AUTONOMOUS VEHICLES

Representation granularity enables time-efficient autonomous exploration in large, complex worlds

C. Cao*, H. Zhu, Z. Ren, H. Choset, J. Zhang

We propose a dual-resolution scheme to achieve time-efficient autonomous exploration with one or many robots. The scheme maintains a high-resolution local map of the robot's immediate vicinity and a low-resolution global map of the remaining areas of the environment. We believe that the strength of our approach lies in this low- and high-resolution representation of the environment: The high-resolution local map ensures that the robots observe the entire region in detail, and because the local map is bounded, so is the computation burden to process it. The low-resolution global map directs the robot to explore the broad space and only requires lightweight computation and low bandwidth to communicate among the robots. This paper shows the strength of this approach for both single-robot and multirobot exploration. For multirobot exploration, we also introduce a "pursuit" strategy for sharing information among robots with limited communication. This strategy directs the robots to opportunistically approach each other. We found that the scheme could produce exploration paths with a bounded difference in length compared with the theoretical shortest paths. Empirically, for single-robot exploration, our method produced 80% higher time efficiency with 50% lower computational runtimes than state-of-the-art methods in more than 300 simulation and real-world experiments. For multirobot exploration, our pursuit strategy demonstrated higher exploration time efficiency than conventional strategies in more than 3400 simulation runs with up to 20 robots. Last, we discuss how our method was deployed in the DARPA Subterranean Challenge and demonstrated the fastest and most complete exploration among all teams.

INTRODUCTION

Autonomous exploration is crucial for applications where environments are often inaccessible or hazardous to humans, such as in search and rescue, surveillance, and planetary exploration (*1*, *2*). Here, we consider determining paths for one or more robots to navigate in an a priori unknown environment, such that the robots can be directed to thoroughly sweep their sensors to make a representation, namely, a map, of the target environment. Such a problem is closely related to coverage path planning (*3*), which seeks a path that allows sensors to sweep over all objects in an environment. The problem of finding the shortest path to achieve such a complete coverage was shown to be non-deterministic polynomial-time hard (NP-hard) (*4*). Here, the exploration problem focuses more on discovering the environment and computing the coverage path on the fly. As a result, computing the exploration paths must be performed iteratively and incrementally to process incoming sensor data, which can be more challenging than solving the coverage path planning problem. Sensor noise, state estimation drift, and terrain variation present additional challenges. The contribution of this work is to prescribe a two-level approach that efficiently explored an unknown region, where efficiency was measured by both explored volume over time and computational runtime. We validated this approach in several large-scale, complex, and cluttered environments both in simulation and in the real world (Movie 1), including the Subterranean (SubT) Challenge held by the Defense Advanced Research Projects Agency (DARPA) (*5*).

Much prior work in exploration based their approaches on either a choice of representation for the environment or the means by which new information was acquired. In the former case, a graph was usually formed to represent the robot's free space. Common approaches to form such a graph include sampling-based methods (*6*–*12*) and topological ones (*13*–*17*). Exploration was then achieved by incrementally growing these structures. In the latter case, the approaches acquired new information by directly growing the "known" free space, which we term the explored areas. These approaches are often called frontier based because the robot sought to expand the boundaries of the explored areas. Alternatively, expanding the explored areas can be achieved by optimizing an information metric, such as entropy, to gather information about unknown portions of the environment (*18*–*22*). To the best of our knowledge, these approaches tend to be "greedy" in that they look for the closest frontier or closest local minimum in entropy to conduct exploration (*23*–*25*). We believe that nongreedy approaches have the potential to improve the time efficiency of exploration.

The issue of time efficiency is more complicated when considering multiple robots that have limited communication among them. Establishing the communication for information sharing among robots often requires them to be physically within a distance range. Such planning requires deliberate consideration to ensure time efficiency when exploring with multiple robots [see (*26*, *27*) for surveys]. Existing strategies for maintaining inter-robot communication are mostly based on heuristics, which disregard whether the information exchange can expedite or hamper the overall exploration. In particular, the rendezvous-based strategy has been widely adopted, where robots gather regularly in predetermined locations and cadences (*28*–*31*). Furthermore, robots are often assigned roles either as "explorers" or as "relays," which are tasked with either exploration or information transmission (*32*, *33*). Other strategies required robots to always maintain connectivity (*34*, *35*) or to regain connectivity frequently (*36*–*38*) or required specialized hardware to deploy communication devices in the environment (*39*, *40*). In those strategies, maintaining communication

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA.
*Corresponding author. Email: ccao1@andrew.cmu.edu

Movie 1. An overview of our proposed exploration method. This video cover is a composite image of the ground robot used in the experiment and the point cloud map constructed during the exploration.

was prioritized over exploration, which can limit the overall exploration time efficiency.

To improve the exploration time efficiency, this paper describes a dual-resolution framework that balances the use of high-resolution information located near the robot and the low-resolution global information across the entire environment. Specifically, the global information was used to effectively guide the detailed exploration that relied on the high-resolution local information. Such an approach was first introduced in our prior work (*41*, *42*) for autonomous exploration using a single ground or aerial vehicle. This paper further extends the approach for multirobot exploration. Specifically, leveraging the dual-resolution world representation, we bring forward the "pursuit" strategy for inter-robot communication, where robots opportunistically pursue each other to facilitate communication if the information exchange can lead to faster exploration overall.

We provide a theoretical analysis of our scheme and evaluated it both in simulation and in real-world experiments. We analyzed the computational complexity and the optimality gap resulting from the dual-resolution representation, specifically, the difference in length between the paths computed with and without the dual-resolution representation. We demonstrated that the optimality gap is bounded by a constant, independent of the environment size (presented in the Supplementary Materials, "Theoretical analysis—Approximation ratio" section). Empirically, for single-robot exploration, our method produced 80% higher exploration time efficiency than three other state-of-the-art methods in both physical and simulation experiments with more than 300 runs. Meanwhile, the computational runtime of our method was 50% lower than that of others. For multirobot exploration, our pursuit strategy produced higher exploration time efficiency compared with the conventional rendezvous-based strategies in more than 3400 simulation runs with up to 20 robots. We further showcased the deployment of our method in the SubT finals, where it achieved the fastest and the most complete exploration among all teams, winning the "Most Sectors Explored Award."

RESULTS

System overview

Here, we describe how the two-level scheme could fulfill both single-robot and multirobot exploration. The processing of the local map at the local level remained the same in both scenarios. The local level solved a combination of the art gallery problem (*43*) and the traveling salesman problem (TSP) (*44*), followed by trajectory optimization to determine the shortest possible path that the robot could follow to thoroughly observe its surrounding areas. The processing of the global map at the global level, on the other hand, solved different combinatorial optimization problems in different exploration scenarios. For single-robot exploration, the global level solved a TSP to determine the shortest possible tour guiding the robot to visit areas that have not yet been fully explored. For multirobot exploration, the global level solved a vehicle routing problem (VRP) (*45*) for allocating robots to explore different areas while minimizing the longest exploration route among all robots. To address the issue of limited communication, where robots need to meet at designated locations within prespecified time intervals for sharing information, the global level solved the VRP with time window constraints (*46*) to find such global paths. Both the conventional rendezvous-based strategies and our pursuit strategy were implemented under such a scheme.

Our scheme could be deployed for autonomous exploration in simulation and in the real world by leveraging a popular three-layer architecture for robotic systems (*47*, *48*), as shown in Fig. 1. Our scheme operated at layer 1 (blue) to compute the exploration path. Layer 2 (red) consisted of navigation modules responsible for avoiding collision while following the exploration path. Layer 3 (green) interfaced with hardware, including sensors and actuators, for state estimation, processing sensory data, and driving the motors.

To facilitate developing and benchmarking algorithms for autonomous exploration, we open-sourced our method together with a software stack that consists of the layer 2 navigation modules, environment models, and visualization tools (www.

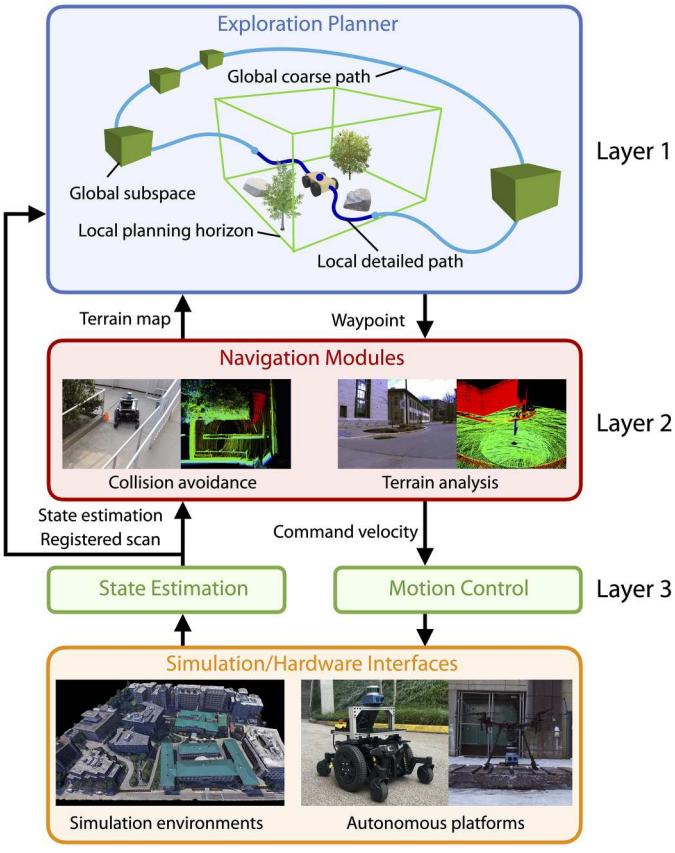


Fig. 1. The three-layer architecture for a robotic system. Our method operates at layer 1. The blue box shows an illustration of our exploration scheme. Inside the local planning horizon (open green box), information is maintained with high resolution to compute a detailed path (dark blue curve). At the global scale, information is maintained with low resolution within subspaces (solid green cubes) to compute a coarse global path (light blue curve). The two paths are concatenated together as the overall exploration path for the robot to follow.

cmu-exploration.com). All simulation experiments discussed in the following were evaluated with the software stack. A detailed discussion of the software stack is presented in the Supplementary Materials ("Development environment and benchmarking system" section).

Experiment platforms

Our experiment platforms included a wheelchair-based ground vehicle and an aerial vehicle, as shown in Fig. 1 (orange box). Both vehicles had their simulated counterparts in the Gazebo simulator (*49*) for simulation experiments. The real and simulated vehicles were equipped with a Velodyne Puck light detection and ranging (LiDAR) for exploration and mapping. For real vehicles, a camera at 640 pixel–by–360 pixel resolution and a micro-electro-mechanical systems (MEMS)–based inertial measurement unit (IMU) were coupled with the LiDAR for state estimations (*50*). The aerial vehicle had a maximum speed of 2.5 m/s in the real world and 5 m/s in the simulation. The maximum speed of the ground vehicle was set at 2 m/s in both the real world and simulation. All experiments were run on a 4.1 GHz i7 computer. All evaluated exploration algorithms were run along the navigation modules from our open-source software stack. The configuration and parameters used in our method are presented in the Supplementary Materials ("Implementation details" section).

Baseline methods

We compared our method against five baseline methods. Specifically, three of them were sampling-based methods: Next-Best-View Planner (NBVP) (*6*), Graph-based Planner (GBP) (*7*), and Motion-primitive–based Planner (MBP) (*8*), which had shown promising results in handling real-world scenarios and were deployed by the winning teams in the SubT challenge. The other two methods included a topology-based strategy, specifically GVGExp (*15*), and an information gain–based strategy, denoted as Information-Theoretical Exploration (ITE) (*51*). Except for ITE, which was implemented by us, all other methods were evaluated using open-source code adapted to our evaluation environments. A brief summary of each of the methods is given as follows.

NBVP is a method that grows a rapidly exploring random tree (RRT) (*52*) in the free space and finds the most informative branch in the RRT as the path to the next viewpoint. GBP is an extension of NBVP, where the method constructs a rapidly exploring random graph (RRG) (*53*) in the entire environment and searches the RRG for routes that maximize the observation in the environment. MBP is a variant of GBP, which constructs a local RRT using motion primitives. The resulting paths are smoother but only span in constrained directions. GVGExp is a method that incrementally builds a generalized Voronoi graph (GVG) to capture the topology of the environment. The robot visits nodes on the GVG that are close to frontiers in a depth-first search fashion. ITE is an information gain–based method. The method uses a motion primitive library and selects the primitive that maximizes the information gain obtained within a local region. To maintain coherency and comply with paper length constraints, the evaluation results of GVGExp and ITE are presented in the Supplementary Materials ("Additional comparison results in simulation" section).

Evaluation environments and metrics

The environments to be explored were chosen or designed to resemble real-world challenges, including but not limited to large-scale space, undulating terrains, convoluted topologies, cluttered obstacles, and unstructured environments. We evaluated the algorithm performance mainly from two perspectives, including exploration time efficiency, measured by average explored volume per second through a run, and algorithm runtime, the time spent in each planning step to compute the exploration path by incorporating sensory information. Detailed statistics on exploration time efficiency and algorithm runtime for all experiments are documented in tables S1 to S4.

In the experiments, a run was terminated if the exploration algorithm reported completion, the vehicle almost stopped moving (less than 9 m of movement within 300 s), or the time limit was met (set to twice the time our method used to explore the environment). For each environment, we restricted the area of exploration by setting a geofence in the software. This involved roughly estimating the expansion of the exploration area and approximating its boundary with a polygon. Frontiers/object surfaces located outside of the geofence were excluded from exploration. Consequently, the robot only explored the area within the designated boundary. The NBVP's performance was dependent on the size of the geofence, which

specified the maximum area within which frontiers were computed and evaluated. The exploration efficiency of NBVP increased as the area became larger, but the runtime also increased, leading to eventual performance degradation. We fine-tuned the area size for NBVP to balance exploration and runtime efficiency for optimal performance in our evaluation environments.

Ground-based autonomous exploration in the real world

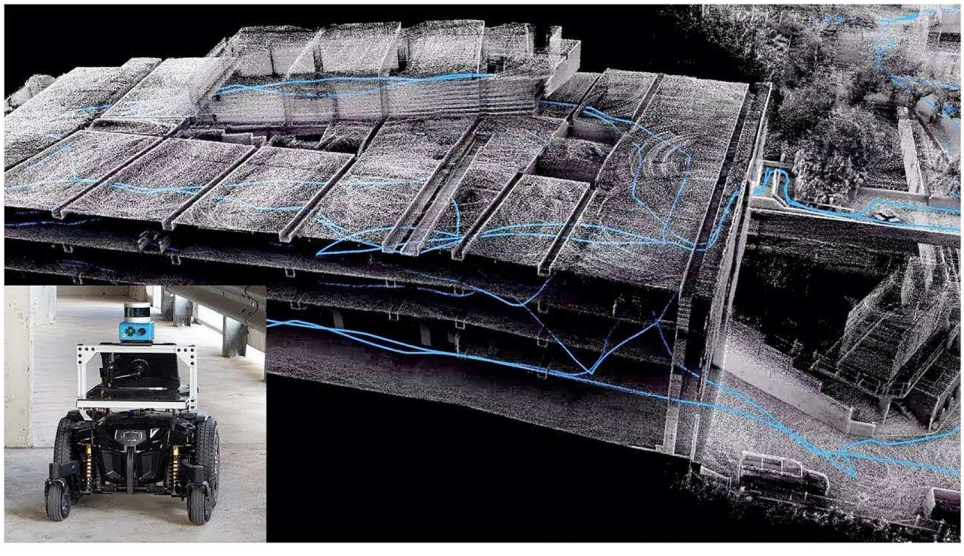
Figure 2 shows the results of three real-world experiments conducted with a wheeled vehicle. Test 1 was conducted in a multistory garage in the real world (Movie 2). The garage had four levels connected by a spiral driveway. An outdoor patio was connected to the top level, and a long corridor was connected to the bottom level. The starting point of exploration was located at the garage entrance at the top level. Figure 2A(i) shows trajectories from all methods separately. The point cloud map and trajectory from our method are shown in Fig. 2A(ii). Figure 2A(iii) compares the explored volume and algorithm runtime over time, where our method achieved the highest exploration time efficiency with the lowest runtime. Only our method could explore the whole environment, which reported completion after traveling 1839 m in 1907 s. All other methods terminated early inside the garage and missed the top-level patio. In particular, NBVP reached the time limit whereby the robot moved back and forth for an extended period of more than 500 s. GBP, on the other hand, failed to plan feasible paths toward the end of the exploration, causing the robot to halt. Last, MBP experienced prolonged runtime after 25 min of exploration, causing the robot to move briefly before becoming inactive for extended periods while waiting for the planning to complete. The repetitive trajectories exhibited by NBVP and MBP, as well as the stopping point of GBP, are highlighted in Fig. 2A(i) by red dotted-line boxes and a red arrow, respectively. All methods except ours exhibited "spikes" in their runtime profiles, which account for the long planning time to process the accumulated information. The runtime of our method constantly stayed below 1 s through the run.

Test 2 was conducted in a large indoor environment consisting of lobbies and dining areas connected by long corridors (movie S1). The primary challenge of the environment was its complex and branching topology. Figure 2B shows the results of test 2 in a layout similar to Fig. 2A. The exploration started from one end of the environment, as indicated by the blue dot in Fig. 2B(i). Our method completed the exploration after traveling over 988 m in 1167 s. Other methods left large portions of the environment uncovered. GBP and MBP exhibited periodic switching between two goal locations, resulting in repetitive back-and-forth movements until the end of the run. Conversely, NBVP encountered difficulty traversing a long corridor and repeatedly veered toward the corners of the building, as shown in the close-up view in Fig. 2B(ii). The point cloud map was manually cleaned for visualization purposes after the exploration, where noise points resulting from pedestrians passing by, reflections of the glass, and structures seen through windows were removed.

Test 3 was conducted in a challenging environment comprising a cluttered indoor space and hilly outdoor terrains (Movie 3). The exploration started from within a cluttered garage as shown in Fig. 2C(i) and proceeded through an open outdoor region, encompassing multiple hills with a cumulative elevation change of 16.0 m, shown in Fig. 2C(ii). The environment posed additional challenges, including a mix of spacious and confined areas, restricted passages, and perceptual challenges resulting from the presence of unstructured vegetation, leading to noisy point cloud data. Only our method was evaluated in this environment, because its difficulty level exceeded the capabilities of other methods. The resulting point cloud maps and robot trajectories are shown in Fig. 2C. Our method completed the exploration after traveling over 1403 m in 1217 s.

Ground-based autonomous exploration in simulation

We further conducted a series of tests in five simulation environments. In these tests, we provided the best possible results from human practice as a reference. The results were collected from a human operator who teleoperated a simulated robot with a joystick



Movie 2. Autonomous exploration using a ground vehicle in a four-story garage in test 1.

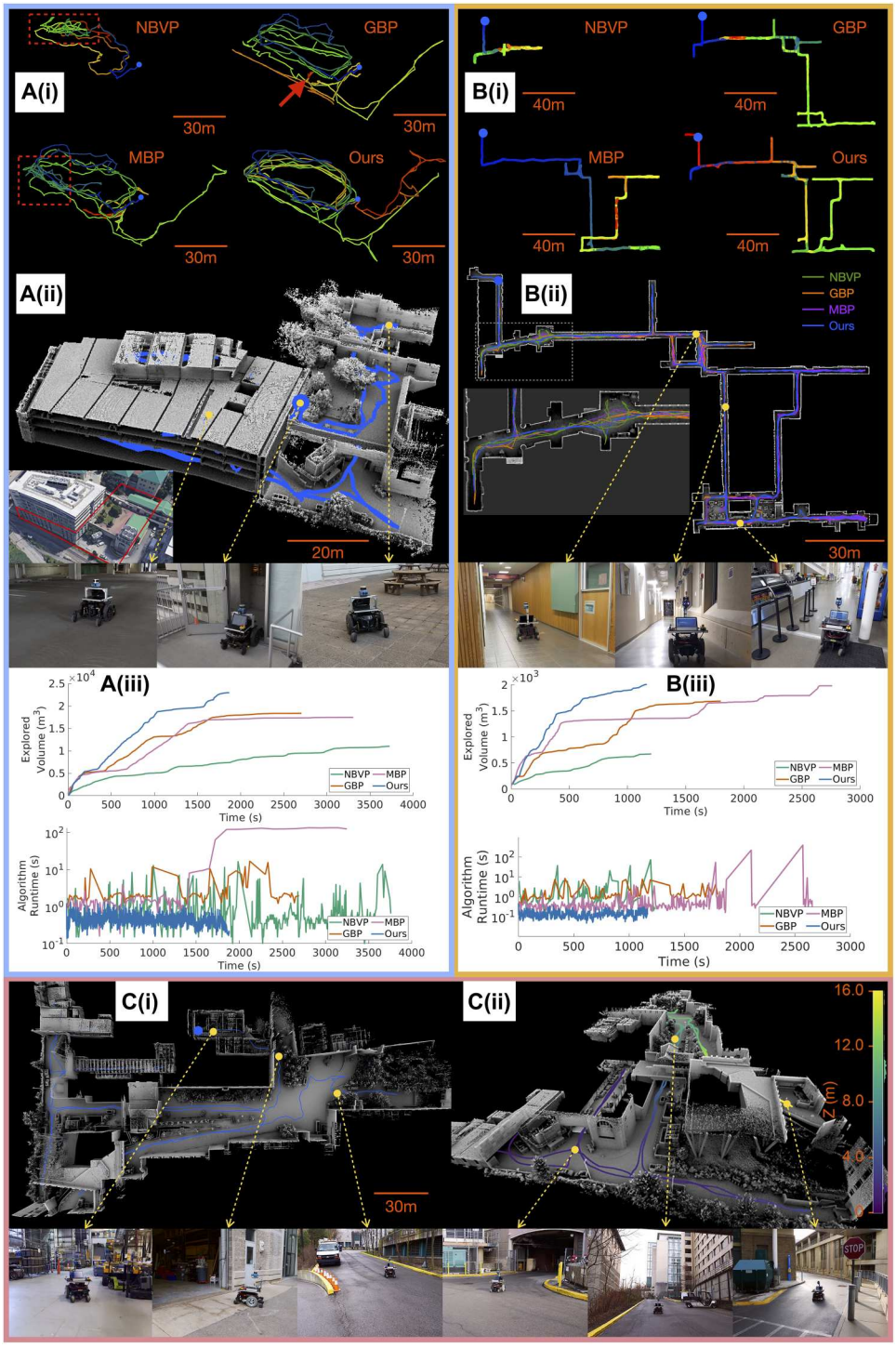
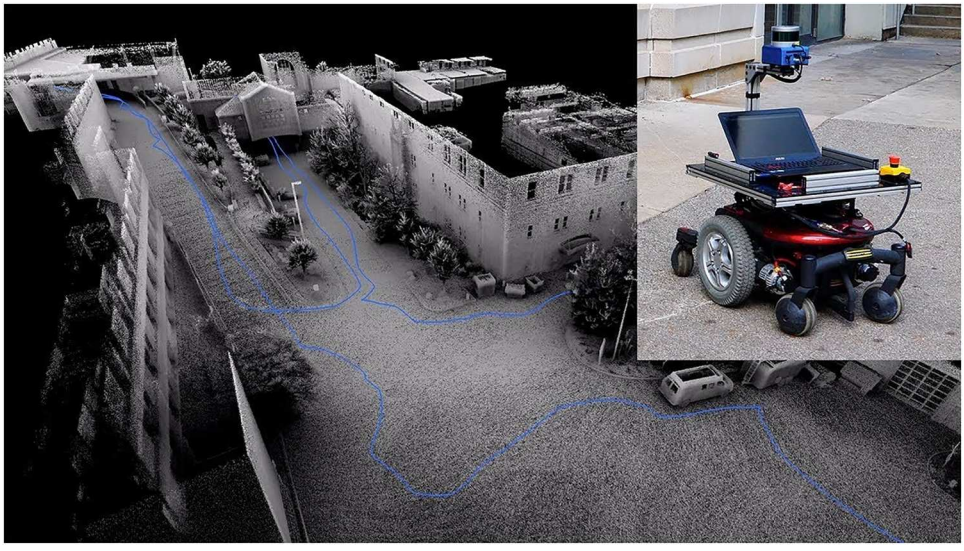


Fig. 2. Ground-based autonomous exploration in the real world. The starting points of exploration are indicated by the blue dots. The point cloud maps were produced by stacking individual LiDAR scans together using the state estimation and mapping algorithm (*50*). (**A**) Test 1: A four-story garage with a connected patio. [A(i)] shows the trajectories from each method. The trajectories are color-coded to indicate temporal progression, starting with blue and ending in red. The point cloud map resulting from our method is shown in [A(ii)]. Explored volume and algorithm runtime over time are shown in [A(iii)]. (**B**) Test 2: An indoor environment with long corridors connecting open areas. The panel layout is similar to (A). (**C**) Test 3: A large-scale environment mixing cluttered indoor and hilly outdoor spaces. [C(i)] and [C(ii)] show the same point cloud map and robot trajectory viewed from different perspectives. Only our method was evaluated in this environment, because the difficulty was beyond what other methods can handle, and thus, there are no plots for the other methods.



Movie 3. Autonomous exploration using a ground vehicle in a complex environment in test 3.

controller to explore each environment. The human operator had prior knowledge of the environment and had practiced it many times before collecting the best results. All methods were run with 10 trials in each environment. Our method could completely explore all environments with the highest exploration efficiency, which was about 80% of the human-level efficiency, whereas the other methods were about 5 to 40% of the human performance, as shown in table S1. In addition, our method had the lowest runtime in all environments, which was about an order of magnitude lower than that of the other methods, as shown in table S2.

Figure 3A shows the results of test 4 in the campus environment. The environment was modeled from a part of the Carnegie Mellon University campus. The challenges of this environment included three-dimensional (3D) terrain and unstructured obstacles. The space for exploration consisted of two parts connected only by a long and narrow bridge. The human practice covered one part of the environment completely before moving on to the other part across the bridge, eliminating the need to come back through the bridge again. Although there were instances where our method finished one side of the environment before crossing the bridge, as shown by the trajectories in the figure, there was no guarantee that this would occur consistently. The other methods often failed to create nodes that could traverse the narrow bridge entrance because of the random sampling process used in constructing the RRT or RRG, resulting in a failure to explore across or return from the bridge.

Figure 3B shows the results of test 5 in the garage environment. It was one of the most challenging environments because of its large scale, multiple levels, and a mix of open space with narrow entrances (movie S2). The human practice explored upward level by level. At each level, it visited all connected side rooms before moving to the next level, thus eliminating the need to return to lower levels. Our method could explore the entire environment but tended to explore some side rooms at the end, causing the overhead of moving between levels. Other methods exhibited back-and-forth movements, prolonged runtime, or inability to navigate through the narrow entrances of side rooms, consequently leaving large areas of the environment unexplored.

Figure 3C shows the results of test 6 in the indoor environment. The environment consisted of lobby rooms connected by long and narrow corridors. Obstacles and occlusion created by objects of irregular shapes and thin structures presented navigation challenges. All methods, except for NBVP, could fully cover the environment in most trials.

Figure 3D shows the results of test 7 in the tunnel environment. The environment had the largest footprint (330 m by 250 m) and was modeled after the Bruceton Research Mine in Pittsburgh, PA. The challenge of this environment was presented by its convoluted topology consisting of loops and junctions. GBP and our method could explore the entire tunnel, whereas NBVP and MBP left large portions of the environment unexplored. In particular, NBVP exhibited back-and-forth movements, whereas MBP experienced prolonged runtime.

Figure 3E shows the results of test 8 in the forest environment. The environment was highly unstructured, with no predefined paths. The vehicle could almost go in any direction. The best human practice followed a predefined route with a lawn mowing pattern. All methods except for NBVP were capable of fully covering the environment. However, GBP and MBP revisited places redundantly, which took more time to complete the exploration.

Aerial-based autonomous exploration

Test 9 used the simulated aerial vehicle to explore the campus environment as in test 4. The aerial vehicle must fly up and down to cover the roofs of low-rise buildings and the exteriors of high-rise buildings while avoiding extruding building structures. Each method was run with 10 trials starting from the same position indicated by blue dots. Figure 4A(i) shows the resulting point cloud maps and trajectories of all methods from a representative trial. Our method could cover the entire environment after traveling 1318 m on average and 366 s in the longest run. The time limit for the other methods was set to four times that of our longest run. Within the time limit, none of the other methods could completely explore the

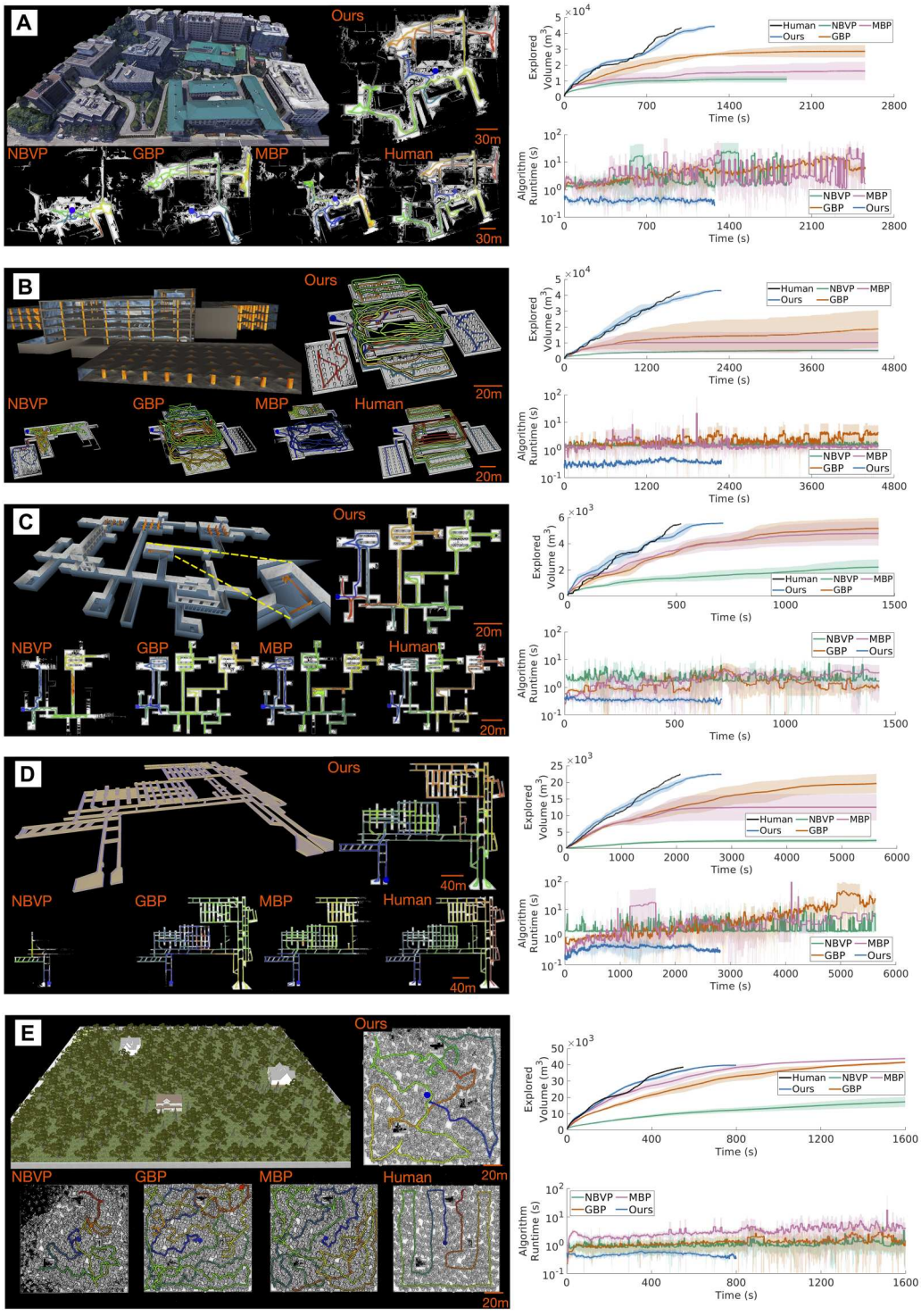


Fig. 3. Ground-based autonomous exploration in simulation. (**A** to **E**) Results of tests 4 to 8, where the environment overview, resulting point cloud maps, and robot trajectories from representative trials from all methods are shown alongside the comparisons of exploration time efficiency and runtime plots. The robot trajectories are color-coded to indicate temporal progression, starting with blue and ending in red. The curves show mean values, and shaded areas represent the SD over 10 trials.

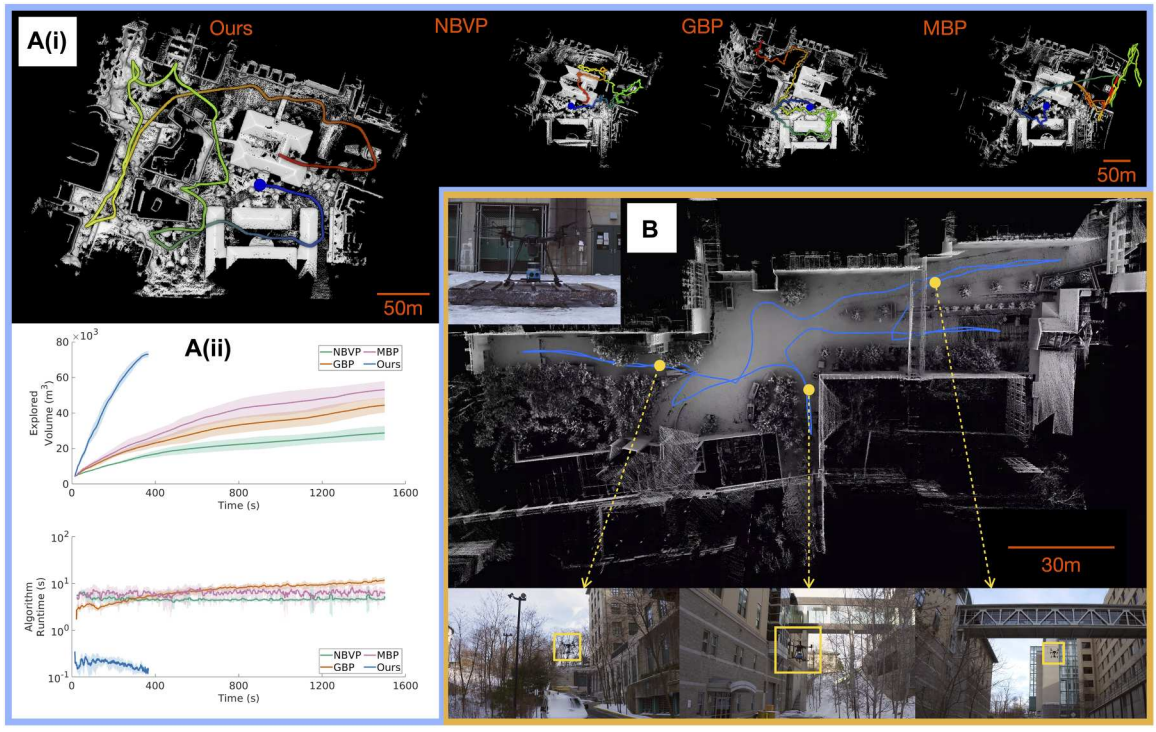


Fig. 4. Aerial-based autonomous exploration. (**A**) Results of test 9, which used an aerial vehicle to explore the campus environment in simulation. [A(i)] shows representative resulting point cloud maps and trajectories of all methods. The blue dots indicate the starting point. The robot trajectories are color-coded to indicate temporal progression, starting with blue and ending in red. Explored volume and algorithm runtime over time are shown in [A(ii)]. The curves show mean values, and shaded areas represent the SD over 10 trials. (**B**) Results of test 10 using an aerial vehicle to explore a complex environment over 3D terrain in the real world. Our method finished exploring the environment after traveling over 550 m in 300 s.

environment. Because of the 3D and more complex space to be explored, all other methods exhibited longer runtimes than the ground vehicle cases. Consequently, the robot experienced notably more idle time while awaiting planning results. Figure 4A(ii) gives results of comparison of the exploration time efficiency and algorithm runtime.

Test 10 used an aerial vehicle to explore an outdoor environment in the real world (movie S3). The environment was where test 3 was conducted but included exteriors of buildings and space that were not reachable by the ground robot. In addition, the overhanging walkways and tree branches challenged the reactiveness of the exploration planner. Because of the limited computational resources on the aerial vehicle, only our method was evaluated in this test. Figure 4B shows the results. Our method finished the exploration after traveling over 550 m in 300 s.

Multirobot exploration with limited communication

We extended our method for multirobot exploration. We investigated how the exploration time efficiency increased with more robots deployed in the environment. In addition, our method was adapted to implement different coordination strategies for exploration under communication constraints. We investigated four communication strategies described in the following.

"Full comms" is a hypothetical scenario where robots always have full communication with each other. "Single-point rendezvous" is the conventional rendezvous strategy where robots meet regularly at predetermined locations to exchange information. "Multipoint rendezvous" is a variant of the rendezvous strategy where robots gather at different locations to expand the communication range while waiting for others. "Pursuit" is our proposed strategy where robots opportunistically pursue each other to communicate if the information exchange can improve the overall exploration time efficiency (movie S4).

All four communication strategies were realized under our proposed scheme. We conducted experiments to evaluate the four strategies in the five simulation environments as in tests 4 through 8. The number of robots deployed for exploration ranged from 2 to 20 at a step size of 2. We ran 10 trials for each combination of environment, number of robots, and communication strategy, resulting in 2000 trials in total. The distance range for communication was set to 30 m to resemble a realistic wireless connection range.

Furthermore, we investigated the influence of communication range on exploration efficiency. We conducted experiments in the tunnel environment using varying communication ranges ranging from 10 to 300 m while maintaining a constant number of robots, namely, 5, 10, 15, and 20. Each configuration of the experiment was run for 10 trials, which resulted in more than 1400 trials in total.

Figure 5A shows the exploration time efficiency over increasing numbers of robots for each environment. Overall, full communication resulted in the highest exploration time efficiency. Our pursuit strategy came in second and was superior to both rendezvous-based strategies in all five environments. The multipoint rendezvous strategy had a slightly higher performance than the single-point rendezvous strategy. For all environments, increasing the number of robots

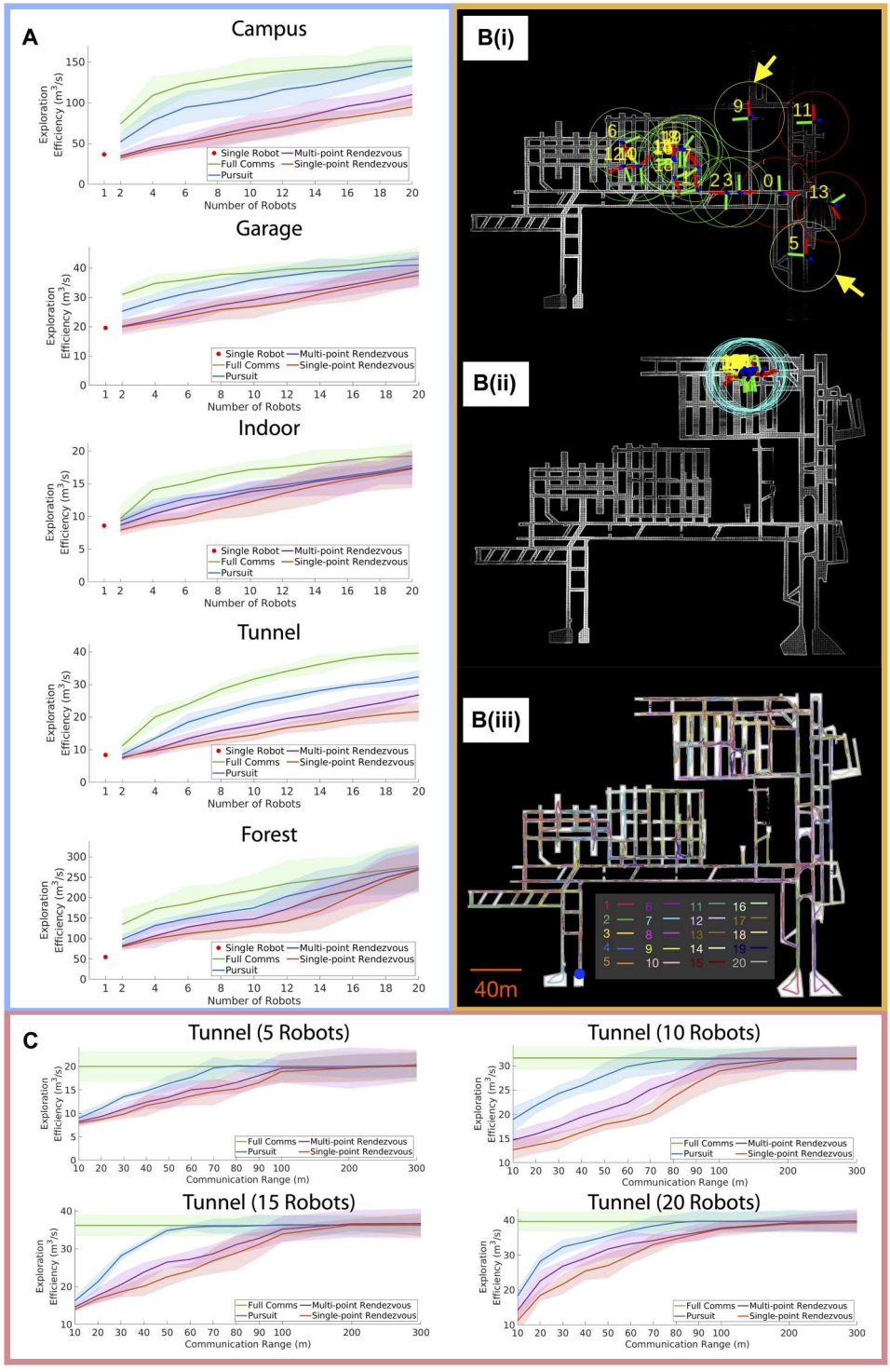


Fig. 5. Results of multirobot exploration under communication constraints. (**A**) Resulting exploration time efficiencies with mean values (curves) and SD (shaded areas) over 10 trials in all five simulation environments against increasing numbers of robots. [**B**(**i**) and **B**(**ii**)] Two instances of 20 robots exploring the tunnel environment using our pursuit strategy. Coordinate frames show the robots' positions. Circles show the communication range (30 m). Red circles indicate that the robot is out of comms with any robots. Green circles indicate that the robot is in comms with at least one robot. Yellow or cyan circles indicate that the robot is pursuing other robots to deliver or request information, respectively. [**B**(**iii**)] Resulting point cloud map and trajectories from the 20-robot exploration. (**C**) Resulting exploration time efficiency against increasing communication ranges with a fixed number of robots (5, 15, 10, and 20) in the tunnel environment.

yielded diminishing returns in time efficiency. The diminishing returns came sooner in environments with simpler topologies, such as the campus, garage, and indoor environments. In addition, a large number of robots could form a well-connected communication network covering most of the environment; thus, the performance gaps between full comms and other strategies became smaller with more robots. In small environments like the indoor and forest environments, all strategies exhibited similar exploration time efficiency eventually.

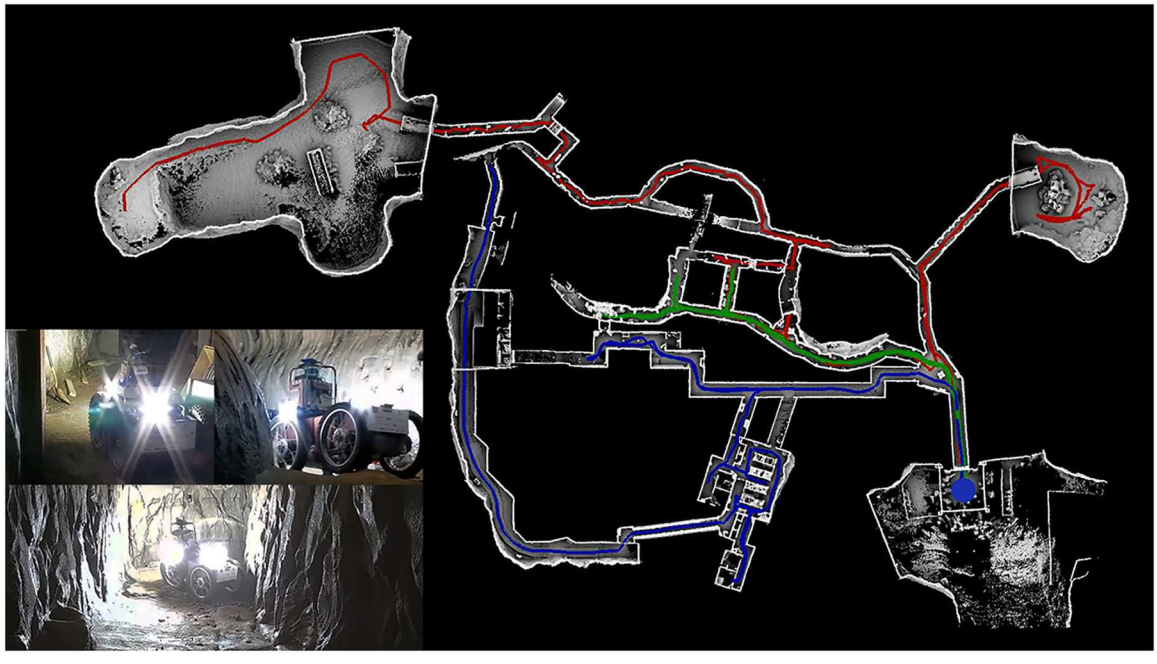
Figure 5B(i) shows an instance of one of the trials where 20 robots were deployed to explore the tunnel environment. Robot positions are indicated by coordinate frames, where circles around them show the communication range. The circles are color-coded to indicate the robots′ communication status. Red indicates that a robot is out of comms with all other robots. Green indicates that a robot is in communication with at least one other robot. Yellow (pointed with arrows) indicates that the robot has surplus information and decides to deliver it to others. Cyan indicates that the robot has nowhere to explore and thus pursues others to request more information. In Fig. 5B(ii), all robots had finished exploring areas known to themselves and thus pursued each other to request more information, where they spontaneously gathered at a location to exchange information. Once they had confirmed from others that there was nowhere else to explore in the entire environment, they reported the completion of the exploration and returned to the starting position. Figure 5B(iii) shows one of the resulting point cloud maps and trajectories from the 20-robot exploration.

Figure 5C shows the exploration time efficiency over increasing communication ranges in the tunnel environment. The results indicated that all strategies converged to full comms level of performance when the communication range was sufficiently large (100 to 300 m). The pursuit strategy outperformed other strategies when the communication range was too small (less than 10 m), albeit with a relatively small performance gap. The results revealed that the pursuit strategy attained 39% higher exploration efficiency on average than other strategies in the mid-range communication distances (20 to 50 m), which are typical in real-world scenarios. This confirmed the strength and practical applicability of our method.

Deployment in the DARPA SubT Challenge

The DARPA SubT Challenge highlighted autonomous navigation and exploration in underground, GPS-denied environments. The challenge involved three types of environments: tunnel systems, urban underground, and cave networks. Teams deployed a fleet of autonomous vehicles to search for artifacts (backpacks, mobile phones, and so on) and reported their locations. A human operator was allowed to command the vehicles from the entrance of the environment over intermittent wireless networks.

Our method was deployed by Team Explorer (*54*) in the competition with three ground vehicles. An overview of the system is presented in the Supplementary Materials ("System integration" section). Figure 6A shows a representative result from a competition run in the Urban Circuit, which was held within an abandoned nuclear plant at Satsop Business Park in Elma, WA. One vehicle used our method to explore the environment fully autonomously, traveling over 886 m in 1458 s. Figure 6B shows the result of the final competition held in Louisville Mega Cavern, KY. The course combines tunnel, urban, and cave settings with complex topology. Dynamic obstacles and heavy fogs were present in the course for additional perception and planning challenges. Three vehicles using our method explored the environment collaboratively. The three vehicles traveled over 596.6, 499.8, and 445.2 m, respectively, over a time span of 2259 s, which covered 26 of 28 sectors of the competition course. Our vehicles explored more than 80% of the environment in the first half of the hour-long competition. The exploration was the fastest and the most complete among all teams, as



Movie 4. Three ground robots running our method were deployed in the DARPA SubT Challenge.

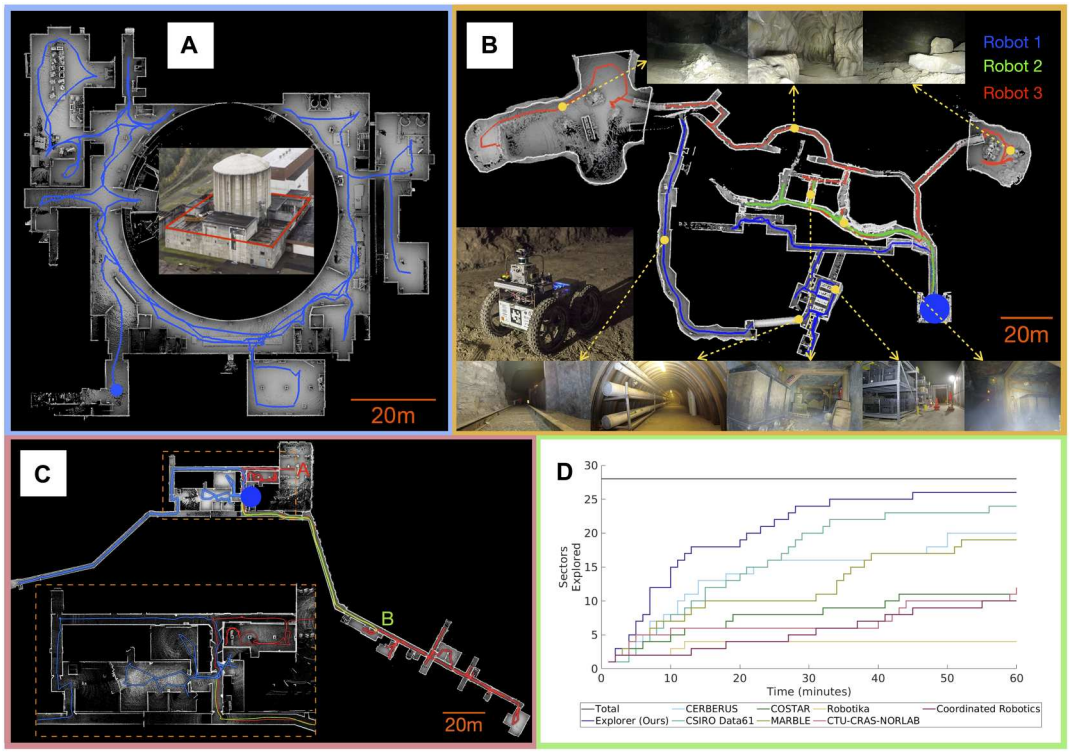


Fig. 6. Deployment in the DARPA Subterranean Challenge. Our method was deployed in two events of SubT. (**A**) The urban circuit took place at Satsop Business Park in Elma, WA in February 2020. Our vehicle traveled over 886 m in 1458 s to explore the entire floor. (**B**) The final event took place in Louisville Mega Cavern, KY in September 2021. Three ground vehicles (blue, green, and red curves) running our method collaboratively explored the environment. The environment included convoluted topology, rough terrains, and cluttered obstacles. (**C**) Sectors explored over time of the three ground vehicles in the SubT final event. The statistics were gathered from the official competition playback from DARPA. Team Explorer (ours) had the fastest and the most complete exploration. (**D**) Practice run with three ground vehicles in an indoor environment. The start point of the exploration is indicated by the blue dot. The vehicle of the red trajectory pursued the vehicle of the green trajectory for communication from point A to point B.

shown in Fig. 6C, which led to our team winning the Most Sectors Explored Award granted by DARPA (Movie 4).

In addition, we present a test run conducted in a real-world indoor environment as part of our preparations for the SubT final event. The results are shown in Fig. 6D. The experiment showcased the effectiveness of our pursuit strategy in facilitating collaborative exploration among three ground robots denoted by red, green, and blue colors. In the figure, robot 1 (red) initiated the exploration, followed by robot 2 (blue) and robot 3 (green). At point A, robot 1 completed its exploration and began to pursue robot 3. At point B, robot 1 reached robot 3 to exchange information and explored the rest of the environment thereafter.

DISCUSSION

In terms of single-robot exploration, our method still has considerable performance gaps compared with human practices, because it is based on reasoning solely on the observed geometry of the environment. In future work, we plan to bridge these gaps by incorporating semantic information and spatial prediction to achieve human-level reasoning during exploration tasks. This will require the incorporation of additional sensing modalities, such as 360° cameras, because humans are proficient at using visual information and predictions to guide exploration. For instance, whereas a doorway indicates more unexplored space beyond it for humans to explore, it appears only as a small frontier to robots from a geometric standpoint. In addition, we plan to investigate strategies for prioritizing specific subspaces for early exploration. This approach holds practical value in certain applications, such as search and rescue, where identifying specific areas for exploration may be prioritized for the purpose of locating survivors rather than exploring the entire environment.

For multirobot exploration, we plan to incorporate heterogeneous robots with different mobility modalities, such as wheeled, tracked, legged, and aerial vehicles, and varied sensing configurations with different fields of view and ranges. To accomplish this, we will need to enhance our scheme to represent the robots' heterogeneity in greater detail and formulate combinatorial optimization problems that consider multiple objectives and constraints. In addition, we plan to investigate the challenges of exploring highly dynamic environments, such as crowded areas, where close proximity to pedestrians or other robots presents unique challenges.

MATERIALS AND METHODS

Method overview

The key aspect of our method is the dual-resolution presentation, which facilitates processing at two distinct data granularities: local planning using high-resolution data and global planning using low-resolution data. This approach offers two notable advantages. First,

it allows for long-term planning, whereby global planning guides local planning. Second, it results in low computational runtime, which enables fast replanning. These two features combine to provide superior exploration efficiency and coverage completeness.

In the following sections, we first present the problem definition, where the exploration task is formulated as a surface coverage problem. Next, we introduce the local planning method, which guarantees thorough coverage of the robot's immediate surroundings. Subsequently, we elaborate on how global planning guides local planning toward distant regions, leading to thorough and efficient exploration of the entire environment. Furthermore, we extend this dual-resolution representation to a multirobot exploration scenario, where the low-resolution global representation can be efficiently shared among robots for coordination. Last, we present the pursuit strategy, which leverages the shared global representation to address the constraints of limited communication during collaborative exploration.

For illustrations of the method, detailed algorithms, and implementation details for handling perception noise, accumulative localization errors, and dynamic obstacles, we direct readers to the Supplementary Materials (fig. S2, Algorithms 1 to 4, and the "Implementation details" section). To enhance readability, we have consolidated the frequently used notations in table S7 for easy reference throughout the subsequent sections.

Problem definition

Define $\mathcal{W} \subset \mathbb{R}^3$ as the workspace to be explored. Let $\mathcal{W}_{\text{trav}} \subset \mathcal{W}$ be the traversable subspace. Define viewpoint $\mathbf{v} \in \mathbf{SE}(3)$ to describe the pose of the sensor onboard the vehicle, $\mathbf{v} = [\mathbf{p}_\mathbf{v}, \mathbf{q}_\mathbf{v}]$, where $\mathbf{p}_\mathbf{v} \in \mathcal{W}_{\text{trav}}$ and $\mathbf{q}_\mathbf{v} \in \mathbf{SO}(3)$, respectively, denote the position and orientation. Denote $\mathcal{L} = \{\mathbf{v}_1, ..., \mathbf{v}_n : \mathbf{v}_i \in \mathbf{SE}(3)\}$ as the set of n viewpoints along the vehicle past trajectory. We here use the term "surface" to refer to the generalized surface perceived by the vehicle, specifically, the boundaries between free space and non-free space, where the non-free space includes both occupied and unknown space. Let $\mathcal{S}_\mathbf{v} \subset \mathcal{W}$ be the surfaces perceived by the sensor at $\mathbf{v}$. The union of surfaces perceived at the viewpoints along $\mathcal{L}$ is

$$\mathcal{S} = \cup_{\mathbf{v} \in \mathcal{L}} \mathcal{S}_\mathbf{v} \tag{1}$$

Note that the same surface can be perceived from multiple viewpoints. The surfaces are considered covered if they meet certain criteria given by Eqs. 2 and 3 in the "Local planning" section. Let $\mathcal{S}_{\text{cov}} \subset \mathcal{S}$ denote a subset of the perceived surfaces that are covered so far during the exploration. The perceived yet uncovered surfaces are denoted as $\overline{\mathcal{S}} = \mathcal{S} \setminus \mathcal{S}_{\text{cov}}$. The exploration problem defined here is to find the shortest path, which, when followed by the vehicle, covers $\overline{\mathcal{S}}$. The path must respect the kinematic and dynamic constraints of the vehicle. Let $\mathbf{v}_{\text{current}}$ be the viewpoint located at the vehicle's current sensor pose. The exploration problem can be defined as follows:

Problem 1: Given $\overline{\mathcal{S}}$ and $\mathbf{v}_{\text{current}}$, find the shortest path $\mathcal{T}^*$ formed by n viewpoints $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_n$, and $n \in \mathbb{Z}^+$, which, when followed by the vehicle, covers $\overline{\mathcal{S}}$, such that $\mathbf{v}_{\text{current}} \in \mathcal{T}^*$, and $\mathcal{T}^*$ is "kinodynamically" feasible.

Problem 1 is solved repetitively at each planning cycle during the exploration. We use $\overline{\mathcal{S}}$, the perceived yet uncovered surfaces, to compute the exploration path. When executing the path, we online update $\mathcal{S}$ with up-to-date sensor readings, processing both displaced and newly perceived surfaces. Then, we move surfaces that are covered from $\overline{\mathcal{S}}$ to $\mathcal{S}_{\text{cov}}$ and use $\overline{\mathcal{S}}$ in the next planning cycle; hence, the exploration continues.

When there are $N \in \mathbb{Z}^+$ vehicles deployed for exploration, we seek a plan to minimize the overall time between when the first vehicle starts exploring and when the last vehicle finishes exploring. Here, we assume all vehicles start exploring simultaneously; thus, it is equivalent to a minimax problem that minimizes the longest path among all vehicles.

The derivation of $\overline{\mathcal{S}}$ considers the vehicle's traversable space and viewpoint coverage, which encode the vehicle's mobility, endurance, and sensing capability. To account for the heterogeneity in such aspects, we use $\overline{\mathcal{S}}^i$ to denote the uncovered surfaces observable by vehicle i, $i \in \{1, ..., N\}$. The heterogeneity of vehicles is reflected in the area size of $\overline{\mathcal{S}}^i$ such that a vehicle with more capable mobility, longer endurance, and a larger sensing field of view can in general result in a larger area size. Denote $\overline{\mathcal{S}} = \cup_i \overline{\mathcal{S}}^i, i \in \{1, \ldots, N\}$ and denote $l(\mathcal{T})$ the length of path $\mathcal{T}$. We can extend Problem 1 to incorporate N vehicles with their current sensor poses $\{\mathbf{v}_{\text{current}}{}^1, ..., \mathbf{v}_{\text{current}}{}^N\}$ as follows:

Problem 2: Given $\overline{\mathcal{S}}^i, i \in \{1, \ldots, N\}$, and $\{\mathbf{v}_{\text{current}}{}^1, ..., \mathbf{v}_{\text{current}}{}^N\}$, find a set of paths $\{\mathcal{T}^{*1}, \ldots, \mathcal{T}^{*N}\}$ formed by viewpoints $\{\{\mathbf{v}_1{}^1, \mathbf{v}_2{}^1, ...\}, ..., \{\mathbf{v}_1{}^N, \mathbf{v}_2{}^N, ...\}\}$, which, when followed by the N vehicles, cover $\overline{\mathcal{S}}$, such that for all $i \in \{1, ..., N\}$, $\mathbf{v}_{\text{current}}{}^i \in \mathcal{T}^{*i}$, $\mathcal{T}^{*i}$is kinodynamically feasible, and $\max\{l(\mathcal{T}^{*1}), \ldots, l(\mathcal{T}^{*N})\}$ is minimized.

Local planning—Viewpoint sampling

We here define the criteria for a surface point to be covered by the sensor. Consider a surface patch centered at $\mathbf{p}_s \in \mathcal{W}$ with normal $\mathbf{n}_s \in \mathbb{R}^3$ pointing toward the free-space side; the center point on the surface patch is covered by viewpoint $\mathbf{v}$, if

$$| \mathbf{p}_s - \mathbf{p}_\mathbf{v} | \leq D \tag{2}$$

$$\frac{\mathbf{n}_s \cdot (\mathbf{p}_\mathbf{v} - \mathbf{p}_s)}{| \mathbf{n}_s || \mathbf{p}_\mathbf{v} - \mathbf{p}_s |} \geq T \tag{3}$$

where D and T are two constants constraining the relative distance and orientation of the surface patch with respect to the viewpoint. Such criteria ensure the surfaces to be perceived well. In practice, D is set to be shorter than the physical sensor range.

Define $\mathcal{H} \subset \mathcal{W}$ as the local planning horizon as shown in fig. S2A. Let $\mathcal{H}_{\text{trav}} \subset \mathcal{H}$ be the traversable subspace identified by considering collision and connectivity, and let $\mathcal{C}^{\mathcal{H}}_{\text{trav}}$ be the corresponding configuration space considering rotation and translation. Define $\overline{\mathcal{S}}_{\mathcal{H}} \subset \overline{\mathcal{S}}$ as the uncovered surfaces that can be perceived from viewpoints in $\mathcal{C}^{\mathcal{H}}_{\text{trav}}$. The problem of viewpoint sampling is equivalent to a set cover problem (*55*), which seeks a minimum set of viewpoints in $\mathcal{C}^{\mathcal{H}}_{\text{trav}}$ to cover $\overline{\mathcal{S}}_{\mathcal{H}}$. Let $\overline{\mathcal{S}}_\mathbf{v} \subset \overline{\mathcal{S}}_{\mathcal{H}}$ denote the uncovered surfaces to be perceived from $\mathbf{v} \in \mathcal{C}^{\mathcal{H}}_{\text{trav}}$. The reward of $\mathbf{v}$ is defined as the area of $\overline{\mathcal{S}}_\mathbf{v}$, denoted as $A_\mathbf{v}$. The viewpoint sampling problem exhibits submodularity (*56*); specifically, with more viewpoints selected, the reward of selecting an additional viewpoint decreases. This is because nearby viewpoints have overlapping field of views, and the same surface can be perceived from multiple viewpoints. Consequently, the reward of a viewpoint is dependent on the viewpoints selected earlier. Let $\mathbf{v}_i$, $i \in \mathbb{Z}^+$, be the ith viewpoint selected.

The uncovered surfaces $\overline{\mathscr{S}}_{\mathbf{v}_i}$ to be perceived from $\mathbf{v}_i$ need to be adjusted to $\overline{\mathscr{S}}_{\mathbf{v}_i} \setminus \cup_{j=1}^{i-1} (\overline{\mathscr{S}}_{\mathbf{v}_i} \cap \overline{\mathscr{S}}_{\mathbf{v}_j})$. Then, $A_{\mathbf{v}_i}$, the surface area of $\overline{\mathscr{S}}_{\mathbf{v}_i}$, is adjusted accordingly. Figure S2B gives an illustration of the local planning process from one of the real-world runs. The detailed algorithm that samples viewpoints for a complete coverage of the local area is presented in the Supplementary Materials (Algorithms 1 and 2).

Global planning—Space subdivision

We divided the space outside $\mathscr{H}$ into even cuboid subspaces. Each subspace stores the covered and uncovered surfaces developed during the exploration. The data are kept in the subspaces only for storage, whereas the data in $\mathscr{H}$ are updated as the exploration proceeds. Each subspace holds a status from "unexplored," "exploring," and "explored." If a subspace does not contain any covered or uncovered surfaces, the status is unexplored. If a subspace contains only covered surfaces, the status is explored. If a subspace contains any uncovered surfaces, the status is exploring. We only consider the exploring subspaces in global planning. Denote $\mathscr{G}_h \subset \mathscr{W}$, $h \in \mathbb{Z}^+$ as an exploring subspace and $\hat{\mathscr{G}}$ as the set of exploring subspaces. The global planning problem is to find a global path $\mathscr{T}_{\text{global}}$ that goes through the current viewpoint $\mathbf{v}_{\text{current}}$ and the centroid of each subspace in $\hat{\mathscr{G}}$. An illustration of the global exploring subspaces during a real-world exploration run is shown in fig. S2B.

During the exploration, we constructed a global roadmap in the traversable space along the vehicle's past trajectory, as shown in fig. S2C. The global roadmap is a graph where nodes represent physical traversable positions in the environment. A pair of nodes are connected by an edge if a traversable path exists between them. To avoid redundant computation for collision checking and path planning, the viewpoint candidates obtained by solving the aforementioned viewpoint sampling problem are reused as nodes in the graph. Recall that viewpoint candidates are uniformly generated in $\mathscr{H}_{\text{trav}}$ with collision and connectivity taken into consideration. In other words, each viewpoint candidate is collision-free and is connected with any other viewpoint candidates in $\mathscr{H}_{\text{trav}}$ with a traversable path. A subset of viewpoint candidates is reused as nodes in the global roadmap to avoid redundant collision and connectivity checking. To ensure sparsity of the roadmap, the viewpoint candidates are randomly sampled at a fixed resolution to maintain a sufficiently connected roadmap without incurring high computational cost. The detailed implementation of the roadmap can be found in our open-source code. Other data structures or route planning algorithms can be used in place of the global roadmap so long as they are probabilistically complete in computing shortest paths between two locations. We used the A* algorithm to search for shortest paths between two subspaces on the global roadmap.

To maintain conciseness, the global planning method for a single robot is provided in the Supplementary Materials (Algorithm 3). In addition, we encourage readers to refer to our prior works (*41*, *42*) for more detailed information.

Global planning—Multirobot exploration

We assume that all robots are initialized in the same coordinate frame and have prior knowledge of other robots' starting positions. For coordination, robots' current positions, the status of exploring and explored subspaces, and their traversability are shared among all robots. Only new information is shared each time to avoid redundant transmission.

With a synchronized global understanding, each robot uses a procedure similar to Algorithm 3 to compute the exploration path. Instead of solving a TSP, each robot solves a VRP to compute a set of global paths $\{\mathscr{T}_{\text{global}}{}^i\}, i \in \{1, \ldots N\}$ for all N robots to visit all exploring subspaces. The ith robot then takes $\mathscr{T}^i_{\text{global}}$ as its global path for execution. Here, the optimal solution to the VRP is the one such that $\max\{l(\mathscr{T}_{\text{global}}{}^i)\}$ is minimized. The computation of solving the VRP is distributed among robots, where the lowest-cost solution is shared among all robots and used as the initial guess to be optimized further by each robot.

Global planning—Multirobot exploration with limited communication

Our dual-resolution scheme was also adapted to address the limited communication constraint in multirobot exploration. Here, we assume that robots can communicate when they are physically within a distance range. In addition, robots can communicate over multiple hops, where the information is relayed by other robots in between.

The conventional rendezvous-based strategies

During a rendezvous, robots meet each other at a fixed time cadence and at a predefined location that all robots agreed upon previously. The rendezvous behavior is realized at the global level of our method. With a synchronized knowledge about the exploring subspaces $\hat{\mathscr{G}}$, robots agree upon the next rendezvous location and time. In particular, the centroid of an exploring global subspace $\mathbf{g}_{\text{rdv}} \in \hat{\mathscr{G}}$ is designated as the rendezvous location. $\mathbf{g}_{\text{rdv}}$ is selected as the one with the least farthest-neighbor distance, given by

$$\mathbf{g}_{\text{rdv}} = \text{argmin}_{\mathscr{G}_h \in \hat{\mathscr{G}}} \max_{\mathscr{G}_l \in \hat{\mathscr{G}}} d(\mathscr{G}_h, \mathscr{G}_l), h, l \in \mathbb{Z}^+, h \neq l \quad (4)$$

where $d(\mathscr{G}_h, \mathscr{G}_l)$ gives the length of the shortest path between two global subspaces. By doing so, robots can take similar traveling time to the rendezvous location from other exploring subspaces. The next rendezvous time is set dynamically on the basis of the traveling time from the current rendezvous location to the farthest exploring subspace in $\hat{\mathscr{G}}$. For timely arrival at $\mathbf{g}_{\text{rdv}}$, we solved a VRP with a time window constraint (*46*) associated with $\mathbf{g}_{\text{rdv}}$ for global paths. Furthermore, each robot estimates the traveling time to $\mathbf{g}_{\text{rdv}}$ to determine when to stop exploring and start going to $\mathbf{g}_{\text{rdv}}$.

We also implemented a multipoint rendezvous strategy, where robots gather at different locations to shorten the traveling distance. In such a strategy, each robot continues to travel to $\mathbf{g}_{\text{rdv}}$ until it is connected with other robots, where one of them is already at $\mathbf{g}_{\text{rdv}}$. In this manner, robots that arrive at $\mathbf{g}_{\text{rdv}}$ earlier form a growing communication network, where the ones arriving later only need to reach the boundary of the network to communicate.

The proposed pursuit strategy

In our pursuit strategy, robots opportunistically go after each other to communicate to improve the overall exploration time efficiency. To evaluate whether a communication attempt is beneficial, a robot reasons about how its own information can affect other robots' exploration plans and the cost of delivering such information.

We define a robot's knowledge of the environment as the set of explored and exploring global subspaces denoted by $\tilde{\mathscr{G}}$. Each robot keeps track of its own knowledge and the knowledge of other robots.

To represent what robot i believes about robot j's knowledge of the environment, we use the notation $\tilde{\mathcal{G}}_i^j$, where i and j are integers between 1 and N. The notation $\tilde{\mathcal{G}}_i^i$ represents robot i's own knowledge. Before the exploration, $\tilde{\mathcal{G}}_i^j$ is initialized with the prior knowledge that robot i has about robot j, for instance, a single global subspace that contains robot j's starting position. When two robots, i and j, can communicate with each other, they synchronize their knowledge with the updated knowledge denoted by superscript †. The synchronization is represented by the equation

$$\tilde{\mathcal{G}}_i^{i\dagger} = \tilde{\mathcal{G}}_j^{j\dagger} = \tilde{\mathcal{G}}_i^{j\dagger} = \tilde{\mathcal{G}}_j^{i\dagger} \leftarrow \tilde{\mathcal{G}}_i^i \oplus \tilde{\mathcal{G}}_j^j \tag{5}$$

Furthermore, they update their understanding of other robots' knowledge such that

$$\tilde{\mathcal{G}}_i^{k\dagger} = \tilde{\mathcal{G}}_j^{k\dagger} \leftarrow \tilde{\mathcal{G}}_i^k \oplus \tilde{\mathcal{G}}_j^k \tag{6}$$

where k is an integer between 1 and N, and k is not equal to i and j. In this context, $\oplus$ refers to the combination of two robots' knowledge, and the explored status has precedence over the exploring status when updating the status of a global subspace.

In the scenario where robot i and robot j are out of the communication range, robot i must decide whether to pursue robot j to communicate by evaluating two options, no communication and assumed communication. Let $\hat{\mathcal{G}}_{\text{new}}$ denote the global subspaces that are currently only known to robot i, such that $\hat{\mathcal{G}}_{\text{new}} \not\subset \tilde{\mathcal{G}}_i^j$. In the scenario of no communication, robot i plans the global path $\{\mathcal{T}_{\text{global}}{}^1, \ldots, \mathcal{T}_{\text{global}}{}^N\}$ with the cost

$$c = \max\{l(\mathcal{T}_{\text{global}}{}^1), \ldots, l(\mathcal{T}_{\text{global}}{}^N)\} \tag{7}$$

where $\mathcal{T}_{\text{global}}{}^j$ does not visit $\hat{\tilde{\mathcal{G}}}_{\text{new}}$ because robot i believes that robot j is not aware of it. In the scenario of assumed communication, assuming robot j knows about $\hat{\mathcal{G}}_{\text{new}}$, such that

$$\tilde{\mathcal{G}}_i^{j\dagger} \leftarrow \tilde{\mathcal{G}}_i^j \oplus \hat{\mathcal{G}}_{\text{new}} \tag{8}$$

robot i can plan the global paths as $\{\mathcal{T}^\dagger_{\text{global}}{}^1, \ldots, \mathcal{T}^\dagger_{\text{global}}{}^N\}$ with the cost

$$c^\dagger = \max\{l(\mathcal{T}^\dagger_{\text{global}}{}^1), \ldots, l(\mathcal{T}^\dagger_{\text{global}}{}^N)\} \tag{9}$$

where $\mathcal{T}^\dagger_{\text{global}}{}^j$ may visit exploring subspaces in $\hat{\mathcal{G}}_{\text{new}}$. If $c^\dagger$ is less than c, it is beneficial for robot i to communicate with robot j so that robot j can share the workload of visiting $\hat{\mathcal{G}}_{\text{new}}$. However, this requires robot i to deviate from its current exploration path, thereby incurring additional cost. In the following, we examine the cost incurred by robot i while pursuing robot j and evaluate the situations where the incurred cost can be justified by the potential benefits.

We proceed to estimate the cost incurred by robot i in pursuing robot j to share the information about $\hat{\mathcal{G}}_{\text{new}}$. Assuming that robot j follows the global path $\mathcal{T}_{\text{global}}{}^j$ to explore the subspaces in $\hat{\mathcal{G}}_{\mathcal{T}_{\text{global}}{}^j}$, robot i can adopt a pursuit strategy whereby it visits the same subspaces to locate robot j. By estimating the arrival time of robot j at each subspace, robot i can plan a path that maximizes the likelihood of encountering robot j with minimal travel time. This can be approximated by solving a TSP problem subject to time window constraints (*57*), which computes the global path for robot i to visit $\hat{\mathcal{G}}_{\mathcal{T}_{\text{global}}{}^j}$ from its current position and eventually returns to the same position. We denote the path taken by robot i in pursuing robot j as $\mathcal{T}_{\text{comms}}{}^i$. The overall cost taking into account this pursuit is given by

$$\begin{aligned} c^\dagger_{\text{comms}} = \max\{&l(\mathcal{T}^\dagger_{\text{global}}{}^1), \ldots, l(\mathcal{T}^\dagger_{\text{global}}{}^i) \\ &+ l(\mathcal{T}_{\text{comms}}{}^i), \ldots, l(\mathcal{T}^\dagger_{\text{global}}{}^N)\} \end{aligned} \tag{10}$$

assuming successful exchange of information between the two robots. If $c^\dagger_{\text{comms}}$ is less than the current cost c, robot i should interrupt its exploration and pursue robot j by following $\mathcal{T}_{\text{comms}}{}^i$. Otherwise, it should continue exploring. After completing the exploration of known areas, a robot should pursue other robots to acquire additional information about the environment or to relay information about previously explored spaces. The process for determining whether to pursue a certain robot and where to pursue it remains the same as described above, with the primary objective of minimizing the overall exploration time. However, in this scenario, the focus of information exchange shifts toward the potential discovery of new areas to explore or avoiding redundant visits to already explored spaces rather than sharing the workload of exploring a large space as in typical cases. An illustration of the strategy is presented in fig. S2 (D and E).

In general, when there are $M (M \in \mathbb{Z}^+, M < N)$ robots that are within communication range of a total of N robots, we use a random sampling method to select communication targets iteratively from the remaining $N - M$ robots. In particular, there are $2^{(N-M)}$ possible combinations of the target robots to communicate. We distributed the selection of target robots over the planning cycles, where only a smaller number of combinations are selected without replacement from the total $2^{(N-M)}$ combinations at each planning cycle. Empirically, we found that targeting a smaller number of robots (one or two) is sufficient for improving the noncommunicative strategy. Therefore, we biased the selection to prioritize fewer targets, where the likelihood of selecting combinations with fewer targets is higher than those with more targets. To determine the routes for the M robots to pursue the selected target robots, we solved a VRP with time window constraints (*46*). Similar to the scheme discussed earlier, the VRP solution for pursuing is jointly optimized and shared among all M robots. If the pursuit attempt fails, we use a rendezvous-like strategy as a fallback, where bots return to a predetermined location to meet each other, which is set in the same way as choosing $\mathbf{g}_{\text{rdv}}$ for the rendezvous-based strategy. Essentially, the pursuit strategy degenerates to the rendezvous-based strategy in the worst case. However, such cases are rare because of the symmetry in the robots' reasoning. A target robot that has found sufficient new information will also try to pursue, which tends to meet the pursuing robots midway. Otherwise, the target robots will not be far from their original exploration paths and can be easily found by the pursuing robot. The procedure for computing the multirobot exploration path with limited communication is presented in the Supplementary Materials (Algorithm 4).

Supplementary Materials

This PDF file includes:
Figs. S1 to S7
Tables S1 to S7
References (*58*–*71*)

Other Supplementary Material for this manuscript includes the following:
Movies S1 to S4

REFERENCES AND NOTES

1. B. Grocholsky, J. Keller, V. Kumar, G. Pappas, Cooperative air and ground surveillance. *IEEE Robot Autom Mag.* **13**, 16–25 (2006).
2. Y. Liu, G. Nejat, Robotic urban search and rescue: A survey from the control perspective. *J. Intell. Robot. Syst.* **72**, 147–165 (2013).
3. E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, D. Hull, Morse decompositions for coverage tasks. *Int. J. Rob. Res.* **21**, 331–344 (2002).
4. B. J. Englot, F. S. Hover, Sampling-based coverage path planning for inspection of complex structures, in *International Conference on Automated Planning and Scheduling* (AAAI, 2012), pp. 29–37.
5. DARPA, DARPA Subterranean Challenge (2021); www.subtchallenge.com/.
6. A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, R. Siegwart, Receding horizon" next-best-view" planner for 3D exploration, in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016), pp. 1462–1468.
7. T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, M. Hutter, Graph-based subterranean exploration path planning using aerial and legged robots. *J. Field Robot.* **37**, 1363–1388 (2020).
8. M. Dharmadhikari, T. Dang, L. Solanka, J. Loje, H. Nguyen, N. Khedekar, K. Alexis, Motion primitives-based path planning for fast and agile exploration using aerial robots, in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), pp. 179–185.
9. Z. Xu, D. Deng, K. Shimada, Autonomous UAV exploration of dynamic environments via incremental sampling and probabilistic roadmap. *IEEE Robot. Autom. Lett.* **6**, 2729–2736 (2021).
10. M. W. M. Gamini Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, M. Csorba, A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Autom.* **17**, 229–241 (2001).
11. S. Thrun, W. Burgard, D. Fox, A probabilistic approach to concurrent mapping and localization for mobile robots. *Mach. Learn. Auton Robots.* **5**, 1–25 (1998).
12. D. Fox, W. Burgard, F. Dellaert, S. Thrun, Monte Carlo Localization: Efficient position estimation for mobile robots, in *Proceedings of the National Conference on Artificial Intelligence* (AAAI, 1999), pp. 343–349.
13. H. Choset, J. Burdick, Sensor-based exploration: The hierarchical generalized Voronoi graph. *Int. J. Robot. Res.* **19**, 96–125 (2000).
14. J. Kim, F. Zhang, M. Egerstedt, A provably complete exploration strategy by constructing Voronoi diagrams. *Auton. Robots.* **29**, 367–380 (2010).
15. K. Masaba, A. Q. Li, GVGExp: Communication-constrained multi-robot exploration system based on generalized Voronoi graphs, in *International Symposium on Multi-Robot and Multi-Agent Systems* (IEEE, 2021), pp. 146–154.
16. P. Beeson, J. Modayil, B. Kuipers, Factoring the mapping problem: Mobile robot map-building in the hybrid spatial semantic hierarchy. *Int. J. Robot. Res.* **29**, 428–459 (2010).
17. D. Wilkes, E. Milios, Robotic exploration as graph construction. *IEEE Trans. Robot. Autom.* **7**, 859–865 (1991).
18. F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, H. F. Durrant-Whyte, Information based adaptive robotic exploration, paper presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Lausanne, Switzerland, 30 September to 4 October 2002.
19. C. Stachniss, G. Grisetti, W. Burgard, Information gain-based exploration using Rao-blackwellized particle filters, in *Robotics: Science and Systems* (MIT Press, 2005), pp. 1–10.
20. F. Amigoni, V. Caglioti, An information-based exploration strategy for environment mapping with mobile robots. *Robot. Auton. Syst.* **58**, 684–699 (2010).
21. W. Tabib, M. Corah, N. Michael, R. Whittaker, Computationally efficient information-theoretic exploration of pits and caves, paper presented at the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 09 to 14 October 2016.
22. S. Bai, J. Wang, F. Chen, B. Englot, Information-theoretic exploration with Bayesian optimization, paper presented at the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9 to 14 October 2016.
23. D. Lee, M. Recce, Quantitative evaluation of the exploration strategies of a mobile robot. *Int. J. Robot. Res.* **16**, 413–447 (1997).
24. F. Amigoni, Experimental evaluation of some exploration strategies for mobile robots, in *2008 IEEE International Conference on Robotics and Automation* (IEEE, 2008), pp. 2818–2823.
25. D. Holz, N. Basilico, F. Amigoni, S. Behnke, Evaluating the efficiency of frontier-based exploration strategies, paper presented at the 41st International Symposium on Robotics (ISR) and 6th German Conference on Robotics (ROBOTIK), Munich, Germany, 7 to 9 June 2010.
26. F. Amigoni, J. Banfi, N. Basilico, Multirobot exploration of communication-restricted environments: A survey. *IEEE Intell. Syst.* **32**, 48–57 (2017).
27. J. Gielis, A. Shankar, A. Prorok, A critical review of communications in multi-robot systems. *Curr. Robot. Rep.* **3**, 213–225 (2022).
28. M. Meghjani, G. Dudek, Combining multi-robot exploration and rendezvous, in *Canadian Conference on Computer and Robot Vision* (IEEE, 2011), pp. 80–85.
29. B. L. Wellman, S. Dawson, J. de Hoog, M. Anderson, Using rendezvous to overcome communication limitations in multirobot exploration, in *2011 IEEE International Conference on Systems, Man, and Cybernetics* (IEEE, 2011), pp. 2401–2406.
30. N. Roy, G. Dudek, Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. *Auton. Robots* **11**, 117–136 (2001).
31. M. Meghjani, G. Dudek, Multi-robot exploration and rendezvous on graphs, in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2012), pp. 5270–5276.
32. J. de Hoog, S. Cameron, A. Visser, Autonomous multi-robot exploration in communication-limited environments, in *Conference Towards Autonomous Robotic Systems* (UKAEA, 2010), pp. 68–75.
33. N. Goddemeier, K. Daniel, C. Wietfeld, Role-based connectivity management with realistic air-to-ground channels for cooperative UAVs. *IEEE J. Sel. Areas Commun.* **30**, 951–963 (2012).
34. M. N. Rooker, A. Birk, Multi-robot exploration under the constraints of wireless networking. *Control Eng. Pract.* **15**, 435–445 (2007).
35. R. Rathnam, A. Birk, Distributed communicative exploration under underwater communication constraints, in *9th IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (IEEE, 2011), pp. 339–344.
36. M. Corah, C. O'Meadhra, K. Goel, N. Michael, Communication-efficient planning and mapping for multi-robot exploration in large environments. *IEEE Robot. Autom. Lett.* **4**, 1715–1721 (2019).
37. Z. Zhang, J. Yu, J. Tang, Y. Xu, Y. Wang, MR-TopoMap: Multi-robot exploration based on topological map in communication restricted environment. *IEEE Robot. Autom. Lett.* **7**, 10794–10801 (2022).
38. G. Best, R. Garg, J. Keller, G. A. Hollinger, S. Scherer, Resilient multi-sensor exploration of multifarious environments with a team of aerial robots, paper presented at the Robotics: Science and Systems Conference (RSS), New York City, NY, 27 June to 1 July 2022.
39. Y. Pei, M. W. Mutka, Steiner traveler: Relay deployment for remote sensing in heterogeneous multi-robot exploration, in *2012 IEEE International Conference on Robotics and Automation* (IEEE, 2012), pp. 1551–1556.
40. K. Otsu, S. Tepsuporn, R. Thakker, T. S. Vaquero, J. A. Edlund, W. Walsh, G. Miles, T. Heywood, M. T. Wolf, A. A. Agha-Mohammadi, Supervised autonomy for communication-degraded subterranean exploration by a robot team, in *2020 IEEE Aerospace Conference Proceedings* (IEEE, 2020), pp. 1–9.
41. C. Cao, H. Zhu, H. Choset, J. Zhang, Exploring large and complex environments fast and efficiently, paper presented at the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May 2021 to 5 June 2021.
42. C. Cao, H. Zhu, H. Choset, J. Zhang, TARE: A hierarchical framework for efficiently exploring complex 3D environments, paper presented at the Robotics: Science and Systems Conference (RSS), Virtual, 12 to 16 July 2021.
43. J. O'Rourke, *Art Gallery Theorems and Algorithms* (Oxford Univ. Press, 1987), vol. 57, p. 282.
44. C. H. Papadimitriou, The complexity of the Lin–Kernighan heuristic for the traveling salesman problem. *SIAM J. Comput.* **21**, 450–465 (1992).
45. G. B. Dantzig, J. H. Ramser, The truck dispatching problem. *Manag. Sci.* **6**, 80–91 (1959).
46. J. C. Molina, J. L. Salmeron, I. Eguia, J. Racero, The heterogeneous vehicle routing problem with time windows and a limited number of resources. *Eng. Appl. Artif. Intel.* **94**, 103745 (2020).
47. E. Gat, On three-layer architectures, in *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, D. Kortenkamp, R. P. Bonnasso, R. Murphy, Eds. (MIT Press, 1997), pp. 195–210.
48. R. Firby, *Adaptive Execution in Complex Dynamic Worlds* (Yale Univ., 1989).
49. N. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2004), pp. 2149–2154.

50. J. Zhang, S. Singh, Laser-visual-inertial odometry and mapping with high robustness and low drift. *J. Field Robot.* **35**, 1242–1264 (2018).
51. W. Tabib, K. Goel, J. Yao, M. Dabhi, C. Boirum, N. Michael, Real-time information-theoretic exploration with Gaussian mixture model maps, paper presented at the Robotics: Science and Systems Conference (RSS), Messe Freiburg, Germany, 22 June to 26 June 2019.
52. S. M. LaValle, Rapidly-exploring random trees: A new tool for path planning (Tech. Rep. 98-11, Iowa State Univ., 1998).
53. S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.*. **30**, 846–894 (2011).
54. S. Scherer, V. Agrawal, G. Best, C. Cao, K. Cujic, R. Darnley, R. DeBortoli, E. Dexheimer, B. Drozd, R. Garg, I. Higgins, J. Keller, D. Kohanbash, L. Nogueira, R. Pradhan, M. Tatum, V. K. Viswanathan, S. Willits, S. Zhao, H. Zhu, D. Abad, T. Angert, G. Armstrong, R. Boirum, A. Dongare, M. Dworman, S. Hu, J. Jaekel, R. Ji, A. Lai, Y. H. Lee, A. Luong, J. Mangelson, J. Maier, J. Picard, K. Pluckter, A. Saba, M. Saroya, E. Scheide, N. Shoemaker-Trejo, J. Spisak, J. Teza, F. Yang, A. Wilson, H. Zhang, H. Choset, M. Kaess, A. Rowe, S. Singh, J. Zhang, G. A. Hollinger, M. Travers, Resilient and modular subterranean exploration with a team of roving and flying robots. *Field Robot.* **2**, 678–734 (2022).
55. T. Grossman, A. Wool, Computational experience with approximation algorithms for the set covering problem. *Eur. J. Oper. Res.* **101**, 81–92 (1997).
56. M. Roberts, D. Dey, A. Truong, S. Sinha, S. Shah, A. Kapoor, P. Hanrahan, N. Joshi, Submodular trajectory optimization for aerial 3D scanning, in *IEEE International Conference on Computer Vision* (IEEE, 2017), pp. 5334–5343.
57. Y. Dumas, J. Desrosiers, E. Gelinas, M. M. Solomon, An optimal algorithm for the traveling salesman problem with time windows. *Oper. Res.* **43**, 367–371 (1995).
58. D. Li, X. Sun, *Nonlinear Integer Programming* (Springer, 2006), vol. 84.
59. D. Bergman, A. A. Cire, W.-J. van Hoeve, T. Yunes, BDD-based heuristics for binary optimization. *J. Heuristics* **20**, 211–234 (2014).
60. S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach* (Pearson Education Inc., 2010).
61. B. Zhou, F. Gao, L. Wang, C. Liu, S. Shen, Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robot. Autom. Lett.* **4**, 3529–3536 (2019).
62. S. G. Johnson, The NLopt nonlinear-optimization package (2014); http://github.com/stevengj/nlopt.
63. T. Schouten, E. L. van den Broek, Incremental distance transforms (IDT), paper presented at the International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010.
64. S. Quinlan, O. Khatib, Elastic bands: Connecting path planning and control, in *Proceedings of the IEEE International Conference on Robotics and Automation* (IEEE, 1993), pp. 802–807.
65. K. Helsgaun, An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems (Roskilde Univ., 2017).
66. E. M. Arkin, R. Hassin, Approximation algorithms for the geometric covering salesman problem. *Discrete Appl. Math.* **55**, 197–218 (1994).
67. A. Dumitrescu, J. S. B. Mitchell, Approximation algorithms for TSP with neighborhoods in the plane. *J. Algorithms* **48**, 135–159 (2003).
68. F. Ferri, M. Gianni, M. Menna, F. Pirri, Dynamic obstacles detection and 3D map updating, in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2015), pp. 5694–5699.
69. V. Indelman, A. Melim, F. Dellaert, Incremental light bundle adjustment for robotics navigation, in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2013), pp. 1952–1959.
70. A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, Y. Zhang, Matterport3D: Learning from RGB-D data in indoor environments, paper presented at the International Conference on 3D Vision (3DV), Qingdao, China, 10 to 12 October 2017.
71. J. Zhang, C. Hu, R. G. Chadha, S. Singh, Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation. *J. Field Robot.* **37**, 1300–1313 (2020).

Acknowledgments

Funding: The funding was provided by the National Robotics Engineering Center (NREC) at Carnegie Mellon University. **Author contributions:** Conceptualization: C.C., J.Z., and H.C. Methodology: C.C. and J.Z. Investigation: C.C., H.Z., and Z.R. Visualization: C.C., H.Z., and J.Z. Funding acquisition: J.Z. Project administration: J.Z. Supervision: J.Z. and H.C. Writing—original draft: C.C., H.Z., and J.Z. Writing—review and editing: C.C., H.Z., Z.R., J.Z., and H.C. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** The code for the autonomous exploration algorithm is available at https://github.com/caochao39/tare_planner. The software stack for the development/benchmarking environment and environment models is available at https://cmu-exploration.com.

Submitted 29 October 2022
Accepted 21 June 2023
Published 19 July 2023
10.1126/scirobotics.adf0970

Science Robotics

Representation granularity enables time-efficient autonomous exploration in large, complex worlds

C. Cao, H. Zhu, Z. Ren, H. Choset, and J. Zhang

Sci. Robot., **8** (80), eadf0970.
DOI: 10.1126/scirobotics.adf0970