal Component Analysis (PCA) on that part of the pointcloud, normalized by the mean value of all octants. In an ideal room-like topology, all the octants will have a similar distribution of pointcloud resulting in the values being closer to 1.0 whereas an unequal distribution will be observed in a tunnel. Let n_1 be the number of octants with $v \in [v_1, v_2], v_1 < 1.0 < v_2$. Then, the belief of the current topology being a room is proportional to n_1 and that of the tunnel is proportional to $8 - n_1$. Second, the central three rows of the depth image of the sensor scan are used to identify sections of points having sudden change in depth called frontiers. A constant weight w_f is added to the tunnel belief if exactly two frontiers are detected. Finally, the beliefs are normalised to range in $[0, 1]$. As the sensor measurements can be noisy, we apply a low pass filter over the beliefs calculated from the last m readings to get the final beliefs for each topology.

C. Behavior Selection Methodology

The proposed planner operates iteratively and at each step optimizes its behavior to perform efficient exploration and visual search within a local window around the current robot location. In each iteration, the method first utilizes the hypergame formulation to decide which main planning mode, Exploration (b_E) or Visual Coverage (b_C) should be employed. A score is calculated for each of these modes and depends on a) the planner's uncertain belief about the environment topology encountered (corridor- or room-like), b) the percentage of the mapped surface seen by the camera $\mathcal{S}^V$, and c) a self-assessment about the performance of each of the planning modes in the previous iterations. The score γ_m for each mode $b_j, j \rightarrow E, C$ is calculated as follows:

$$\begin{aligned} \gamma_m(b_E) &= (\%S_{seen})(1 + P(e = e_c))(1 + \mathcal{P}(b_E, n)) \\ \gamma_m(b_C) &= (1 - \%S_{seen})(1 + P(e = e_r))(1 + \mathcal{P}(b_C, n)) \end{aligned} \quad (2)$$

where $\%S_{seen}$ is the percentage of mapped surface seen by the camera within the local window around the robot location. $P(e = e_c)$ and $P(e = e_r)$ are the probabilities of the current environment being a corridor (e_c) or a room (e_r) respectively. The term $\mathcal{P}(b_j, n)$ accounts for the cumulative performance of the planning mode $b_j, j \rightarrow E, C$ at steps when it was selected in the last n planning iterations. The performance in an iteration is calculated using the change in the percentage of volume mapped, $\Delta\%V_{seen}$, by $\mathcal{S}^D$ and the change in the percentage of surface covered, $\Delta\%S_{seen}$, by $\mathcal{S}^V$. Volumetric exploration and visual search being the main objectives of the exploration and coverage modes respectively, the other objective is discounted in the performance term. For the iteration i the performance ρ^i is formulated as:

$$\rho^i(b_E) = \Delta\%V_{seen} + \eta\Delta\%S_{seen} \quad (3)$$

$$\rho^i(b_C) = \Delta\%S_{seen} + \eta\Delta\%V_{seen} \quad (4)$$

where $\eta > 0$ is the discount factor. Combining the performance of the last n planning iterations we get the cumulative performance term $\mathcal{P}(b_j, n)$ for behavior $b_j, j \to E, C$ in the iteration k as defined in Eq. (6). Let:

$$\rho^i_{rel}(b_j) = \begin{cases} \rho^i(b_j) - \rho_{th} & , \quad \text{if behavior } b^i = b_j \\ 0 & , \quad \text{otherwise} \end{cases} \tag{5}$$

where b^i is the behavior selected in iteration i and ρ_{th} is a tunable threshold below which the performance is considered to be non satisfactory. Then, we derive $\mathcal{P}(b_j, n)$ as:

$$\mathcal{P}(b_j, n) = \sum_{i=k-n}^{k-1} \left(\frac{i+n-k+1}{n}\right)\rho^i_{rel}(b_j) \tag{6}$$

Finally, using the calculated scores, the behaviour having the highest score is selected as the next-best-planning mode. If the selected mode is Exploration ($\gamma_m(b_E) \geq \gamma_m(b_C)$) then the path is computed as described in Section IV-D. If Coverage mode is selected ($\gamma_m(b_C) > \gamma_m(b_E)$), we further evaluate the optimal sub-behavior, namely Sparse (b^S_C) or Dense (b^D_C). The decision happens based on a) the environment light conditions perceived by the camera, and b) the performance of the sub-behaviors in the previous planning iterations. Another score is calculated for each sub-behavior and the highest scoring sub-behavior is selected. For sub-behavior $b^\ell_C,\ \ell \to S, D$ the score γ_v is calculated as:

$$\gamma_v(b^\ell_C) = w_i U_i(b^\ell_C) + w_p U_p(b^\ell_C) + k_p(b^\ell_C),\ \ell \to S, D \tag{7}$$

where $k_p > 0$ is the preference weight for the sub-behavior b^ℓ_C giving a default preference to a particular sub-behavior. The functions $U_i(b^\ell_C)$ and $U_p(b^\ell_C)$, with tunable parameters $w_i, w_p > 0$, are the expected scores of the robot for executing b^ℓ_C given the current light conditions and the performance of that sub-behavior in the previous iterations respectively. The term $U_p(b^\ell_C)$ is equal to the performance $\mathcal{P}(b^\ell_C, n)$ as defined in Eq. (6), whereas we define $U_i(b^\ell_C)$ as:

$$U_i(b^\ell_C) = \begin{cases} 1 - i_d & , \quad \text{if } b^\ell_C = b^S_C \\ i_d & , \quad \text{if } b^\ell_C = b^D_C \end{cases} \tag{8}$$

where i_d is the fraction of camera image pixels having intensity below a threshold value λ_{th}.

The sub-behavior having the highest score is selected and the path to follow is calculated as described in Section IV-E. The robot's beliefs about the environmental challenges are subject to uncertainty. Formulating a hypergame allows the planner to incorporate this uncertainty in the decision process and through performance monitoring, it is able to overcome the incorrect decisions caused by inaccurate environment inference.

D. Exploration Planner

The exploration mode of HABPlanner is based on our previous work on graph-based exploration planning (GBPlanner) [10]. First, an undirected graph $\mathbb{G}_L$ is incrementally built inside a local map space $\mathbb{M}_L$ around the current robot configuration ξ_0 involving the robot position and heading ($\xi = [x, y, z, \psi]^T$). Given the updated occupancy map $\mathbb{M}$, a bounded "local" volume V_{D_L} with dimensions D_L centered around ξ_0 and the respective map subset $\mathbb{M}_L$, and a bounding box D_R representing the robot physical size (alongside the respective map subset $\mathbb{M}_R(\xi_*)$ for a robot state ξ_*), the planner samples a set of random configurations ξ_{rand} inside V_{D_L} and after checking which local connections (within a defined radius δ) are collision-free, builds the graph $\mathbb{G}_L$ and its vertex and edge sets $\mathbb{V}, \mathbb{E}$ respectively. Subsequently, using the Dijkstra's algorithm [25] it derives paths Σ_L in the graph starting from the root vertex at ξ_0 (root to all destinations). Then the best path in terms of maximizing an information gain relating to the new volume explored, called $\mathbf{VolumeGain}$, is identified and provided to the robot for execution. More specifically, given a path $\sigma_i \in \Sigma_L, i = 1...\mu$ with a set of vertices $\nu^i_j \in \sigma_i, j = 1...m_i$ along the path (corresponding to a set of robot ξ configurations), the exploration gain $\mathbf{\Gamma}_E(\sigma_i)$ is calculated as follows:

$$\mathbf{\Gamma}_E(\sigma_i) = e^{-\zeta \mathcal{Z}(\sigma_i, \sigma_e)} \sum_{j=1}^{m_i} \mathbf{VolumeGain}(\nu^i_j) e^{-\delta \mathcal{D}(\nu^i_1, \nu^i_j)} \tag{9}$$

where $\zeta, \delta > 0$ are tunable factors, $\mathcal{D}(\nu^i_1, \nu^i_j)$ is the cumulative Euclidean distance from vertex ν^i_j to the root ν^i_1 along the path σ_i, and $\mathcal{Z}(\sigma_i, \sigma_e)$ is a similarity distance metric between the planned path as compared to a pseudo straight path σ_e with the same length along the currently estimated exploration direction. Details are provided in [10].

E. Coverage Planner

HABPlanner utilizes an efficient coverage planner enabled in the visual search mode in order to maximise the mapped surface seen by the camera sensor $\mathcal{S}^V$. The planner builds an undirected graph $\mathbb{G}_L$ and its vertex and edge sets $\mathbb{V}, \mathbb{E}$ respectively as described in Section IV-D. Subsequently for each vertex $\nu \in \mathbb{V}$, an information gain, called $\mathbf{VisualGain}$, is evaluated and corresponds to the number of unseen occupied voxels perceived by the onboard visual camera sensor $\mathcal{S}^V$ from vertex ν. Then, we sample a set $\mathbb{C}$ containing p sets of q vertices each from $\mathbb{G}_L$ as candidate vertices for our final path. The cumulative visual search gain $\mathbf{\Gamma}_C$, for all the vertices in set $c_i \in \mathbb{C}$ is calculated as shown in Eq. (10). It is noted that in order to avoid counting a voxel perceived by multiple viewpoints more than once, we store the hash keys generated by Voxblox for each voxel and remove the duplicate keys while counting the voxels for $\mathbf{\Gamma}_C$. The gain $\mathbf{\Gamma}_C$ for the set $c_i \in \mathbb{C}, i = 1...p$ having vertices $\nu^i_j \in c_i, j = 1...q$ is calculated as follows:

$$\mathbf{\Gamma}_C(c_i) = \sum_{j=1}^{q} \mathbf{VisualGain}(\nu^i_j) \tag{10}$$

Finally, the set c^*_i having the highest $\mathbf{\Gamma}_C$ is selected and the order in which the vertices are to be visited is found by solving the Traveling Salesman Problem (TSP) [26] on the set c^*_i. The consecutive vertices in the tour are then connected by the shortest path in the graph $\mathbb{G}_L$ between the two vertices using Dijkstra's algorithm [25]. This final path is then provided to the robot controller for execution.

We further define the two sub-behaviors of the Coverage Planner, namely Sparse and Dense Coverage. Both sub-behaviors work on the same core principle as described above but the Sparse Coverage is tailored towards faster search

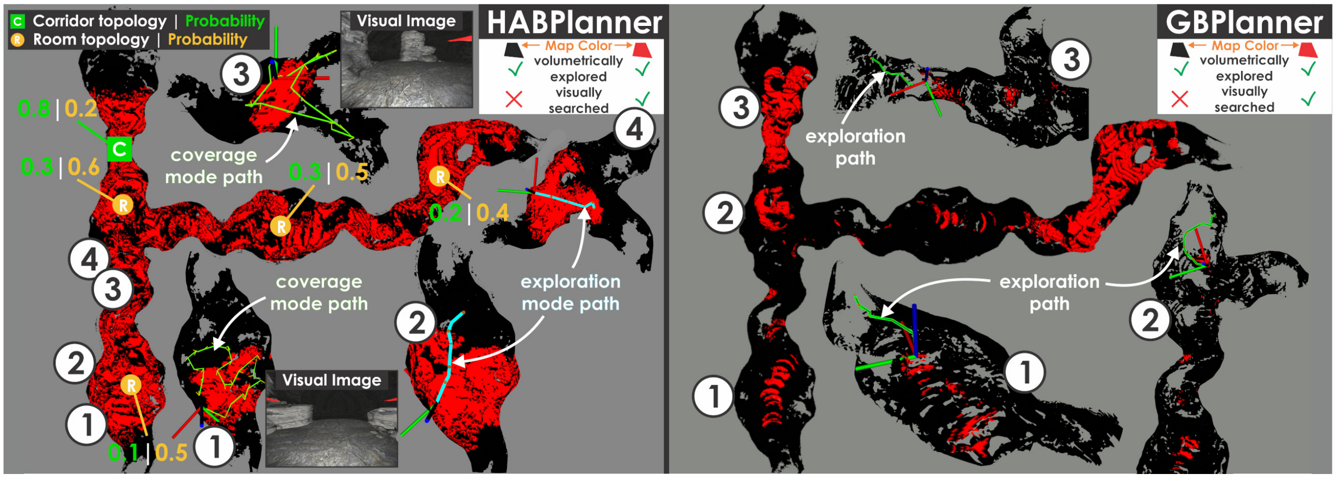

Fig. 3. Performance comparison of the proposed HABPlanner versus GBPlanner in a cave network. Both planners are given the same amount of mission time. Areas in the map that are red are both explored and seen with the camera sensor, whereas black regions are only explored volumetrically but not seen by the camera sensor. The planner's beliefs about the environment topology at certain locations are also shown in the left figure. As demonstrated HABPlanner by far outperforms GBPlanner in terms of visual search coverage ratio. The dark yet spacious and large scale cave network necessitates the use of discrete behaviors optimised for specific settings. It is noted that the effective range of the camera changes in relation to the light conditions.

and for well-lit environments, whereas Dense Coverage is tuned for dark settings. The two sub-behaviors differ in a) the number of vertices q in the candidate sets with Dense Coverage having more vertices, and b) the camera frustum range used for calculating the $\mathbf{VisualGain}$ with Dense Coverage using a smaller range. Furthermore, to speed-up Sparse coverage, we remove vertices having $\mathbf{VisualGain}$ smaller than a threshold w_{th} from the best set.

V. EVALUATION STUDIES

A set of experimental and simulation studies were conducted to evaluate the ability of HABPlanner to achieve simultaneous exploration and visual search. The simulation studies involve a cave network and an urban structure. The experiment took place inside a university building.

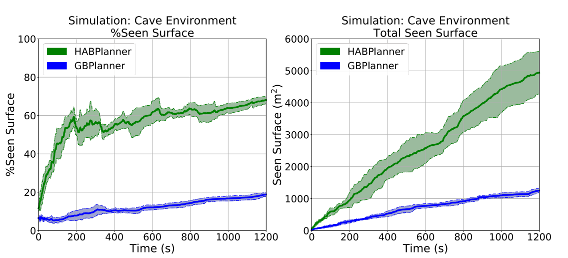

Fig. 4. Percentage of the explored surface seen and total surface seen using HABPlanner and GBPlanner inside the cave. The large scale of the cave environment demonstrates the benefits of the co-optimization of exploration and visual search from HABPlanner. GBPlanner naturally fails to search visually a large subset of the surface as this is not considered in the optimization.

A. Simulation Studies

In order to evaluate and fine-tune the proposed HABPlanner prior to its experimental deployment, two simulation studies were conducted inside a) a cave network and b) an underground urban environment. The cave environment is dark consisting of large open spaces whereas the urban environment has better lighting conditions and is made of several large rooms connected by subway tunnels. The simulations utilized the RotorS Simulator [27] and a quadrotor MAV model integrating a LiDAR sensor with $[F_H^D, F_V^D] = [360, 30]^\circ$, $d_{\max} = 50$m as the depth sensor $\mathcal{S}^D$ and a color camera with $[F_H^V, F_V^V] = [120, 60]^\circ$, $r_{\min} = 3$m, $r_{\max} = 7$m as the camera sensor $\mathcal{S}^V$. The robot bounding box D_R was set to *length*×*width*×*height* = $1.1 \times 1.1 \times 0.6$m.

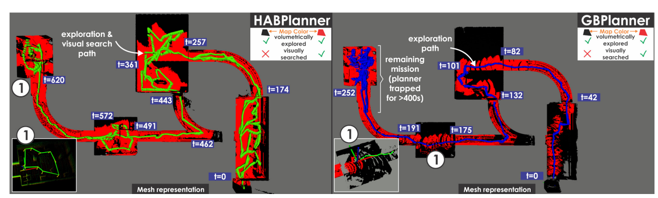

Fig. 5. Results of the proposed HABPlanner in an urban environment involving corridors and very large rooms and its comparison with GBPlanner. Red areas in the map are both explored and seen with the camera, whereas black regions are only explored volumetrically but not seen by the camera sensor. As shown HABPlanner individually focuses to visually search each room it encounters due to its hypergame formulation.

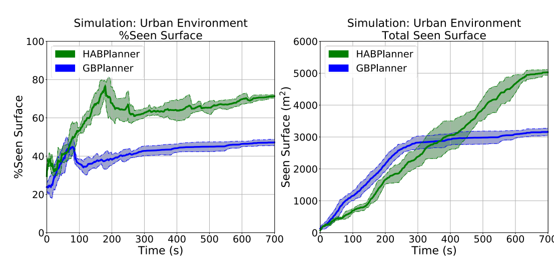

Fig. 6. Performance metric comparison in urban environment. The smaller scale of this environment allows GBPlanner to perform well in terms of visual coverage too in the corridors, however, HABPlanner soon outperforms it especially because of its local coverage of the encountered rooms which the GBPlanner fails to visually cover.

The performance of the proposed HABPlanner was compared against our previous Graph-based exploration planner

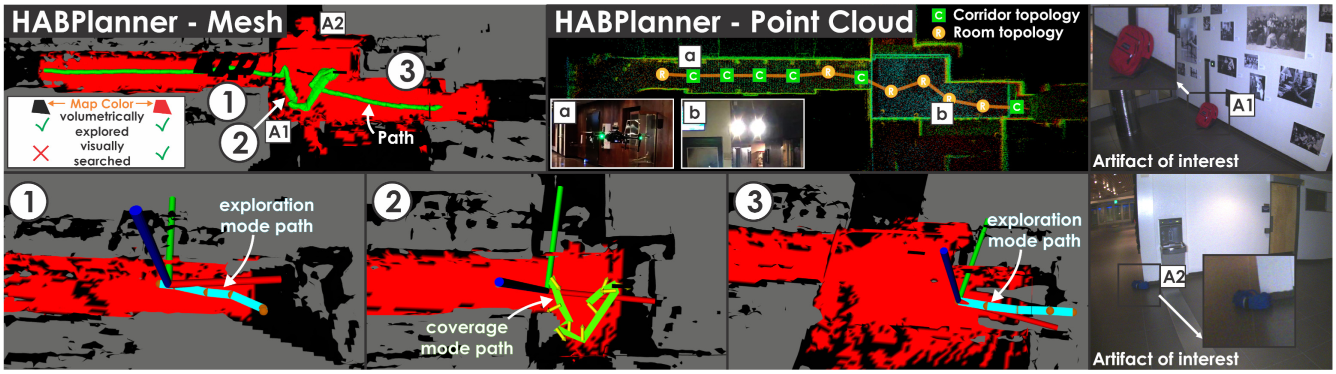

Fig. 7. Results of the experiment conducted in the Arts Building of the University of Nevada, Reno. The sub-figures (1), (2) and (3) show the exploration and coverage paths at various instances of the mission. Red areas in the map are both explored and seen with the camera sensor, whereas black regions are only explored volumetrically but not seen by the camera sensor. Motivated by our participation in the DARPA Subterranean Challenge [28], two artifacts of interest, a backpack and a rope, were placed in the environment for the robot to find. The artifact locations A1 and A2 are shown in the top left sub-figure. The robot was able to detect both as shown in the onboard camera images on the right. The central sub-figure shows the reconstructed pointcloud map of the environment along with the topological map. In spite of certain incorrect inferences, the robot was able to perform efficient exploration and coverage of the environment given the complete hypergame formulation.

(GBPlanner) [10] to further verify the importance of the adaptive methodology. Both planners succeed in the volumetric exploration task but perform differently in terms of visual coverage. The metrics used to evaluate the performance of both the planners are a) the total amount of surface seen by the visual camera $\mathcal{S}^V$ and b) the percentage of total mapped surface seen by $\mathcal{S}^V$. Five finite time missions were conducted for each planner in both the environments starting from the same location. Figures 3 and 5 present the results for the cave and urban setting respectively while Figures 4 and 6 present the comparison of respective performance metrics. Especially in the large-scale and dark cave environment but also in the smaller and more well-lit urban setting, HABPlanner outperforms GBPlanner. For the first environment, HABPlanner outperforms for the entire mission. In the latter, the more focused search of each individual room of HABPlanner slows it down initially but pays-out halfway through the mission.

B. Experimental Evaluation

Alongside the simulation studies, an experimental evaluation was conducted. In particular, a quadrotor aerial robot as in [10] integrating both a depth sensor $\mathcal{S}^D$ using a 3D LiDAR and a camera sensor $\mathcal{S}^V$ was utilized. The parameters for these sensors are $\mathcal{S}^D$: $[F_H^D, F_V^D] = [360, 30]^\circ$, $d_{\max} = 50$m and $\mathcal{S}^V$: $[F_H^V, F_V^V] = [85, 64]^\circ, r_{\min} = 3$m, $r_{\max} = 7$m. The robot bounding box D_R is considered equal to *length*$\times$*width*$\times$*height* = $1.4 \times 1.4 \times 0.8$m. The test was conducted in a subset of the Arts Building of the University of Nevada, Reno and involved the exploration and visual search of a topology involving sequentially a) a long corridor, b) a room, and c) a smaller corridor before a dead-end. The light conditions for most of the mission were dim and especially in the room were dark, thus requiring the robot to utilize its onboard illumination and necessitating dynamic adaptation of the useful camera frustum range for **VisualGain** calculation and map annotation. The experimental results are presented in Figure 7. A test with GBPlanner was also conducted in the same environment and the performance comparison is shown in Figure 8. Due to the very small size of the environment GBPlanner has good visual coverage as well. However, due to the ability to adapt to the varying topology and light conditions, HABPlanner is able to outperform GBPlanner throughout mission.

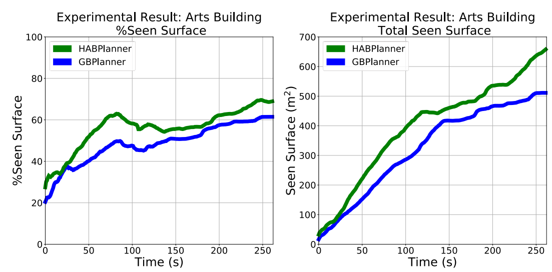

Fig. 8. Performance metric comparison in the Arts Building. The very small scale of this environment allows GBPlanner to perform good enough, however HABPlanner still outperforms it in all the instances of the mission and especially when entering the larger room (80s).

VI. CONCLUSIONS

In this work, a hypergame-based adaptive behavior path planner for simultaneous exploration and visual search of unknown environments was proposed. The method utilizes a hypergame formulation to select the next-best planning behavior from the available discrete behaviors optimised for either exploration or visual search. The ability of the planner to infer current environmental challenges and incorporate its own performance in the decision process to correct itself are its key features which allow it to efficiently explore and visually search environments with diverse geometric complexities and light conditions. The method was verified through a set of simulation and experimental studies and a comparative analysis was shown against a state-of-the-art exploration path planner.

References

[1] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *IEEE International Conference on Robotics and Automation (ICRA), 2014*. IEEE, 2014, pp. 6136–6143.

[2] A. Jones, M. Schwager, and C. Belta, "A receding horizon algorithm for informative path planning with temporal logic constraints," in *IEEE International Conference on Robotics and Automation*. IEEE, 2013.

[3] D. Berenson, P. Abbeel, and K. Goldberg, "A robot path planning framework that learns from experience," in *IEEE International Conference on Robotics and Automation (ICRA), 2012*.

[4] M. Popovic, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran, "Online informative path planning for active classification using uavs," *arXiv preprint arXiv:1609.08446*, 2016.

[5] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty–aware receding horizon exploration and mapping using aerial robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2017.

[6] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2016. [Online]. Available: https://github.com/ethz-asl/nbvplanner

[7] L. Yoder and S. Scherer, "Autonomous exploration for infrastructure modeling with a micro aerial vehicle," in *Field and Service Robotics*. Springer, 2016, pp. 427–440.

[8] C. Connolly *et al.*, "The determination of next best views," in *IEEE International Conference on Robotics and Automation 1985*.

[9] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *CIRA'97*. IEEE, 1997, pp. 146–151.

[10] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "Graph-based path planning for autonomous robotic exploration in subterranean environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[11] M. R. Dharmadhikari, T. Dang, L. Solanka, J. B. Loje, D. H. Nguyen, N. Khedekar, and K. Alexis, "Motion primitives-based path planning for fast and agile exploration using aerial robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2019.

[12] S. Arora and S. Scherer, "Randomized algorithm for informative path planning with budget constraints," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4997–5004.

[13] T. Dang, C. Papachristos, and K. Alexis, "Autonomous exploration and simultaneous object search using aerial robots," in *2018 IEEE Aerospace Conference*. IEEE, 2018, pp. 1–7.

[14] W. Tabib, M. Corah, N. Michael, and R. Whittaker, "Computationally efficient information-theoretic exploration of pits and caves," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3722–3727.

[15] K. Shubina and J. K. Tsotsos, "Visual search for an object in a 3d environment using a mobile robot," *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 535–547, 2010.

[16] P.-E. Forssén, D. Meger, K. Lai, S. Helmer, J. J. Little, and D. G. Lowe, "Informed visual search: Combining attention and object recognition," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 935–942.

[17] P. Bennett, "Toward a theory of hypergames," *Omega*, vol. 5, no. 6, pp. 749–751, 1977. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0305048377900561

[18] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[19] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Autonomous Robots*, pp. 1–25, 2015.

[20] C. Papachristos, M. Kamel, M. Popović, S. Khattak, A. Bircher, H. Oleynikova, T. Dang, F. Mascarich, K. Alexis, and R. Siegwart, "Autonomous exploration and inspection path planning for aerial robots using the robot operating system," in *Robot Operating System (ROS)*. Springer, 2019, pp. 67–111.

[21] T. Dang, C. Papachristos, and K. Alexis, "Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[22] J. Delmerico, E. Mueggler, J. Nitsch, and D. Scaramuzza, "Active autonomous aerial exploration for ground robot path planning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 664–671, 2017.

[23] T. Andre and C. Bettstetter, "Collaboration in multi-robot exploration: to meet or not to meet?" *Journal of Intelligent & Robotic Systems*, vol. 82, no. 2, pp. 325–337, 2016.

[24] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[25] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[26] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the travelling-salesman problem," *Operations Research*, vol. 21, pp. 498–516, 1973.

[27] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors-a modular gazebo mav simulator framework," in *Robot Operating System (ROS)*. Springer, 2016, pp. 595–625.

[28] Defense Advanced Research Projects Agency (DARPA), "DARPA Subterranean Challenge." [Online]. Available: https://subtchallenge.com/