Autonomous Multi-robot Exploration with Ground Vehicles in DARPA Subterranean Challenge Finals

Jan Bayer* **Petr Čížek*** **Jan Faigl***

Abstract

Autonomous navigation and multi-robot exploration framework for ground robots are key parts of the robotic system deployed by the team CTU-CRAS-NORLAB in the final event of the Subterranean (SubT) Challenge organized by the Defense Advanced Research Projects Agency (DARPA) in 2021. The SubT Challenge aimed to advance technologies related to search-and-rescue missions with multi-robot systems in underground environments where communication is unavailable and global navigation satellite systems are denied. This field report describes the developed multi-robot exploration framework focusing on planning, exploration, and traversability estimation for a heterogeneous team of ground vehicles in large-scale rough terrains and multi-robot coordination with limited communication. The developed method employs a dense local mapping for precise traversability estimation combined with a sparse topometrical map shareable between multiple robots and is thus used in the decision-making of the exploration strategy. The topometrical map is designed to support the decentralized coordination of heterogeneous teams of robots, which is demonstrated by deploying the developed framework in the SubT competitions. The framework has been employed in the Virtual track for the full autonomous control of the ground robots, where our team scored second. Besides, in the Systems track, a human supervisor exploited autonomous behaviors provided by the proposed framework to control a heterogeneous team of six ground robots. We report on real-world experimental results from the deployment of the SubT Challenge. Furthermore, we present results from the post-event testing of the SubT Challenge, where three quadruped robots controlled by the framework explored over five hundred meters fully autonomously.

1 Introduction

In the presented text, we report on the deployment results of the developed autonomous navigation and multi-robot exploration framework targeted to large-scale environments where global navigation satellite systems are denied, and communication infrastructure is not a priory available. The presented work is motivated by the search-and-rescue missions (Amigoni et al., 2017), where knowledge of the geometrical model of the area, such as collapsed buildings and positions of the objects of interest, provides a significant advantage to first responders and may speed up the rescue operation or avoid exposing humans to danger. The development of underground search and rescue operations has been accelerated by the Subterranean (SubT) Challenge (Chung, 2022), organized by Defense Advanced Research Projects Agency (DARPA). In the DARPA SubT Challenge, the task is to find, within a one-hour time limit, defined artifacts in an underground environment that is unknown before the deployment of the robots. The environment can be entered only by robots, and only a single human supervisor outside the mission area is allowed to remotely connect to

*Authors are with the Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic. The corresponding author's e-mail: bayerja1@fel.cvut.cz.

the robots through a dedicated computer called the base station. Since there is only a single supervisor for all the robots, autonomous behaviors support the supervisor's efficiency in managing multiple robots simultaneously. Besides qualitative requirements on each component of the system, the SubT Challenge also puts high demands on the reliability and fidelity of each part of the system and the whole system together, which is a key factor in applying autonomy in search-and-rescue missions (Delmerico et al., 2019).

The autonomous behavior of the robots in conditions defined by the DARPA SubT Challenge puts high requirements on many robot capabilities. In the presented report, we focus on the parts of the multi-robot exploration system used by the CTU-CRAS-NORLAB team of the Czech Technical University in Prague and Laval University participating in the DARPA SubT Challenge. More specifically, we focus on the essential parts of autonomous behaviors, including planning for navigation in rough unstructured terrains and the generation of long-distance plans allowing for prompt exploration decisions. The developed exploration framework enables exploration using a team of heterogeneous ground mobile robots autonomously coordinated in environments with limited communication. In particular, the employed exploration strategy is based on the decentralized method for multi-robot exploration using communication bandwidth limited to broadcasting less than $100\,\mathrm{B\,s^{-1}}$. The provided system's description includes details of the related system's parts that directly influence the coordination of the robots and planning. The considered contributions of the presented work are as follows.

- A method of aligning the coordinate frames of multiple robots using a total station, which is important for missions in a large-scale environment.
- Traversability estimation of 3D unstructured environments allowing safe navigation in narrow corridors and rough terrains supported by changing locomotion gaits (for multi-legged robots) when approaching rough terrain.
- An exploration planning method based on combining dense local maps and a global topometric environment model to support efficient navigation within the robot's surroundings and fast calculation of long paths required in deciding where to explore next.
- Communication protocol and decentralized coordination of heterogeneous robots using two independent communication systems while requiring only low communication bandwidth to coordinate robots in exploration missions of unknown environments.

The report is organized as follows. Related multi-robot approaches and robotic competitions are briefly summarized in Section 2, together with an overview of approaches applied by other teams participating in the Systems track of the DARPA SubT Challenge. The developed exploration framework and its components for underground exploration missions are described in Section 3. Evaluation results of the framework achieved in both simulated and real-world deployments at the SubT Finals and post-event testing of fully autonomous exploration are reported in Section 4. Concluding remarks are in Section 5.

2 Related Work

The motivation for the presented work is the Search-and-Rescue (SAR) scenario (Delmerico et al., 2019; Murphy, 2014), where autonomous robots explore hazardous parts of the environment to provide critical information, a model of the environment, and locations of possible victims. The focus of the work is planning for autonomous multi-robot exploration in underground environments, where communication suffers from bandwidth limitation and communication range (Amigoni et al., 2017). Besides, we can also find exploration applications in regular data collection missions, such as inspection or mapping of underground infrastructure, utility tunnels, and sewers (Kolvenbach et al., 2019). Since the deployment of the developed solution is motivated by the DARPA SubT Challenge, we further provide an overview of related robotic competitions and a brief description of the solutions reported by other teams in the DARPA SubT Challenge.

2.1 Mobile Robot Exploration

Robotic exploration as a problem to create a map of the environment using a 3D mapping technique with scans collected by a wheeled robot is presented in an early underground mapping approach (Thrun et al., 2003). Autonomous exploration of underground caves is described in (Husain et al., 2013), where the authors generate navigation waypoints using the so-called frontier-based exploration. The concept of frontiers was introduced in (Yamauchi, 1997) for an occupancy grid model of the environment, where the grid cells are classified as free, unknown, or occupied. The frontier cells are identified as free cells neighboring unknown cells.

Several works further improved the determination of the exploration waypoints, such as (Amigoni and Caglioti, 2010; González-Banos and Latombe, 2002), to name a few, where the distance utility functions are employed to evaluate frontier cells that would most contribute to the exploration. In (Faigl and Kulich, 2013), the authors propose increasing the efficiency of frontier-based exploration by determination of the possible exploration waypoints from which the robot would cover multiple frontier cells. An overview of the further improvements can be found in the survey (Jain et al., 2017).

Deploying multiple robots in an exploration mission can increase resiliency and exploration performance, or multi-robot teams can benefit from the complementary properties of the robots (Heppner et al., 2013). Suppose the communication bandwidth and connectivity between the robots are not an issue. In that case, the robots can use their local environment models, such as occupancy grids, to build a global map that can be utilized to determine the next navigational waypoint for each robot using the frontier-based method (Yamauchi, 1998). Utility-based assessment of the next-to-visit locations evaluates the distance to the next waypoint location and the expected coverage of the not yet explored part of the environment (Burgard et al., 2000; Burgard et al., 2005). Distributed building of the occupancy map is presented in (Fox et al., 2006), where the robots are actively verifying their knowledge about other robots' locations to ensure that combining the occupancy maps provided by other robots is consistent. For multi-robot exploration, the selection of the next-to-visit locations can be considered as a variant of the task allocation problem that can be addressed by several assignment strategies (Faigl and Kulich, 2015), such as the Hungarian algorithm or as a formulation of the Multiple Traveling Salesmen Problem (MTSP) (Faigl et al., 2012).

In addition to direct evaluation of the navigational waypoints based on travel cost and expected area to be covered, ranking-based methods have been employed to support the coordination of multiple exploring units. The waypoint rank characterizes the suitability of the waypoint to be visited by a particular robot. Thus, each robot ranks the waypoints and then follows the waypoint with the best rank. Since the robots can rank the waypoints by themselves, the ranking-based coordination support decentralized decision-making, albeit the rank computation might still depend on information sharing. An example of ranking-based exploration is the MinPos method (Bautin et al., 2012), in which the rank is computed as the number of robots closer to the particular waypoint than the robot computing the rank. Hence, the robot selects the waypoint with the lowest rank as the next navigational waypoint. In (Benkrid et al., 2019), the authors rank exploration waypoints for each robot based on the energy required to reach the waypoints. However, both ranking methods require sharing the occupancy grid maps between the robots to compute the rank, which can be communication demanding.

Low-bandwidth communication is addressed in (Burgard et al., 2000) by sharing maps represented as sets of polygons that are more memory efficient than grid maps. In (Schulz et al., 2019), the occupancy Normal Distributions Transform (NDT) (Biber and Strasser, 2003) is used as the environment model representation with a low memory footprint, shareable via a network with a bandwidth of 37.5 kB/s. The authors report that at least 4.1 kB/s of the bandwidth has been used.

In our SubT deployment, we consider that even lower bandwidth would be available. Therefore, based on the overviewed methods, we propose the ranking-based strategy to be a suitable choice if the rank is computed from locally available information about the robot surroundings and sparsely sampled positions of the other robots shared via a low-bandwidth communication channel.

2.2 Related Robotic Competitions

Past robotic competitions accelerated improvements and consolidation of state-of-the-art methods to obtain frameworks with sufficient capabilities to solve problems designed by the competition organizers. In (Delmerico et al., 2019), the authors indicate that the principal problem of using novel methods in SAR missions is the deficient level of reliability, which is also pushed by robotic competitions, where the number of mission trials is limited. For example, the first robotic Grand Challenge organized by the DARPA in 2004 requested an autonomous uncrewed ground vehicle to travel along a 240 km long route through the Mojave Desert. The route has been described as a sequence of waypoints; thus, no global path planning was necessary. However, the off-road terrain required road-finding, obstacle detection, and obstacle avoidance by vehicles driving even over $60\,\mathrm{km\,h^{-1}}$. Unfortunately, none of the participating teams finished the route successfully, while a year later, a total of five teams completed the course of the second Grand Challenge of 212 km long route with the robot Stanley (Thrun et al., 2006) successfully winning the challenge, finishing the tour in 6 hours and 54 minutes. It marked significant progress in the reliability of the utilized systems and fostered further development in self-driving vehicles.

RoboCup Rescue League competition is an example of a robotic competition targeted at SAR scenarios in an urban environment. The competition's goal is to build a precise and preferably complete map of the environment and localize possible victims. During the competition, robots encounter uneven terrain and unstructured obstacles, which they must avoid while autonomously exploring the area. Competition details and methods of the successful teams of the 2006 competition are summarized in the field report (Balakirsky et al., 2007), where the authors claim that most of the teams used grid-based planning, simplistic frontier-based exploration (Yamauchi, 1997), or combinations of reactive behaviors.

The Multi Autonomous Ground-robotic International Challenge (MAGIC) was focused on multi-robot coordination and autonomy. The task of the competition was to explore the environment, provide its map, and "neutralize" detected threats, including hostile humans and simulated bombs. The environment was about $500\,\mathrm{m} \times 500\,\mathrm{m}$ large area with both indoor and outdoor environments, and each team was allowed to have two human supervisors. The team Michigan won first place in the competition (Olson et al., 2012). Their solution used centralized task allocation, and when a robot received a task, it executed the task using its terrain classification and path planning methods. The exploration waypoints were identified based on frontiers (Yamauchi, 1997). The robot used a trajectory graph to speed up planning when returning to explore distant areas. The work of two other successful teams is summarized in (Lacaze et al., 2012) and (Butzke et al., 2012). The second team also used the centralized approach, in which the supervisors may affect which areas should or should not be explored by a particular robot. The autonomous exploration by each robot was based on maximizing information gain computed for each exploration waypoint detected within a grid with a 20 cm cell size according to the frontier-based approach (Yamauchi, 1997). In contrast to the first and second teams, the third team (Butzke et al., 2012) used distributed coordination that can operate even without connecting to the supervisor control unit. The area is partitioned into smaller parts, with the exploration route planned by a solution of the corresponding instance of the MTSP.

DLR SpaceBot-Cup 2015 competition focused on developing robotic systems to perform tasks required in exploration scenarios on a planetary surface, including exploration of a Mars-like environment, finding and transporting objects, taking soil samples, and performing assembly tasks. Robot supervision was allowed, but there was 4 s latency and blackouts in the communication. The solution used by the team NimbRo Explorer is based on an elevation map for planning (Schwarz et al., 2016) and a combination of the global A* planner, which plans a rough plan with low-frequency replanning, and a precise local planner based on the dynamic window approach (Fox et al., 1997), employed in generating a local plan once per 4 s.

2.3 Overview of the DARPA SubT Challenge Teams

The motivational competition for the presented framework is the DARPA SubT Challenge (Chung, 2022), which consists of two parallel tracks: the *Systems* and *Virtual* tracks. Teams developed their full hardware

and software solutions in the Systems track to compete in physical courses. The Virtual track allows the teams to concentrate on software solutions evaluated in the DARPA-provided simulation environment. In both tracks, the task is to find artifacts in a given unknown environment within a one-hour time limit and report the artifacts' type and positions with 5 m accuracy in the global coordinate frame given by the entry gate to the mission course. The artifacts are objects like backpacks, fire extinguishers, or survivors. A single human supervisor outside the mission area can connect and control the robots through the base station in the Systems track. On the other hand, no human supervisor is allowed in the Virtual track. The herein presented work is mainly concerned with the Systems track that provides realistic conditions, close to real underground SAR missions, albeit it has also been deployed within the Virtual track. In the rest of this section, we briefly review selected solutions for the exploration, coordination, and planning proposed by other teams competing in the Systems track to provide a context of the solutions deployed in the SubT. In particular, solutions of the teams CERBERUS, CSIRO Data61, MARBLE, Explorer, CoSTAR, and PLUTO are briefly reviewed.

In Tunnel and Urban circuits (Tranzatto et al., 2022b), team CERBERUS used a local map centered around the robot to distinguish between known and unknown cells. The robot continuously generates an admissible path by a local planner that maximizes an exploration gain. When there are no places to explore within the local map, a global graph-based map planner (Dang et al., 2020) was used to reposition to the next unexplored area outside the local map or initiate homing. The homing is triggered if no unexplored areas are reachable by the robot while considering the remaining resources (battery). In the final circuit (Tranzatto et al., 2022c; Tranzatto et al., 2022a), the team used a similar idea of combining local exploration with switching to a global exploration planner when no places to explore were identified on the local map. In addition, they introduced a behavior tree in charge of handling commands from the human supervisor and high-level autonomous behaviors like time-limited exploration, unstacking, or homing. The human supervisor is in charge of coordinating the robot team.

Team CSIRO Data61 (Hudson et al., 2022) details the way to address the exploration in (Williams et al., 2020). Their approach developed during the Tunnel, Urban, and self-organized cave circuit is based on detecting a boundary between known and unknown space using visibility (Katz and Tal, 2015). Such boundaries are shared between the robots, where each boundary represents a task. The market-based task allocation method allocates the tasks to the robots.

Team MARBLE focused on an independent, autonomous exploration by each robot in Tunnel and Urban circuits (Ohradzansky et al., 2022). MARBLE used graph-based planning to generate a path to the next exploration waypoint like the CERBERUS team. Each waypoint is scored by a utility function combining information gained at the exploration waypoint and the travel cost to get to the waypoint.

Team Explorer used a hierarchical exploration planner with two levels (global and local) described in (Scherer et al., 2022), developed during the Tunnel, Urban, and self-organized cave circuit. The hierarchical planner uses sparse environment representation to plan coarse paths to distant locations outside a fine local map at the global level. At the local level, the planner uses a dense terrain model to plan a detailed path to improve the terrain coverage by the robot's sensors. The exploration strategy divides the environment into sub-maps with three possible states: "covered", "uncovered", and "covering." Based on the sub-map states, robots select uncovered sub-maps by changing the sub-map state to "covering" to inform the other robots that the sub-map is being covered so that they may select different uncovered sub-maps.

The techniques developed by team CoSTAR within the Tunnel and Urban circuit (Bouman et al., 2020; Agha et al., 2022) include a multi-layer traversability map. Local planners use the traversability map to avoid obstacles and generate plans that minimize the total path risk induced by uncertain perception. The exploration is based on a global graph that captures the connectivity of the free space in the environment. The global graph is built from edges and two types of nodes: breadcrumb and frontier nodes. The breadcrumb nodes correspond to already explored locations, and the frontier nodes indicate unexplored areas. The autonomous coordination of the robots is addressed by describing the whole task allocation problem using Planning Domain Definition Language (PDDL) (Fox and Long, 2002) solved by the OPTIC solver

(Benton et al., 2012). During the final circuit of the DARPA SubT, the team tested an exploration approach that takes into account connectivity between robots (Saboia et al., 2022).

Team PLUTO participated in the Tunnel circuit with an exploration framework designed specifically for exploring tunnel-like environments (Miller et al., 2020). The explorable parts of the tunnel are identified in the depth panoramas generated from the 3D LiDAR scans. The human supervisor is in charge of the robot coordination by specifying sequences such as "Left → Right → Right" to indicate which way each robot should take at the crossroads when multiple explorable ways are identified in the panorama.

3 Methodology

The whole exploration framework for the mission deployment consists of five core parts providing autonomous behaviors that a human mission supervisor can select. Alternatively, the system can be considered completely autonomous. The core parts include communication with a protocol suitable for low-bandwidth systems (Section 3.1) and proper initialization of the coordinate frames of all robots (Section 3.2) at the beginning of the mission. Safe autonomous navigation capabilities further need local mapping and traversability estimation (Section 3.3), while planning for relatively long distances is based on topometric maps (Section 3.4). Finally, the framework builds on decentralized multi-robot coordination (Section 3.5). The individual parts are detailed in the rest of this section. Note that technical details of the used technologies can be found in (Rouček et al., 2022).

3.1 Communication

The communication connecting the mission supervisor with the robots and the robots themselves is achieved by two independent communication systems used in parallel. The first system is considered to be high-bandwidth with more than $100\,\mathrm{kB\,s^{-1}}$ bandwidth, further referred to as the *high-bandwidth communication system*.The system is based on modules attached to the robots and base station. The bandwidth allows sharing of images of detected objects and eventually also compressed panoramic images from the cameras of the ground robots. However, its disadvantage is that it requires configuration and a known number of units before the mission because of utilized precise time multiplexing. Besides, the communication modules are relatively large, have high power consumption, and are also relatively expensive (thousands of USD per module), making them unsuitable for breadcrumbs dropped from the robots during the mission to extend the communication range to distant locations.

Table 1: Properties of the utilized communication systems

Communication system	**Bandwidth**	**Droppable modules**	**Images**	**K3**
High-bandwidth system	$>100\,\mathrm{kB\,s^{-1}}$	×	✓	✓
Low-bandwidth system	$<100\,\mathrm{B\,s^{-1}}$	✓	×	✓

The drawbacks of the first system are addressed by the second communication system based on affordable, small modules suitable to be dropped from the robots to extend the communication range. The system has a low bandwidth allowing broadcasts of less than 100 B per second from each robot. Therefore, the system is referred to as the *low-bandwidth communication system*, and it is utilized to transport only small messages called K3[1]. The properties of the considered communication systems are summarized in Table 1.

Both communication systems are used to share K3 messages that allow the commanding of the robots by the supervisor and sharing of information required for decentralized robot coordination when the robots operate fully autonomously. The data payload of each K3 message consists of 35 B structured into five fixed-sized blocks, according to Table 2.

[1]K3 stands for *Komunikace* (communication in Czech), ver. 3.

Table 2: Structure of the K3 message

Block	**Size** [Byte]
Type ID	1
Robot/Sender ID	1
Timestamp	4
Robot status	4
Message-specific data (such as navigational goal)	25

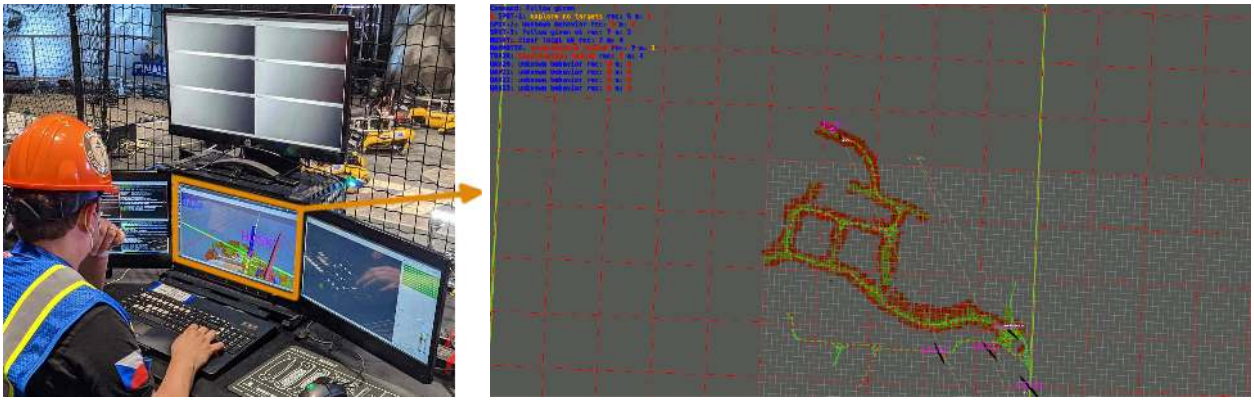


Figure 1: Four display setup of the supervisor station: the left display is used for the secure shell (ssh) connections to the robots, the right display shows the interface for artifact reporting, the top display is used to visualize compressed panoramic images received from the ground robots, and the middle display shows control interface used to command the robots.

The human supervisor triggers autonomous behaviors to control the robots using the base station composed of the workstation computer connected to both communication systems and equipped with four displays, see Fig. 1. The supervisor's commands to the robots are sent from the control interface visualized in the middle display of the base station. The supervisor can select one of the possible commands listed in Table 3 to be executed by all or a particular robot.

Table 3: Supervisor's commands to robots.

Command	**Description**
Plan and Navigate to the given waypoint (PW)	*Navigate the robot to a selected waypoint while avoiding obstacles.*
Navigate directly to the given waypoint (NW)	*Navigate directly to the waypoint, do not avoid obstacles.*
Navigate and stop before the obstacle (NSW)	*Navigate directly to the waypoint, and stop before the first obstacle.*
Stop (S)	*Stop the robot and deactivate planning.*
Explore (E)	*Start autonomous exploration.*
Reset local map (RLM)	*Reset the local map.*
Reset global map (RGM)	*Reset the topometrical map and observations.*
Deploy communication module (DCM)	*The robot attempts to drop a module to extend the communication.*
Remove waypoints in the area	*Remove waypoints and forbid waypoint creation in the selected area.*

Once the command is selected and confirmed, the corresponding K3 message is broadcasted by both communication systems. The robot receives the message given the message's *Robot ID*, either directly or through retranslation by breadcrumbs or other robots. The received command is executed if it is the most recent command received from the supervisor. Filtering the most recent command prevents executing delayed commands because of two independent communication channels with variable communication latency. The procedure of message sharing is summarized in Fig. 2.

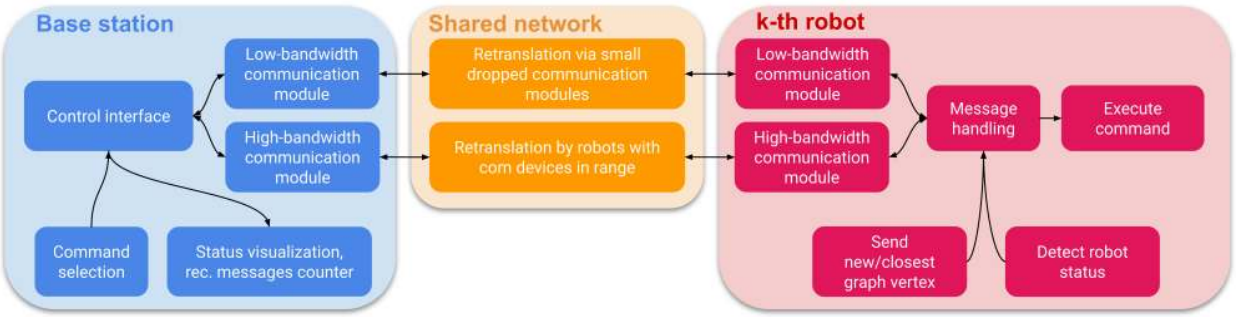


Figure 2: Sharing of the supervisor's commands.

Table 4: Status and coordination messages

Robot status	**Priority**	**Description**
The main sensor failed	19	Indicates the main sensor (LiDAR) is not providing range measurements.
Localization failed	18	The position of the robot was not provided for at least 0.5 s.
Local mapping failed	17	Local mapping is not providing any local map.
Waypoint reached	16	The selected location is reached, and the robot is stopped.
Plan cannot be found	15	No path to the selected waypoint was found, and the robot was stopped.
Automatically unstuck	14	Employing unstuck procedure.
No waypoints to follow	13	No explorable waypoints remain. Exploration finished.
No time to follow waypoints remain	12	No waypoints are reachable within the exploration time limit.
Exploration ok	11	Exploration in progress.
Exploration finished	10	Exploration finished. No waypoints remain.
Confirmation of the command† NW	9	-
Confirmation of the command† NSW	8	-
Confirmation of the command† PW	7	-
Confirmation of the command† RLM	6	-
Confirmation of the command† RGM	5	-
Confirmation of the command† S	4	-
Confirmation of the command† DCM	3	-
Secondary sensors failed	2	Optional sensor failure, sensor not necessary for localization or mapping.
Robot is waiting	1	-

† Confirmation of the commands according to Table 3.

K3 messages are utilized for multi-robot coordination and indicate the robots' current status as defined by the K3 message structure depicted in Table 2. The specific robot status and coordination entries are listed in Table 4. The status information appears at the base station according to the following principle. The status detection runs at 10 Hz continuously on each robot. The particular status values corresponding to the current robot situation are added to the 5 s long sliding window, from which the status with the highest priority is selected according to Table 4. The status is then combined with the closest collocated (or newly sampled) position of the robot, represented by the vertex of the topometrical model of the environment (further detailed in Section 3.4). Both communication systems broadcast the K3 message to propagate the information to the base station and other robots. If the broadcasted message reaches the base station, it increases the number of messages received from the robot to indicate connection quality to the supervisor. Moreover, if the message is the newest from the given robot, the robot's status is presented to the supervisor in the control GUI. Simultaneously, when other robots intercept the message, they extract the topometrical information from it that is then used for coordinated exploration.

3.2 Localization and Initial Coordinate Frame Estimation

The integral part of the localization is the initialization of the reference coordinate frame with respect to (w.r.t.) the DARPA-defined *SubT frame* given by the markers on the entry gate, as shown in Fig. 3. In the Tunnel circuit, the reference coordinate frame was initialized using images from the robot cameras and the AprilTag detector (Olson, 2011). The initialization of each robot was done just before the robot entered the mission area since the robot needed to observe the AprilTag markers. Therefore, the frame initialization procedure had to be triggered by the human supervisor. For Urban and Final circuits, the supervisor load has been decreased by using the total station to initialize the robot coordinate frame before the mission start. The initialization is performed for all robots in front of the entry gate by a pit crew member, as shown in Fig. 3.

Figure 3: The robots line up in front of the entry gate to the mission area for (left) the Urban circuit and (right) the Final circuit at the beginning of the run. The coordinate frame of the robot is initialized using the Leica TS-16 total station with a laptop computer mount and the retroreflective markers attached to the robot and entry gate.

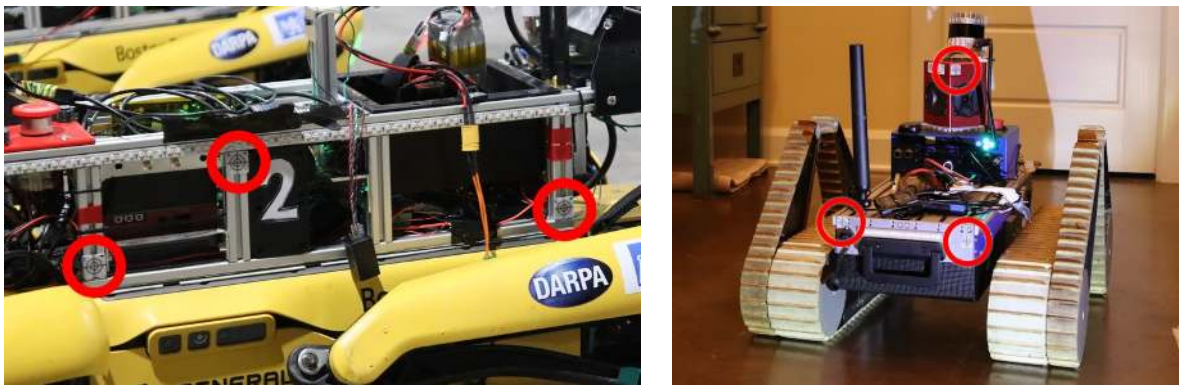


Figure 4: Three retroreflective markers (highlighted by red circles) rigidly attached to the robot serve for the robot coordinate frame initialization of (left) quadruped and (right) tracked robot.

The robot coordinate frame is initialized w.r.t. the entry gate using a set of three retroreflective markers rigidly attached to the robot, as visualized in Fig. 4. The position of the markers on each robot differs based on its construction. For quadrupeds, the markers are placed facing the robot's right side, and for the tracked (and wheeled) robots, the markers are placed facing the robot's rear.

The retroreflective markers attached to the robot define their coordinate frame called *markers frame* that is calibrated w.r.t. the robot's *base frame* prior to the deployment, as it is visualized in Fig. 4. The calibration

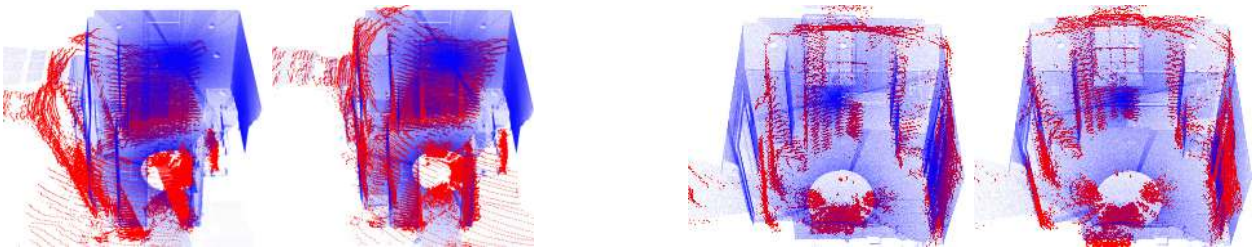


Figure 5: The left and right images show unaligned and aligned clouds in marker frame identification using a dense scan of a small room. The robot cloud (red) is aligned with the reference cloud (blue). Notice the density and precision of the reference (blue) cloud obtained by the Leica BLK-360 laser scanner.

of the static transformation between the *markers frame* and the robot's *base frame* can be established easily from the construction drawings of the robot if the precise position of the retroreflective markers on the body of the robot is known. However, in most cases, it is not the case; therefore, the following procedure has been utilized for the calibration.

First, three retroreflective markers are placed in a preferably small and empty room to establish a reference frame (*r*oom frame), similar to the mission entry gate. Next, a dense reference point cloud scan of the room is created using the total station or a 3D laser scanner such as the Leica BLK-360, which is a preferable option due to the speed of the process. Then, each robot is statically placed in the room, and a point cloud from its LiDAR is captured and aligned to the reference point cloud using the Iterative Closest Point (ICP) algorithm (Pomerleau et al., 2013), as visualized in Fig. 5. By precisely measuring the positions of the markers on the robot that establish the *markers frame* w.r.t. the *room frame* using the total station and having the LiDAR pose calculated by the ICP w.r.t. the *room frame*, the transformation between the LiDAR and the markers frame is calculated. The transformation between the robot's *base frame* and LiDAR sensor is known; hence, establishing the static transformation between the *markers frame* and the robot's *base frame* is a matter of capturing a single point cloud from the robot's sensors and measuring the markers on the robot using the total station.

Having the transformation between the *markers frame* and the robot's *base frame*, the initialization of the robot within the *SubT frame* at the beginning of the mission becomes a resection of the *markers frame* w.r.t. the *SubT frame*. Thus, the total station is resected w.r.t. the entry gate first, during the robot preparation to enter the mission course. Then, the transformation between the *SubT frame* and the *markers frame* is used to initialize the robot pose within the *SubT frame*.

The resection is employed to initialize the coordinates x, y, z, and the yaw angle of the robot only. The pitch and roll angles are initialized w.r.t. the gravity vector measured by the robot's onboard Inertial Measurement Unit (IMU). The reasons are twofold. First, the XY-plane of the *SubT frame* is defined to be horizontal, and the ICP-based localization utilizes a tight coupling of the IMU that estimates the pitch and roll angles during the map initialization. Therefore, only the yaw angle is initialized using the total station. Second, the farther apart the markers are placed on the robot, the smaller the induced overall angular resection error is. As the markers on the robot are placed in such a configuration that their vertical spread is lower than the horizontal, the yaw angle estimate is thus more precise than the pitch and roll estimate.

3.3 Local Mapping and Traversability Estimation

Based on the 3D scans localized by the method (Pomerleau et al., 2013) with various improvements, including deskewing (Deschênes et al., 2021), a dense local map is built for traversability assessment and precise local planning to avoid unpassable and hard-to-traverse areas. 3D scans are fused using the Bayes filter into the 3D occupancy grid map $\mathcal{M}_{3D}$, where each cell $\nu \in \mathcal{M}_{3D}$ also stores the height of the terrain $elevation(\nu)$, further used for traversability assessment. The map $\mathcal{M}_{3D}$ has the size limited by the distance r_{local} from

the robot's current position. The traversability assessment is performed for each map cell considering the kinematic and motion properties of the particular robot that are characterized by a tuple of parameters: r_{rad}, r_{height}, r_{step}, and r_{incl}. The radius of the robot circumference is marked by r_{rad}, r_{height} is the robot height, r_{step} is the maximal height of the climbable step, and r_{incl} is the maximal robot inclination.

Figure 6: A schema of the local elevation map creation and traversability assessment.

The traversability assessment of the cell ν evaluates its neighborhood $n_{cl}(\nu)$ defined by the horizontal size $2r_{rad}$ and vertical size $2r_{height}$. First, the vertical clearance $v_{clear}(\nu)$ is examined to satisfy the robot's limits $v_{clear}(\nu) > r_{height}$. Next, a *small elevation map* is created from $n_{cl}(\nu)$ to determine passability inspired by the approaches (Wermelinger et al., 2016; Stelzer et al., 2012). However, we use only the inclination of the robot and the step height and do not assess terrain roughness. The step height $\Delta(\nu)$ for a cell $\nu \in \mathcal{M}_{3D}$ is computed as

$$\Delta(\nu) = \min\left(\max_{\zeta \in 8nb(\nu)} |elevation(\nu) - elevation(\zeta)|, r_{step}\right), \tag{1}$$

where $8nb(\nu) \subset n_{cl}(\nu)$ is 8-neighborhood within the small elevation map related to ν, and $elevation(\nu)$ is the height associated with a particular cell within the small elevation map. The impassable areas (determined based on the maximal climbable step height) are enlarged by r_{rad} to account for the robot embodiment. The procedure for local map creation and traversability assessment is summarized in Fig. 6.

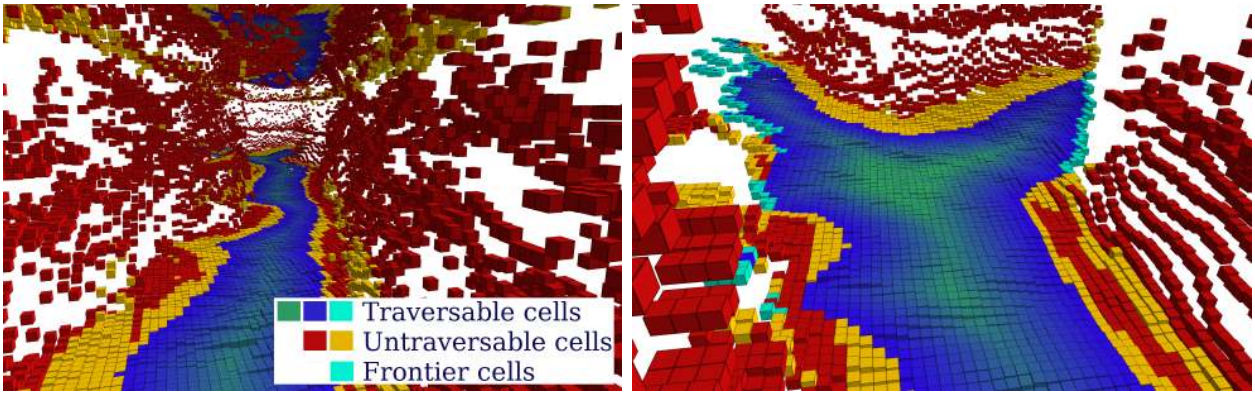


Figure 7: Examples of the local maps from the DARPA SubT Final circuit. The visualized traversability assessment is determined for the quadruped walking robot. The cells belonging to the tunnel's ceiling are assessed as traversable since the traversability assessment is independent of the current robot position. The untraversable walls do not allow the planner to find a path that would connect such traversable ceiling cells.

In the map with the assessed traversability, each passable cell is marked as the frontier cell if the cell has at least one three-cell column of the unoccupied cells in its 26-neighborhood. An example of the assessed local map with marked passability and frontier cells is visualized in Fig. 7.

In addition to the binary passability, we determine a suitable locomotion gait and calculate the traversability cost $c(\nu)$ associated with each cell $\nu \in \mathcal{M}_{3D}$. The step height $\Delta(\nu)$ is used to adjust the robot's locomotion parameters. Specifically for the quadruped robots, $\Delta(\nu)$ is utilized to adjust the locomotion gait and walking speed. For the **fast gait**, the robots need less time to explore flat areas than for the **crawl gait** that walks slowly with always at least three legs in contact with the ground. However, the fast gait is not suitable for rough terrains, for which the crawl gait is preferred. The gait type and walking speed are changed according

to $\Delta(\nu)$ associated with a set of cells $\mathrm{P} \subset \mathcal{M}_{3D}$ generated as the orthogonal projection of the current robot's path to the local map. The gait switching is based on thresholding of $c_P(\mathbf{r})$ value computed as

$$c_P(\mathbf{r}) = \max\left(c_{P,min}, \min\left(c_{P,max}, \underset{\zeta \in \mathrm{P}, |\mathbf{p}(\zeta)-\mathbf{r}|<\alpha r_{rad}}{\operatorname{argmax}} (\Delta(\zeta))\right)\right), \tag{2}$$

where $\mathbf{r}$ is the position of the robot and the gait is selected according to the limit value c_{crawl} as

$$\text{locomotion gait} \begin{cases} \textbf{crawl gait} & \text{if } c_P(\mathbf{r}) > c_{crawl}, \\ \textbf{fast gait} & \text{otherwise.} \end{cases} \tag{3}$$

The maximal walking speed is then computed as

$$v_{max}(\mathbf{r}) = v_{max} - \frac{(v_{max} - v_{min}) \cdot (c_P(\mathbf{r}) - c_{P,min})}{c_{P,max} - c_{P,min}}, \tag{4}$$

where v_{min} and v_{max} are the robot speed limits, $c_{P,min}$ and $c_{P,max}$ are parameters of the roughness-speed mapping.

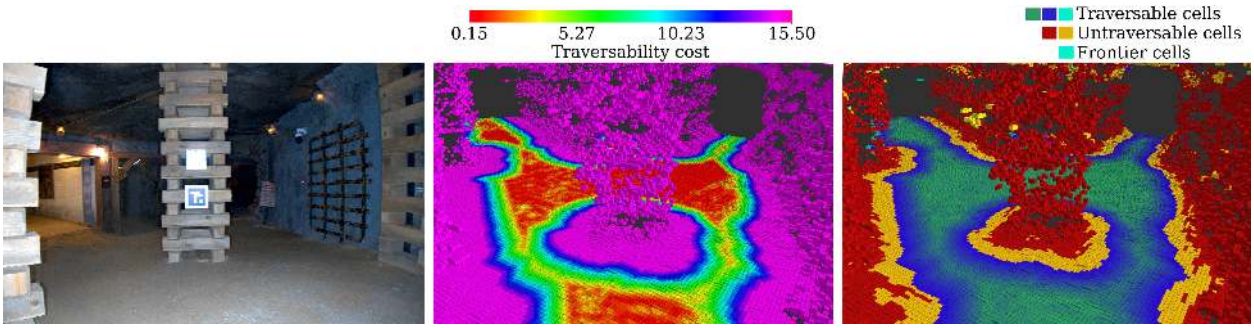


Figure 8: (left) Mostly flat terrain with walls and a column in the middle. (middle) Traversability cost evaluated for the quadruped walking robots shows the lowest costs associated with the cells that are farthest from the walls. (right) Traversability cost is visualized together with the passability assessment.

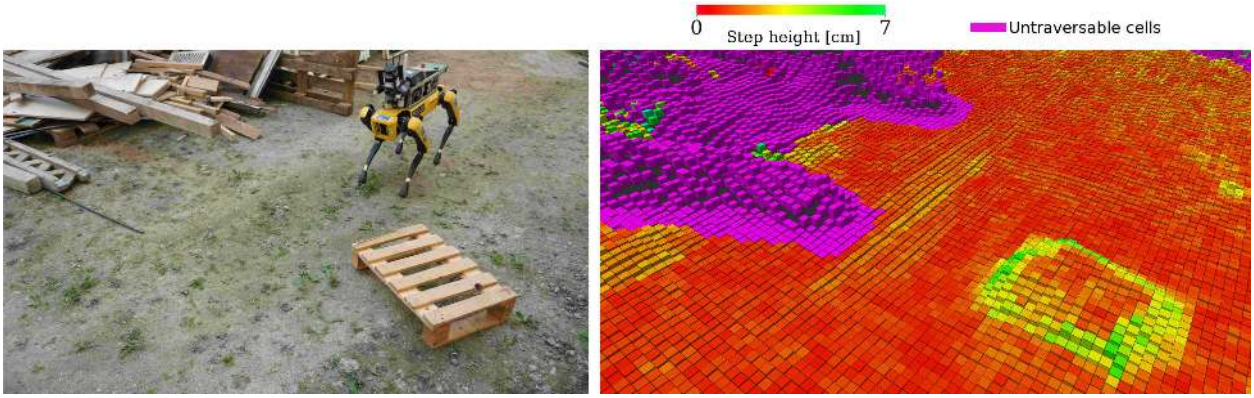


Figure 9: (left) Outdoor area with rough and untraversable terrains. (right) Terrain model built from LiDAR mounted on a quadruped robot with a visualized step height $\Delta(\nu)$.

The traversal cost combines the step height $\Delta(\nu)$ with the distance from the closest unpassable cell as

$$c(\nu) = c_{scale} \cdot \Delta(\nu) + \underset{\zeta \in M_{untrav}}{\operatorname{argmax}} \left(\max(0, c_{max} - c_{decr} \cdot |\mathbf{p}(\nu) - \mathbf{p}(\zeta)|)\right), \tag{5}$$

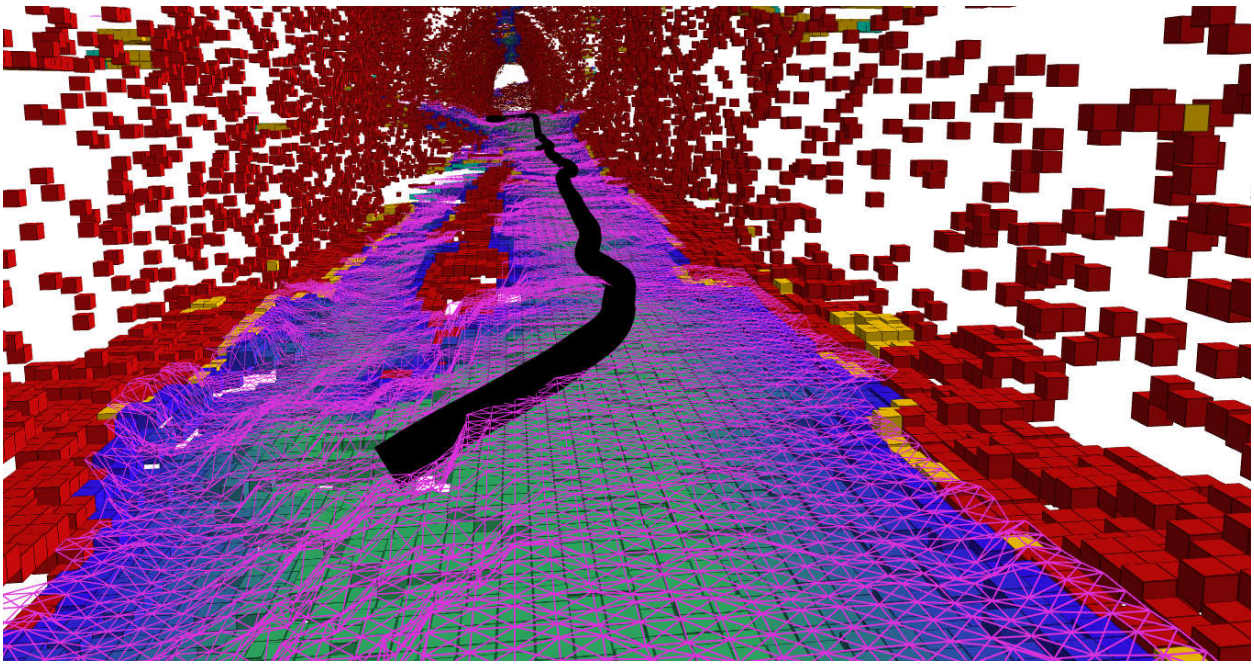


Figure 10: The graph (purple segments) for local planning generated from the traversability map. An example of a planned path smoothed by a sliding average filter is shown in black. The traversability cost pushes the path from the obstacles. The laser scan-based map is from the former military base located in Czechia.

where c_{scale}, c_{max}, and c_{decr} are empirically set parameters of the cost function, $M_{untrav} \subset \mathcal{M}_{3D}$ is a set of all untraversable cells, and $\mathbf{p}(\nu)$ is the position associated with the cell ν. Examples of the traversability cost assessment are visualized in Figs. 8 and 9.

The traversability cost weights the edges of the planning graph built from the local grid map. Vertices of the *local planning graph* are generated for the centers of all passable cells, and edges are created between all vertices corresponding to the neighboring cells in the grid map, see an example in Fig. 10. The path is found by Dijkstra's algorithm using the LEMON library (Dezső et al., 2011), where the traversability cost steers the path from the obstacles. The found plan is executed by the path-following module that inputs the path and current robot position to determine the next path waypoint. The Proportional-Integral-Derivative (PID) controller is utilized to steer the robot heading and correct the position error toward the waypoint, which moves on the horizon along the path as the robot moves forward.

The local map further plays an important role in estimating parts of the environment already covered by the robot's vision system and where the robot should navigate next to cover unseen parts and thus detect possible artifacts. For each passable cell ν, it is measured how much of the environment can be covered by visiting ν, and the amount of the uncovered environment is estimated based on the number of occupied cells marked as uncovered within the local map. The information gained by observing the occupied cell ν' is approximated by the information entropy. Since the distribution of the uncovered/covered cells is binary, the information gained by observing the environment from the cell ν is approximated as

$$h(\nu) = \sum_{\zeta \in \delta(\nu, d_{cam})} \begin{cases} 1 & \text{if observable}(\nu, \zeta) \text{ and } \neg frontier\_cell(\zeta), \\ h_{frontier} & \text{if observable}(\nu, \zeta) \text{ and } frontier\_cell(\zeta), \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $\delta(\nu, d_{cam})$ is the set of all cells ζ such that $\| \nu - \zeta \| < d_{cam}$. The function observable(ν, ζ) returns `true` if the cell ζ is observable from ν that is determined by ray-casting from ν to ζ in the current 3D grid

map $\mathcal{M}_{3D}$. d_{cam} is the distance at which the robot's camera can cover a cell.[2] From (6), it can be seen that information gained by observing cells at the edge of the environment is defined by $h_{frontier}$ that is set $h_{frontier} > 1$ to reflect the robot would possibly also cover neighboring cells that are currently unknown.

Algorithm 1: Cluster Entropy Representatives

Input: $\mathcal{M}_{3D}$– 3D grid map with assessed frontiers, c_{radius}– Cluster radius, c_{min_cells}– Minimal cluster size.

Output: $\mathbf{W}_k$– Clustered entropy.

Procedure `cluster(`$\mathcal{M}_{3D}$`)`
- $A \leftarrow \emptyset$ ▷ Init. set of clusters.
- **for** $\nu \in \mathcal{M}_{3D} : h(\nu) > h_{min}$ **do** ▷ For each map cell with non-zero entropy.
  - **if** $A = \emptyset$ **then** ▷ If no clusters in set.
    - $A \leftarrow \{\{\nu\}\}$ ▷ Create a new cluster.
  - **else**
    - $d \leftarrow$ `distanceToClosestCluster(`ν, A`)`
    - **if** $d < c_{radius}$ **then**
      - `addToClosestCluster(`ν, A`)` ▷ Add ν to existing cluster.
    - **else**
      - $A \leftarrow A \cup \{\{\nu\}\}$ ▷ Create a new cluster with ν.
- $\mathbf{W}_k \leftarrow \emptyset$ ▷ Init. cluster representatives.
- **for** $A_i \in A$ **do** ▷ For each cluster.
  - **if** $|A_i| > c_{min\_cells}$ **then**
    - $\mathbf{W}_k = \mathbf{W}_k \cup \{$`cellClosestToAverageCoordinates(`$A_i$`)`$\}$ ▷ Create new representatives.
- **return** $\mathbf{W}_k$

The number of cells ν with $h(\nu) > 0$ can be relatively high, and thus we further employ a two-pass clustering algorithm summarized in Algorithm 1 to decrease the number of possible exploration waypoints. The clustering algorithm creates a set of waypoints $\mathbf{W}_k$ that are the cluster representatives selected as a subset of frontier cells from $\mathcal{M}_{3D}$ so that the cluster representatives are at least c_{radius} distant from each other, and each cluster representative is associated with at least c_{min_cells} cells ν with $h(\nu) > h_{min}$, which removes cells that are considered to be noise.

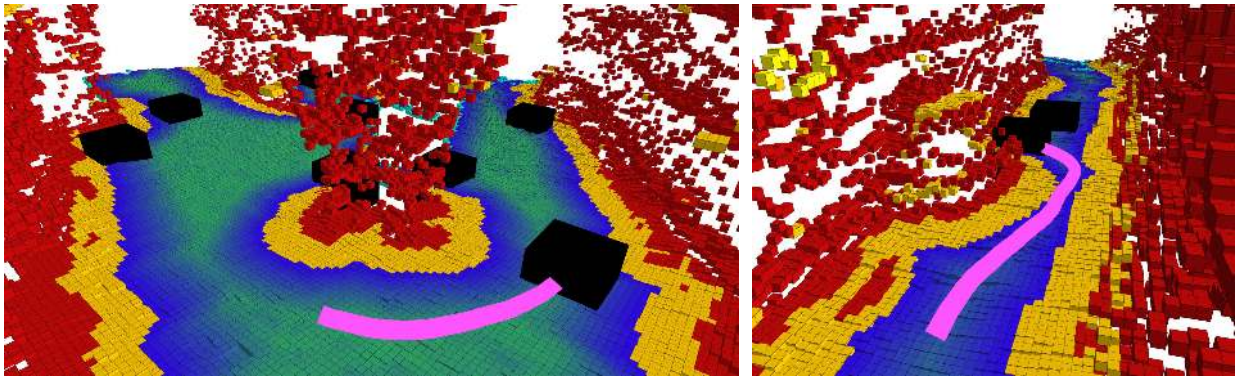


Figure 11: Exploration waypoints (visualized as the black boxed markers) used during the exploration of the DARPA SubT Final circuit by one of the deployed quadruped robots.

The resulting set of waypoints $\mathbf{W}_k$ is merged with the set of all active waypoints $\mathbf{W}$. The list of waypoints $\mathbf{W}$ is maintained by each robot independently, including the waypoint removal. The waypoints are removed

[2] All the deployed ground robots are equipped with an omnidirectional vision camera sensing.

from **W** based on the local map when a related part of the environment is covered. Moreover, a waypoint is also removed from **W** if another robot indicates its presence closer to the waypoint than d_{cam}. The presence of other robots is determined based on the shared K3 status messages, and it is maintained within the topometrical map based on (Bayer and Faigl, 2021b), further detailed in the following section. Examples of the determined exploration waypoints are shown in Fig. 11.

3.4 Topometric Mapping

The local map provides high accuracy because its high resolution allows precise local navigation, but its range is limited to r_{local}. However, because of memory requirements, it cannot capture a large-scale environment, which is essential for coordinating multiple robots and holding information about all the covered parts of the environment. Moreover, the local map is not shared through the considered communication systems since the required bandwidth would be too high for a low-bandwidth communication system, and the high-bandwidth communication system is already allocated for reporting the artifacts and streaming data from cameras. Therefore, a global environment map is maintained using a relatively sparse model using the topometric map (Bayer and Faigl, 2021b), sharable through both communication systems using K3 messages. The local and topometrical maps are used together for planning to locations outside the local map, as visualized in Fig. 12.

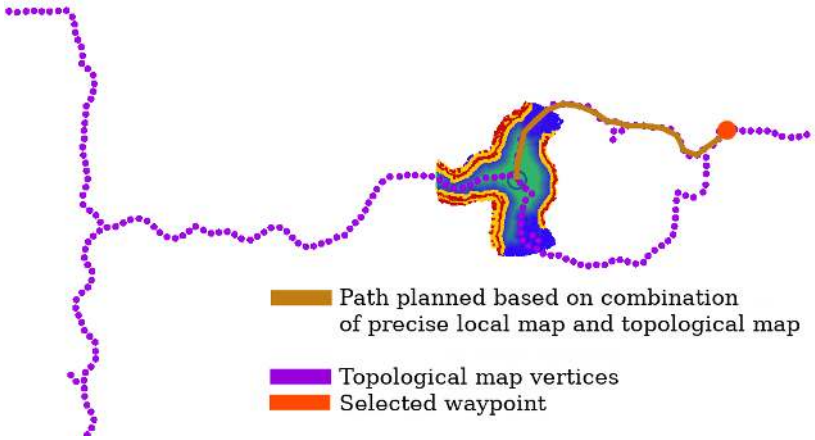


Figure 12: Path planned using dense local and topometrical maps. The local map is used for avoiding obstacles and places with high traversability costs within the close surroundings of the robot. The topometrical map extends the plan from the local map to a distant waypoint location by assuming the waypoint is reachable because a robot can visit that location while building the topometrical map. In the case of a dynamic environment, some of the waypoints might become unreachable. Then, the robot removes the task related to the waypoint when an attempt to reach the waypoint fails. Suppose the robot detects an untraversable environment incident with the topometrical map while attempting to reach a certain waypoint. In that case, the part of the topometrical map incident with the untraversable area is removed.

The topometrical map can be defined as a graph $\mathbf{T}(\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is a set of graph vertices, and $\mathbf{E}$ is a set of bidirectional edges. Each edge $\gamma(v_1, v_2) \in \mathbf{E}$ represents a traversable path connecting the positions of the incident vertices. A cost associated with each edge $\gamma(v_1, v_2) \in \mathbf{E}$ is based on the Euclidean distance between the positions of the corresponding vertices v_1 and v_2. A single vertex $v \in \mathbf{V}$ is associated with the tuple $(\mathbf{p}, ids, h_c)$ denoting a 3D position of the corresponding place $\mathbf{p} = (p_x, p_y, p_z)$ visited by one or more robots recorded in the list ids, and the estimated number of the nearby waypoints h_c. A new vertex is added to the graph when the position of the robot $\mathbf{r}$ is at least t_{build} far from the closest $\mathbf{p}$ of the current vertices $\mathbf{V}$.

The number of nearby waypoints h_c means that when a robot is at the vertex location, it should recognize h_c waypoints within its local map. h_c is initialized when the vertex is created, and its value is decreased

when a waypoint with the corresponding position is removed because it has been covered (visited by the robot or other robots). Since the value h_c can only decrease, it prevents possible oscillations in selecting the next exploration waypoints. A path in graph $\mathbf{T}$ can be found by a graph-search algorithm such as Dijkstra's algorithm, where the cost of each edge is based on the Euclidean distance between the vertices using vertex's $\mathbf{p}$.

Table 5: Compatible traversability capabilities

Robot	Compatible traversability capabilities					
	Spot 1	Spot 2	Spot 3	Husky	TRADR	Marmotte
Quadruped Spot 1	✓	✓	✓	✓	✓	✓
Quadruped Spot 2	✓	✓	✓	✓	✓	✓
Quadruped Spot 3	✓	✓	✓	✓	✓	✓
Wheeled Husky	×	×	×	✓	×	×
Tracked TRADR	×	×	×	✓	✓	✓
Tracked Marmotte	×	×	×	✓	×	✓

The topometrical map, in addition to the large-scale planning and capturing waypoints, is further utilized to share knowledge about the environment and multi-robot coordination. In a heterogeneous team of robots, the robots have different traversability capabilities that need to be taken into account. For example, our quadruped robots can traverse most of the terrain passable by the wheeled robots, but some passages traversable by the quadruped robots are considered impassable by the wheeled robot, see the used robots in Fig. 20. Therefore, building the topometrical map using data from the other robots needs to consider the traversal capabilities of each robot that are defined prior to the mission deployment and thus associated with the particular robot identifier *id*.

During the deployment in Systems track, our ground robots team (depicted in Fig. 20) consists of a wheeled robot (Husky), two tracked robots (TRADR and Marmotte), and three quadrupeds (Spot 1–3). The considered compatible traversability capabilities among the deployed robots are listed in Table 5. The most capable platforms are quadruped robots. On the other hand, the wheeled robot is considered less capable, while the tracked robot TRADR is more capable than Marmotte. Notice that the topometrical information is always utilized in coordination to determine parts of the environment already covered. The capabilities are considered only in estimating that a waypoint can be reachable by the particular robot.

The topometrical map is built from low-memory footprint K3 messages that are transferable by the low-bandwidth communication channel. Thus, the topometrical maps are efficiently shared by the K3 messages between the robots. The update of the topometrical map is summarized in Algorithm 2. For readability, the algorithm does not include handling compatible traversal capabilities, as listed in Table 5.

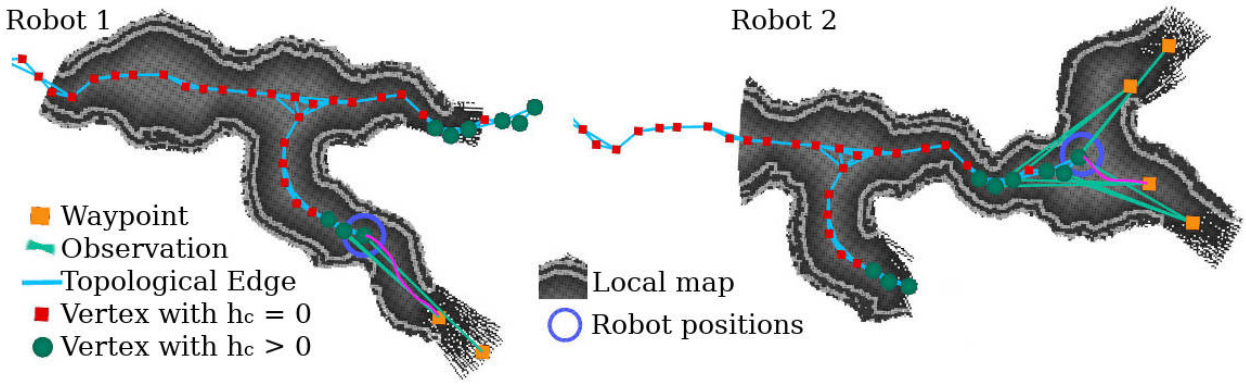


Figure 13: Visualization of the shared topometrical map.

Algorithm 2: Topometrical map update.

Input: $\mathbf{T}(\mathbf{V}, \mathbf{E})$ - Topometrical map, where each $v \in \mathbf{V}$ is associated with the position $\mathbf{p}$, a set of robot $ids[]$, and h_c, $(\mathbf{p}, id, h_c)$ - New topometrical information, t_{build}, $t_{connect}$ - Topometrical parameters.

Output: $\mathbf{T}'(\mathbf{V}, \mathbf{E})$ - Updated topological map.

if $\mathbf{V} = \emptyset$ **then**
  $v \leftarrow$ create_vertex$(\mathbf{p}, id, h_c)$
  $\mathbf{V} \leftarrow \mathbf{V} \cup \{v\}$ ▷ Initialize map.
else
  $q \leftarrow \underset{v \in \mathbf{V}}{\operatorname{argmin}} (\| v.\mathbf{p} - \mathbf{p} \|)$ ▷ Find the closest vertex.
  if $\| q.\mathbf{p} - \mathbf{p} \| < t_{build}$ **then**
    $q.ids \leftarrow q.ids \cup id$ ▷ Record the visitor's ids.
    if $h_c < q.h_c$ **then**
      $q.h_c \leftarrow h_c$ ▷ No. of waypoints can only decrease.
  else
    $v \leftarrow$ create_vertex$(\mathbf{p}, id, h_c)$
    $\mathbf{E} \leftarrow \mathbf{E} \cup \{(q, v) \,|\, q \in \mathbf{V} \cap \| q.\mathbf{p} - v.\mathbf{p} \| \leq t_{connect}\}$
    $\mathbf{V} \leftarrow \mathbf{V} \cup \{v\}$
return $\mathbf{T}'(\mathbf{V}, \mathbf{E})$

When updating the topometrical map with new information from other robots, the set of all waypoints $\mathbf{W}$ is updated by removing waypoints related to the covered areas and possibly adding new waypoints $\mathbf{W}_k$. The update also includes waypoints generated on each vertex unvisited by the robot, where some other robots indicated $h_c > 0$. An example of shared topometrical maps between two robots is visualized in Fig. 13.

3.4.1 Planning Time Requirements

A new plan is generated using the planning with both parts: the dense local map and topometrical model using Dijkstra's algorithm with the computational complexity that can be bounded by $O(e_\alpha + v_\alpha \log(v_\alpha))$, where e_α is the number of graph edges and v_α is the number of vertices. The main difference between the run times of both parts of the planning is induced by the sizes of the graphs that are searched. We benchmarked the proposed planning approach by exploring a cave-like test scenario and using the obtained local and topometrical models to generate a plan. The parameterization of the exploration framework is the same as for the Virtual track of the SubT Challenge. The relevant parameters consist of the minimal distance of topometrical vertices 0.7 m and the resolution of the local map 7 cm.

The generated topometrical map consisted of 350 vertices and 1724 edges, and it is shown in Fig. 14. The graph corresponding to the local map contained from 30 k to 45 k vertices and from 100 k to 180 k edges. Planning the 208 m long path shown in Fig. 14 took less than 1.5 ms to plan within the topometrical map. A safe plan is generated in 199.7 ms within the local map (with 41 k vertices) to connect the robot position with the topometrical plan. Note that since Dijkstra's algorithm is used to find a path within the topometrical map, the time to find the path is approximately the same as the time required to compute path lengths to all waypoints lying on topometrical vertices, which is further used for the coordination.

3.5 Decentralized Multi-robot Coordination

The coordination in multi-robot exploration is approached as the task allocation problem, where the tasks correspond with visiting the waypoints $\mathbf{W}$. The time to complete each task is estimated as the time required to reach the waypoint location and report information found at the location to the base station. The

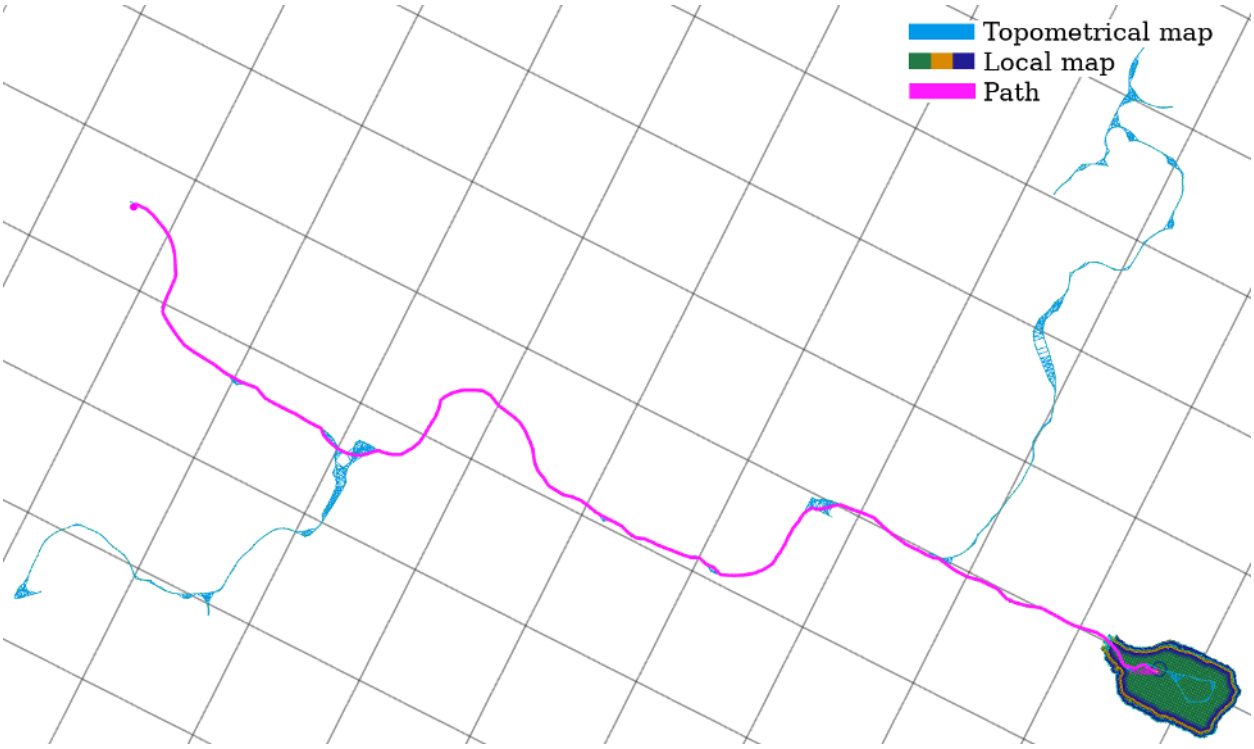


Figure 14: Topometrical map with highlighted path and a local map used for benchmarking the path planning using the Intel 8-th generation of the i7-class CPU with 32 GB RAM. The cell size of the visualization grid is 20 m.

developed solution is based on a ranking method inspired by the MinPos (Bautin et al., 2012) that is generalized to fully decentralized decision-making, where each robot solves the task allocation itself.

The set of all waypoints (tasks) is first pre-filtered by removing tasks associated with unreachable waypoints and tasks that cannot be finished within the remaining time to the mission end. The required time $t(\mathbf{w}_i, \mathbf{r})$ to reach the waypoint $\mathbf{w}_i$ from the current robot position $\mathbf{r}$ is further prolonged by the time to report a possibly found artifact that is estimated as the minimal time $t_{com}(\mathbf{w}_i)$ required to return from $\mathbf{w}_i$ to a location $\mathbf{c} \in \mathbf{C}$, where the robot detected the possibility of transmitting data back to the base station.

$$t_{com}(\mathbf{w}_i) = \underset{\mathbf{c}\in\mathbf{C}}{\operatorname{argmin}} \left(\frac{\mathcal{L}(p_{\mathbf{w}_i,\mathbf{c}})}{\overline{v}_r} \right), \tag{7}$$

where $\mathcal{L}(p_{\mathbf{w}_i,\mathbf{c}})$ is the length of the shortest path from $\mathbf{w}_i$ to $\mathbf{c}$. So each task associated with a waypoint location $\mathbf{w}_i$, for which $t(\mathbf{w}_i, \mathbf{r}) + t_{com}(\mathbf{w}_i)$ is above the remaining mission time, is filtered out from the tasks set.

The task assignment problem ranks the tasks w.r.t. the most recently known positions of other robots, and information about areas visited by the robots stored in the topometrical map $\mathbf{T}(\mathbf{V}, \mathbf{E})$ (Bayer and Faigl, 2021a). We split the set of all n robots $\mathbf{R}$ into the actual robot r, for which the ranks are computed, and the set of the remaining $m = n - 1$ robots $\mathbf{R}' = \mathbf{R} \setminus \{r\}$. The rank G_i of the i-th task associated with the waypoint $\mathbf{w}_i \in \mathbf{W}$ is composed of two parts, each enforcing the different behavior of the robot

$$G_i = G_{i,cross} + G_{i,spread}. \tag{8}$$

The rank part $G_{i,cross}$ aims to spread the robots at the crossroads uniformly by penalizing the waypoints close to the locations visited by the other robots $\mathbf{R}'$. The locations visited by the other robots are obtained

from the vertices of the topometrical map $\mathbf{V}$, which have *ids* different than the actual robot r. The rank $G_{i,cross}$ is computed as

$$G_{i,cross} = 2\sum_{k=1}^{m} g_{i,cross}, \tag{9}$$

where $g_{i,cross}$ is

$$g_{i,cross} \begin{cases} 1 & m_r > \left\| \mathbf{p}_i - \underset{\mathbf{h}_{k,l} \in \mathbf{H}_k}{\operatorname{argmin}}(\|\mathbf{h}_{k,l} - \mathbf{p}_i\|) \right\|. \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$\mathbf{H}_k$ is a set of positions associated with the set of vertices received by the actual robot r, marked as visited by the k-th robot from $\mathbf{R}'$ with id β: $\{v \subset (\mathbf{V}), v.id = \beta\}$, and $\mathbf{p}_i$ is the position of the i-th waypoint, thus the waypoint associated with the task that is being ranked. The thresholding parameter m_r limits the distance from the recorded robot positions on which the ranks of possible waypoints are increased.

Suitable values of the threshold m_r have been analyzed empirically. The value of m_r should always be higher than half of the robot's largest diameter and always smaller than the sensory range. Furthermore, for subterranean environments, the suitable value of m_r should be set between half of the tunnel width and the distance between tunnel entrances at the crossroads if these parameters are known.

The purpose of the rank part $G_{i,spread}$ is to split robots as far from each other as possible. It is used to distinguish what task is more suitable to be handled by the robot r if the task is not distinguishable only based on $G_{i,cross}$. The idea is to get the r-th robot as far from the other robots as possible based on the most recent positions of the robots. It is inspired by the MinPos ranking (Bautin et al., 2012) that counts how many of the robots $\mathbf{R}'$ have a shorter path to the i-th waypoint than the actual robot r.

Under low-bandwidth communication, detailed maps for computing shortest paths are not available. Besides, the traversal capabilities might be incompatible between heterogeneous robots. Therefore, the path lengths are approximated by the Euclidean distance (further denoted as $\|\cdot\|$) between a particular robot position $\mathbf{r}$ and possible waypoint $\mathbf{p}_i$. We further propose to take into account distances between the robots and waypoints and not only count the number of robots closer to the waypoints than the robot r as in the MinPos. Thus, the proposed global rank $G_{i,spread}$ is computed as

$$G_{i,spread} = \frac{1}{m} \min \left(\sum_{k=1}^{m} \frac{\|\mathbf{p}_i - \mathbf{r}\|}{\|\mathbf{p}_i - \mathbf{h}_{k,\zeta}\|}, m \right), \tag{11}$$

where $\mathbf{r} \in \mathbb{R}^3$ is the current position of the r-th robot and $\mathbf{h}_{k,\zeta}$ is the most recently received position of the k-th robot from $\mathbf{R}'$.

The resulting task is selected as the task with the lowest rank G_i, according to (8). If multiple tasks have the lowest rank, the resulting task is selected as the one finishable in a shorter time. The task allocation procedure is triggered when the robot travels 75 % of the actual path to the next waypoint or 15 s after the last task allocation is performed. The exploration ends when the task list is empty, or none of the waypoints is reachable.

Remark 1: Based on (11), it always holds that $G_{i,spread} \leq 1$. If task A has a lower local part of the rank than task B, task A would also have a lower total rank, regardless of the global part of the rank.

4 Results

The developed autonomous multi-robot exploration system has been experimentally examined in several deployments. In this section, we present results from the deployments in both the Virtual and Systems tracks of the SubT Final circuit, including the Post-event testing session. The proposed system has been deployed in a fully autonomous and supervised mode. In the Virtual track, the system runs in fully autonomous mode. Contrary, the supervisor was in charge of teleoperating or selecting autonomous behaviors for particular robots in the Systems track. Besides, the system was deployed in fully autonomous mode during the Post-event testing within the real environment of the Final circuit of the Systems track.

Table 6: System's parametrization of the ground robots for the Virtual and Systems tracks of the SubT Finals.

Parameter	Virtual track	Systems track			
	DTR	Spot 1-3	Husky	TRADR	Marmotte
Local map resolution [cm]	7	4.5	5	5	5
Local map size r_{local} [m]	36	15	15	15	15
Radius of the robot r_{rad} [cm]	31	28	33	30	33
Robot height r_{height} [m]	0.90	1.00	0.85	1.00	1.00
Maximal climbable step r_{step} [cm]	12	24	15	12	20
Maximal inclination r_{incl} [°]	33	40	24	24	22
Minimal robot speed v_{min} [$\mathrm{m\,s^{-1}}$]	-	0.3	-	-	-
Maximal robot speed v_{max} [$\mathrm{m\,s^{-1}}$]	1.0	0.7	0.7	0.7	0.7
Roughness-speed mapping $c_{P,min}$ [cm]	-	6	-	-	-
Roughness-speed mapping $c_{P,max}$ [cm]	-	10	-	-	-
Cost function parameter c_{scale}	240	100	20	0	0
Cost function parameter c_{max}	343	344	64	21	21
Cost function parameter c_{decr}	23	16	4	1	1
Camera observation distance d_{cam} [m]	6	6	6	6	6
Informain from frontiers $h_{frontier}$ [bit]	15	12	12	12	12
Minimal cluster size c_{min_cells}	7	6	6	6	6
Cluster radius c_{radius} [m]	3.5	5.0	5.0	5.0	5.0
Minimal cell information h_{min} [bit]	100	60	60	60	60
Distance to add new vert t_{build} [m]	0.7	0.4	0.4	0.4	0.4
Distance to create edge $t_{connect}$ [m]	1.75	1.00	1.00	1.00	1.00
Threshold related to the coordination m_r [m]	5.0	5.0	5.0	5.0	5.0

The proposed system's parameters for the employed ground robots in the evaluation scenarios are summarized in Table 6. When setting the cost function parameters, we prefer to decrease the risk that robots would hit obstacles. It was done by setting the cost field induced by the presence of obstacles so that the robots would prefer to pass the obstacles while being approx. 0.8 m and more from the obstacles. When the corridor is $2 \cdot 0.8 + r_{rad}$ wide or tighter, the robots tend to pass the corridor while being in the middle between the walls. The particular results and lessons learned are presented in the following parts.

4.1 Virtual Track Competition

In the Virtual track of the DARPA SubT Challenge, the total number of deployable vehicles and their types are limited by the fixed amount of virtual credits reflecting the estimated real-world price of the robots and their sensory equipment. For our CTU-CRAS-NORLAB team, we chose two Unmanned Ground Vehicles (UGVs) and five Unmanned Aerial Vehicles (UAVs). The models of the used vehicles are visualized in Fig. 15.

The deployment strategy of the whole team was as follows. First, ground vehicles were deployed to explore the environment and build a communication network by dropping retranslation modules (breadcrumbs).

Figure 15: Models of the used robots in the Virtual track of the DARPA SubT Final circuit. Dynamic Tracked Robot (DTR) developed by CSIRO Data61 team and CTU-CRAS-NORLAB aerial vehicle.

Then, one aerial vehicle was deployed after the deployment of the UGVs. The remaining four UAVs were deployed half an hour before the end of the one-hour mission because of the limited flight time of 30 min. Thus, the UAVs can utilize the already built communication network supporting possible coordination and enabling reports of the found artifacts without returning close to the base station. The rest of the presented results are focused on the ground vehicles as the presented methods targeted the exploration with UGVs because aerial vehicles have been addressed independently (Petráček et al., 2021).

The final round of the DARPA SubT Challenge Virtual track consists of three exploration trials in eight different environments in addition to several environments for testing the exploration methods before the final round. There were 20 artifacts in each environment. The final round environments were partially based on the testing environments and partially composed of new environments that were unknown before the final round. The environments used in the final round are visualized in Fig. 16, together with the paths of the ground vehicles from the first out of three performed trials per each environment. A commented example of the proposed coordination method performance in the fifth environment is visualized in Fig. 17.

Table 7: Summarized results achieved during the Virtual track of the Final round.

Environment (trial)	1(1)	1(2)	1(3)	2(1)	2(2)	2(3)	3(1)	3(2)	3(3)	4(1)	4(2)	4(3)
l_{avg} for UGVs [km]	1.3	0.8	0.8	0.8	1.8	1.2	1.8	1.9	1.7	1.1	1.1	1.3
l_{avg} for UAVs [km]	1.9	2.1	1.9	1.2	0.7	0.9	2.4	1.8	2.5	0.8	0.9	0.8
v_{avg} for UGVs [km h^{-1}]	1.4	0.8	1.1	2.2	2.2	2.2	1.8	1.9	1.8	1.4	2.2	2.2
v_{avg} for UAVs [km h^{-1}]	6.1	6.1	6.1	6.5	4.7	4.3	5.8	5.4	6.1	7.2	5.4	6.8
Artifacts detected	9	10	12	16	11	12	16	11	18	7	4	5

Environment (trial)	5(1)	5(2)	5(3)	6(1)	6(2)	6(3)	7(1)	7(2)	7(3)	8(1)	8(2)	8(3)
l_{avg} for UGVs [km]	1.1	1.4	1.7	0.2	0.6	0.2	1.5	1.4	0.9	0.3	0.5	0.3
l_{avg} for UAVs [km]	0.4	0.6	0.5	0.2	0.1	0.1	2.3	1.4	2.2	0.2	0.2	0.2
v_{avg} for UGVs [km h^{-1}]	1.8	1.8	1.8	1.4	1.1	1.1	1.5	1.8	1.8	0.7	1.1	1.1
v_{avg} for UAVs [km h^{-1}]	4.7	4.3	5.0	3.2	3.2	4.0	6.1	6.1	5.8	4.3	5.8	4.7
Artifacts detected	5	6	7	5	4	4	13	9	14	6	5	6

A summary of the results for all the trials of the final rounds, including average traveled path lengths l_{avg} for the UGVs and UAVs and average traveling speeds v_{avg} is presented in Table 7. The results reveal that the UGVs did not travel far in some environments, which indicates limitations of the proposed method

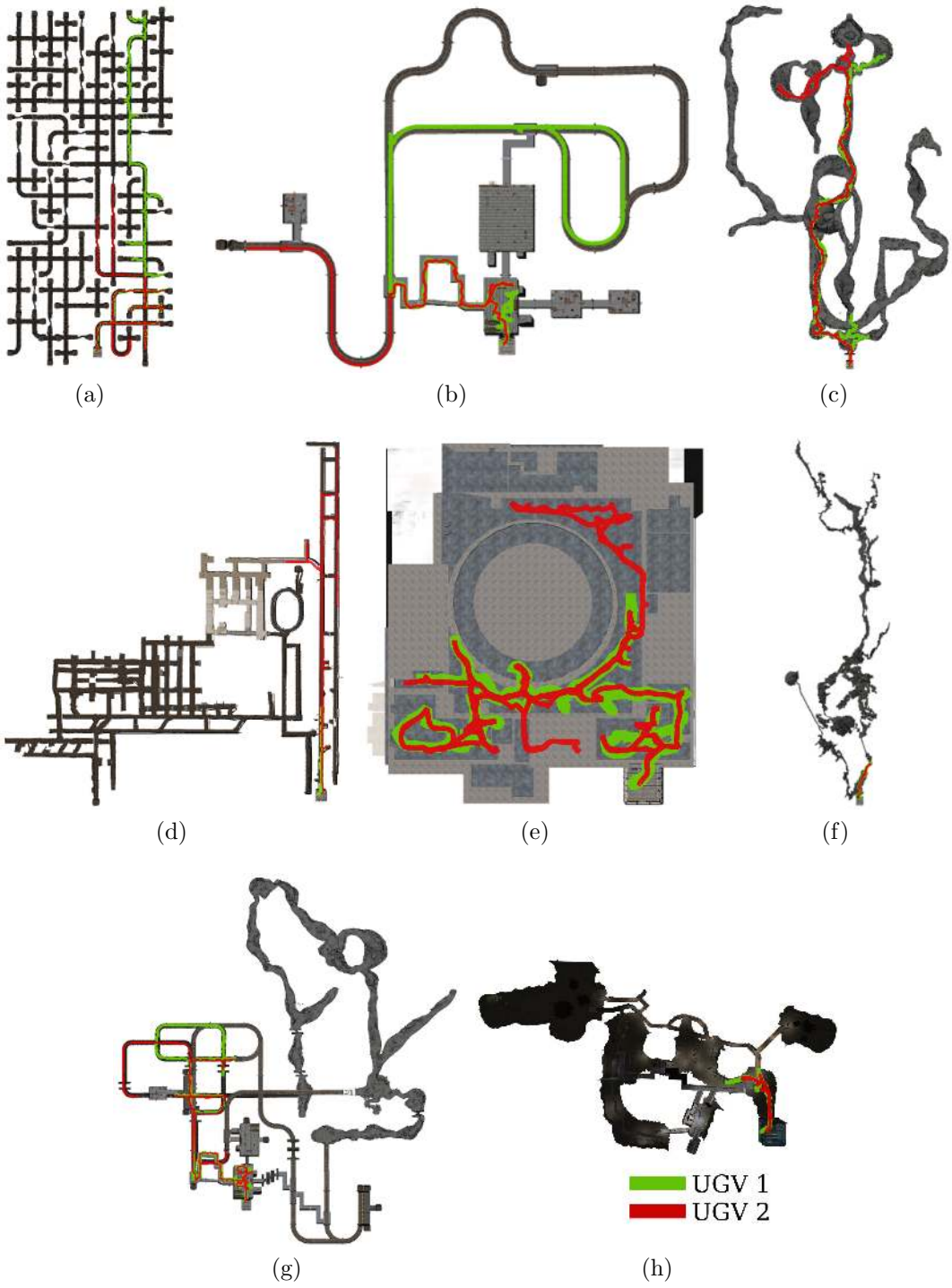


Figure 16: The environments of the final round of the DARPA SubT Challenge Virtual track. Visualized paths of the ground vehicles were generated from the first trials of the competition.

parametrization.

On the other hand, in other environments, such as the third trial of Environment 4, detailed in Fig. 18, the robots operate until the time limit. We further detail the behavior of the proposed method in selected

(a) 820 s (b) 1190 s (c) 3040 s

Figure 17: Visualization of the exploration progress by colored paths of the exploring ground robots at particular time instances of Environment 5 (Fig. 16e). After 820 s, the robot with the green path separates from the other robot at the main crossroad since waypoints near the path marked by the red color had higher ranks than the waypoint located at the not yet visited part of the environment. After 1190 s, the robot with the red path returns to the main crossroad. Since the area shown in the lower part of the visualized environment map is explored, the robot enters the second part of the environment already being explored by the other robot. After 3040 s from the mission start, the environment is explored by the robots with partially overlapping paths. Such behavior is intentional, and it increases the system's robustness.

environments and describe factors that kept ground robots from exploring some parts of the environments or caused the robots to get stuck.

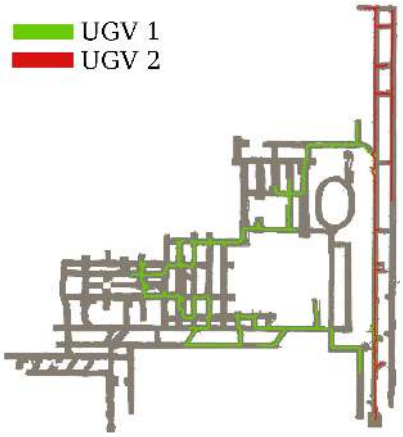


Figure 18: Traveled paths of the ground robots in the final round of the Virtual track, Environment 4, Trial 3.

4.1.1 Lessons Learned from the Final Circuit of the Virtual Track

By analyzing robots' behavior in environments shown in Fig. 16 and the results achieved by the UGVs summarized in Table 7, we provide the following lessons learned.

- We have observed a few limitations related to segments with tracks, especially in environments shown in Figs. 16b and 16g. The areas where the probability that the robots would get stuck or not explore further was increased at the crossroads shown in Fig. 19a. In straight tunnels with rails, the

(a) Crossroad with a marked part, where it was hard to assess traversability

(b) Long monotonous railway tunnel

(c) Entry to the railway with a marked edge

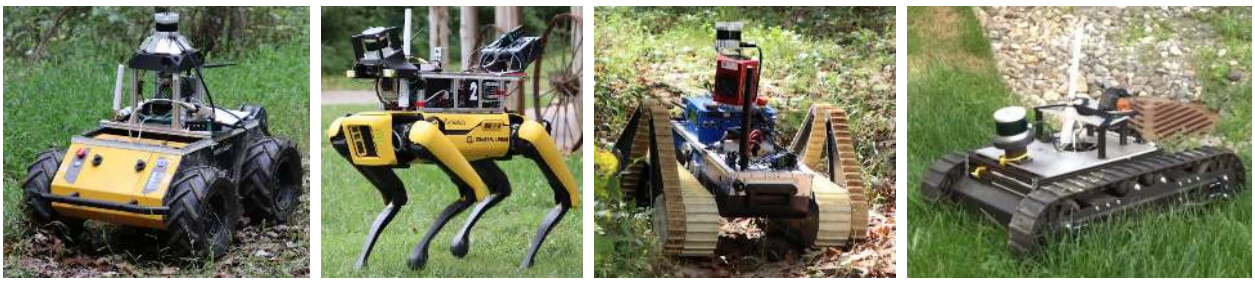
Figure 19: Selected parts of the environment with railways.

UGVs would pass such areas while being in the center of the tunnels. However, for the crossroads with the tracks, we have observed that the cost of traversing rail marked in Fig. 19a was, in some cases, wrongly estimated. It is caused by missing measurements in the geometrical model of the terrain. Thus, the robot's probability of getting stuck on the rail was increased.

- We have also observed that the localization system fails in long narrow corridors with the rails, such as the one shown in Fig. 19b. Since we were using only LiDAR data for the localization, which is insufficient for determining where the robot was in the such a monotonous tunnel.
- The next place found problematic is shown in Fig. 19c, where the robots need to get close to the marked red edge to identify a passable way to explore the next part of the station. The factor that played an important role in the probability that the robot would explore the whole station is the orientation of the robot because the used DTR robot has a slightly tilted sensory payload. Thus, it matters if the robot approaches the edge from its front or backside.
- The parts of the environments with high vertical drops and steep passages shown in Fig. 16c, Fig. 16f, and Fig. 16g were intentionally avoided by the UGVs since these parts are too risky for the ground robots or not passable at all.
- The environment shown in Fig. 16h, a virtual version of the Final circuit of the Systems track, has tighter corridors than the environments used for the preliminary testing. Because of that, the robots mostly assessed corridors after the first crossroad as unpassable and avoided exploring them. Tight corridors were addressed during the Systems track of the challenge using the finer resolution of the local maps.
- Based on the analysis, we have also realized that when robots meet within the environment, they occasionally mess up one of their maps by observing the other robot. We have addressed this issue by improving the ray tracing during fusing measurements into the map $\mathcal{M}_{3D}$.
- We have also observed minor imprecision in the simulations, such as a simulation of the robot tracks by a set of wheels or absolute rigidity of all the materials, which led our robots to get stuck at configurations that would be plausible in real-world environments.
- By accident, in the final submission, the artifact detections from the ground robots were not submitted in messages carrying artifacts. Thus, the UAVs provided all the artifacts reported by our team during the Final round of the Virtual track. UGVs still helped by building a communication network by dropping retranslation modules used by the UAVs to submit the detections.

4.2 Systems Track Competition

In the Final circuit of the Systems track, we have deployed a heterogeneous team of ground robots, including a wheeled robot Husky, three quadrupeds, and two tracked robots, as shown in Fig. 20. Three quadrupeds are Spot robots from Boston Dynamics, marked as Spot 1, 2, and 3. The tracked robot TRADR is the updated version of the tracked robot with flippers used in the previous circuits (Rouček et al., 2022).



(a) Wheeled robot Husky (b) Quadruped walking robot Spot (c) Tracked robot TRADR (d) Tracked robot Marmotte

Figure 20: Ground robots used during the Final circuit of the DARPA SubT Challenge.

The Spot robots, Husky, and TRADR robots were equipped with unified payloads that include high-bandwidth communication module Mobilicom MCU-30 Lite, low-bandwidth 868/915 MHz communication modules, the computational unit consisting of the Intel NUC with i7 CPU, 32 GB RAM, and 128-line Ouster OS0 LiDAR as the main sensor. The second tracked robot Marmotte is based on the HD2 platform by SuperDroid Robots. It used a similar payload to other robots except for the LiDAR and computational unit, LeiShen C16 and Dell 3070 Micro tower were utilized, respectively. All UGVs also used Nvidia Jetson AGX Xavier to speed-up artifact detection from the omni-vision RGB camera pack. The camera pack built for the UGVs consists of five Basler Ace 2 a2A1920-51gcPRO cameras with Basler Lens C125-0418-5M-P. The cameras were directed to the front, rear, sides, and up.

The human supervisor selected autonomous behaviors for each robot in the Systems track. The paths of the UGVs depicted in Fig. 21 indicate the environment explored during the mission. The robots of the combined UGVs and UAVs team explored 13 out of 28 sectors of the environment and found seven artifacts.

Table 8: Position error of the correctly detected artifacts during the Final circuit of the Systems track. One of the errors is marked by "*Guess*" meaning that the artifact was spotted from the video stream provided by the robot, and the human supervisor guessed the location of the artifact.

Artifact	Artifact type	Spot 1 [m]	Spot 2 [m]	Spot 3 [m]	Husky [m]	TRADR [m]	Marmotte [m]
L02	Vent	-	-	-	0.34	0.44	0.83
L32	Survivor	-	-	*Guess*	-	-	-
L51	Drill	0.67	2.57	-	0.39	0.08	-
L53	Backpack	1.87	4.05	-	-	2.11	-
L55	Rope	1.76	-	-	-	-	-
L58	Helmet	1.94	-	-	-	-	-

One of the seven artifacts was spotted only by a UAV, the artifact was a fire extinguisher, and it is marked in Fig. 21 as L31. The remaining six of the seven artifacts were each detected by at least one UGV, and two were detected by UAVs too. The position error of the artifacts detected by UGVs is summarized in Table 8. Artifacts not correctly detected are indicated in Fig. 21 as nearby artifacts. More specifically, the helmet artifact L62 was detected by Spot 2; however, the detection was provided with an error over 7 m since it was detected from a long distance of approx. 20 m. The drill L44 was detected by Spot 3 but not localized.

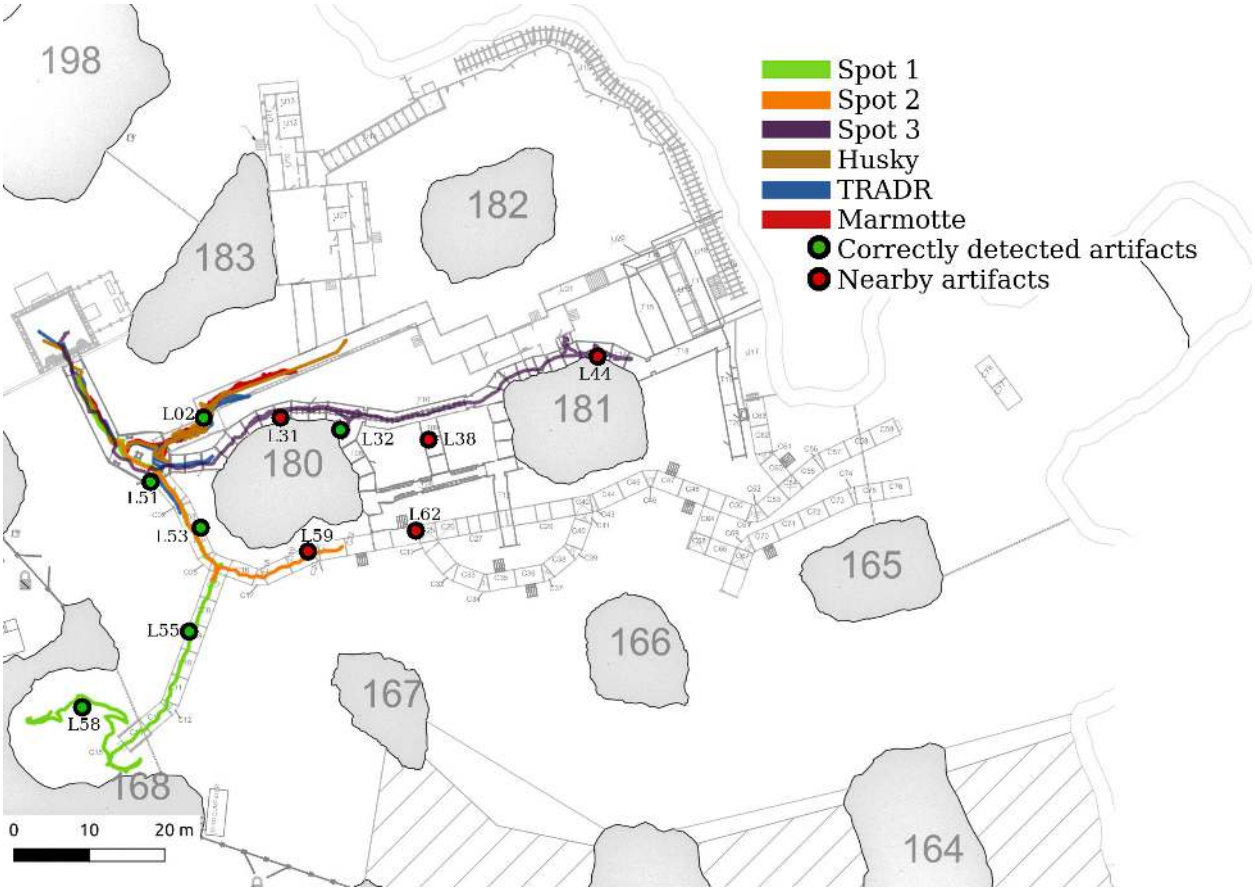


Figure 21: Artifacts detected by the ground robots and paths showing where the UGVs traversed through the Final circuit of the Systems track. The mapping error is captured in Fig. 23. The small red disks denote nearby not correctly detected artifacts.

Since the robot was sent to the mission area close to the mission end, the human supervisor did not have enough time to estimate the location of the artifact manually. The extinguisher L38 was visible by Spot 3 for a too short time to be correctly detected. Spot 2 passed close to the cellphone L59, but the cellphone not appeared in any field of view of the robot's cameras.

Table 9: Results from the Final circuit of the Systems track, including traveled path lengths, average robot's speed, the time elapsed before the robots were deployed, and the time during which the robots were active.

Robot	Spot 1	Spot 2	Spot 3	Husky	TRADR	Marmotte
Traveled path length [m]	96.50	55.30	156.60	124.35	88.00	141.80
Average speed [km h^{-1}]	0.90	1.04	1.12	0.43	0.43	0.18
Deployment start [s]	46	245	3097	760	1830	435
Active deployment duration [s]	386	191	503	1041	737	2836

The mission statistics of the traveled path lengths and average speeds of the UGVs are summarized in Table 9. The average speeds are calculated from the traveled distance and the difference between the first and last time the robot moved. The last two rows in Table 9 report the mission time when the robot was sent to the mission area and the duration the robot was active, respectively.

Spot 1 was sent to the area first, and the supervisor forced the robot to explore the cave-like environment. After the robot explored a corridor and most of a small cave, it unfortunately slipped and did not manage

to recover. The reason for the slip is related to the trade-off between avoiding exploring unknown parts of the environment and misclassifying a partially unknown environment. The trade-off is taken in situations when the geometrical model is incomplete or noisy because shadows or reflections occur in LiDAR data. Such incomplete data might lead to some parts of the environment being misclassified as untraversable and prematurely halt the exploration. In the competition run, the method was parametrized not to be conservative. Thus, in the case of missing data in the geometrical model, the terrain would more likely be misclassified as traversable, which has been considered desirable and expected to be corrected when the robot gets closer to the area with missing measurements. However, in the case of Spot 1, the misclassification caused the fall since the misclassification was corrected too late.

Spot 2 successfully explored another cave tunnel, but unfortunately, it experienced hardware failure, including the restart of the Nvidia Jetson AGX Xavier to which LiDAR was connected. Thus LiDAR data became unavailable, and the robot was inoperable by the end of the mission.

Initially, seventeen and a half minutes after the start, Spot 3 was sent into the course; however, after a few meters, it was observed that the data from the robot were not aligned with the coordinate frame given by the entry gate. Thus, after approximately one minute, the robot was teleoperated back by the human supervisor. While a part of the team in the staging area was trying to reinitialize Spot 3, the human supervisor managed the robots on the course. Spot 3 was successfully reinitialized in the last ten minutes of the mission, and it was sent to the course. Although it was sent in the last ten minutes, based on results reported in Table 9, it traveled the longest distance from all UGVs since it did not experience other failures and operated to the end of the mission.

The wheeled and tracked robots suffered from incorrect initialization, but both localization and initial coordinate frame estimation were correct, which can be seen in Figs. 22d to 22f. However, the frame used for filtration of the input range measurements cropping the sensory data was started with wrong headings, which led to cropping range measurements when robots reached a certain distance from the staging area. With such initialization, wheeled and tracked robots required teleoperation by the human supervisor. Teleoperating three robots at once by one human supervisor was impossible. Therefore, the human supervisor was commanding them and alternating the robots, which decreased the average speeds of the robots shown in Table 9.

Table 10: Map coverage and quality based on the localized joined local maps of the Final circuit of the Systems track.

Robot	Spot 1	Spot 2	Spot 3	Husky	TRADR	Marmotte	All robots
Map coverage [%]	14.8	5.3	13.3	8.4	9.2	7.8	22.6
Map outliers [%]	22.3	51.0	39.8	13.7	17.9	2.9	36.4
Map points	273.6 k	114.3 k	500.9 k	158.4 k	169.3 k	70.4 k	989.4 k

(a) Spot 1 (b) Spot 2 (c) Spot 3 (d) Husky (e) TRADR (f) Marmotte

Figure 22: Map coverage and outliers are visualized for each ground robot deployed in the Final circuit of the Systems track. Ground truth is black, inlier points are colored green, and outlier points are red.

Besides the traveled paths, we evaluate map coverage and map error by comparing joined localized local

maps with the ground truth (SubT Ground Truth Datasets (2021), 2022) of the environment using the tool provided by the DARPA SubT organizers (A. Schang, J. Rogers, and A. Maio, 2022). The map coverage is computed by identifying the percentage of the ground truth points closer than 1 m to the point of the joined local maps. The map error is computed as a percentage of outliers in the provided joined local map points. The outlier points are points further than 1 m from the closest ground truth point. The joined local maps for all robots are shown in Figs. 22 and 23, and the evaluation is quantified in Table 10. Fig. 22a indicates a map error in the cave at the end of the corridor passed by Spot 1, which is induced by the fall of the robot. From Fig. 22b, the localization of Spot 2 drifted in the z-axis, causing many map outliers quantified in Table 10.

Figure 23: Map coverage and outliers visualized for joined local maps received from all the robots deployed in the Final circuit. Ground truth is black, inlier points are colored green, and outlier points are red.

4.2.1 Lessons Learned from the Final Circuit of the Systems Track

- The sensory payloads of the ground robots, except the Marmotte, are connected to the Nvidia Jetson AGX Xavier and Intel NUC. The LiDAR data are transferred into the Nvidia Jetson AGX Xavier using an external network adapter connected to the development kit and from the Nvidia Jetson AGX Xavier, then into the Intel NUC, where the LiDAR data are processed for localization and autonomy. Thus, if one of the components in the connection chain failed, data were not transferred from LiDAR to the Intel NUC. During the testing before deploying the robots in the SubT Finals, we detected that the external network adapter was sensitive to vibrations, which was addressed by the mechanical holder of the adapter, which decreased the probability that the adapter would disconnect. During the Final circuit deployment, the payloads of Spot 3, Husky, and TRADR were working reliably, but the payload of Spot 2 failed relatively frequently. We further observed that the utilized development kit is sensitive to changes in the input voltage, and the used power connectors may temporarily disconnect when exposed to high vibrations.
- When Spot 1 fell inside the cave, the human supervisor was still able to connect to the robot and trigger an automatic recovery procedure to allow the robot to move again. Unfortunately, the recovery procedure cannot directly access low-level controllers of the Spot's motors. Thus, it was

unable to recover the robot from all types of faults, including the induced fault in the cave. For the tracked robot TRADR, we had full access to the low-level controllers, allowing us to directly reset specific controllers when there was a hardware fault, leaving less space for unrecoverable faults.

- Using the shared topometrical information only for coordination and not for extending the planning graph was beneficial, especially during testing when the precise synchronization of the coordinate frames was unavailable. Then, the planning graph was not damaged by adding wrongly placed vertices generated by other robots, but the knowledge of the topometrical maps of other robots helped the coordination.
- Based on our previous experience with tuning map parameters for the Virtual track competition, we have increased the resolution parameter of the local maps from 7 cm to 4.5 cm for quadrupeds that were expected to go through the narrowest corridors and to 5 cm for the rest of the robots. The increased resolution allowed the robots to explore narrow corridors of the environment, including cave-like tunnels in the Final circuit of the Systems track.
- We have also improved the implementation of the local mapping by precomputing the planning graph so that the legged robots were able to replan faster. Therefore, the maximum exploring speed could increase above $1\,\mathrm{m\,s^{-1}}$. However, we limited the maximum speed to $0.7\,\mathrm{m\,s^{-1}}$ because the artifacts were not reliably detected when robots walked faster than $0.7\,\mathrm{m\,s^{-1}}$.
- The initialization of the localization and initial coordinate frame estimation was visually verified by observing the transformed local map before sending the robot to the course in the Final circuit. In the Post-event testing, the verification procedure was violated, leading to the offset shown in Fig. 25c.

4.3 Post-event Testing

Since the human supervisor can hugely affect the autonomous exploration of the environment in the Final circuit, we conducted fully autonomous mission deployment during the Post-event testing event. The exploration method was deployed with the same parametrization as for the competition run of the Systems track. The differences were that the system was operated in fully autonomous mode (without the human supervisor), and three quadruped robots, as the robots with the best traversal capabilities, were used. And the mission time was set to 1000 s instead of one hour, which is roughly about a quarter of the competition mission time.

Since both communication systems allow sharing of the K3 messages, it is possible to use just one system to coordinate the robots. Therefore, only the low-bandwidth communication system has been utilized to examine whether it would be sufficient for coordinating the robots. However, the retranslation modules for the low-bandwidth communication were placed prior to the deployment of the robots. The module placement improved the message delivery's reliability by reducing the counts of communication modules close to each other, which is the case of the modules held by the robot in the tray to be used as breadcrumbs during the mission. Since the camera streams are not available with the low-bandwidth communication system, we focus on examining the exploration and not detecting the artifacts. Thus, the robots explored and coordinated fully autonomously, using only a low-bandwidth communication system.

The paths taken by the robots are visualized in Fig. 24, from which it can be seen that all the robots took different paths at the first crossroad, which is induced by the proposed coordination method. Because the initial alignment of the first deployed robot (Spot 3) was slightly off (see Fig. 25), the part of the rank $G_{i,cross}$ penalized waypoints corresponding to the two farthest corridors from the staging area. Because of the penalization, the second robot chose the corridor closest to the staging area at the first crossroad. Therefore, the last deployed Spot selected the remaining corridor.

If the initial alignment of Spot 3 would be correct, the part of the rank $G_{i,cross}$ would be similar for all waypoints at the first meters of the corridors going from the first crossroad since Spot 3 visited the beginnings of the two corridors close to the staging area before choosing the farthest corridor. However, in

Figure 24: Paths taken by the quadruped robots in the fully autonomous mission of the Post-event testing scenario. The mapping error is captured in Fig. 26.

such a hypothetical case, the decisions of the two following robots at the first crossroad would be the same because the part of the rank $G_{i,spread}$ would make corridors closer to the staging area more attractive for Spots 1 and 2. The waypoints corresponding to the corridors close to the staging area would be the farthest from the latest reported position of Spot 3.

The importance of the topometrical map for the planning can be further seen from the path of Spot 1, shown in Fig. 24. The robot managed to return to the previously visited crossroads and continue exploring different corridors. The range of the local map is lower than the length of the corridors from the crossroads. Thus, using only the local map, the robot would not have sufficient information to choose the next unexplored corridor from the crossroad.

Table 11: Path lengths and robot speeds measured during the Post-event testing in the Final circuit environment.

Robot	Spot 1	Spot 2	Spot 3
Traveled path length [m]	243.10	125.50	191.70
Average speeds [$\mathrm{km\,h^{-1}}$]	1.15	1.19	1.08
Deployment start [s]	239	137	37
Deployment duration [s]	761	380	639

Fig. 24 further suggests that Spot 1 visited most of the sectors of the tunnel part of the environment. Spot 2 explored fewer sectors of the environment since it experienced a hardware fault of the payload. A localization

fault caused local map distortion that ended the exploration of Spot 3. Although there were faults related to hardware and localization, the robots managed to correctly spread at the first crossroad and together traverse over five hundred meters while achieving the average speeds summarized in Table 11.

(a) Spot 1 (b) Spot 2 (c) Spot 3

Figure 25: Map coverage and outliers visualized for each robot deployed in the Post-event testing. Ground truth is black, inlier points are colored green, and outlier points are red.

Table 12: Map coverage and quality based on localized joined local maps during the Post-event testing.

Robot	Spot 1	Spot 2	Spot 3	All robots
Map coverage [%]	23.6	15.2	7.8	30.3
Map outliers [%]	13.6	10.1	67.5	32.6
Map points	550.1 k	387.3 k	440.1 k	1177.1 k

Map coverage and outliers can be seen in Figs. 25 and 26. The initialization of Spot 3 was not visually verified, which led to the high amount of outlier points, which is visible in Fig. 25c and quantified in Table 12.

Comparing quantified results in Table 10 and Table 12, we can see that the percentage of the map outlier points is roughly similar. It is because, in both cases, some of the Spot robots provided maps with a high amount of outliers, increasing the overall scores. Although the experiment during the Post-event testing took only 1000 s, the total distance traveled by the UGVs and map coverage is competitive to the deployment of the UGVs during the competition run.

Here, it is important to comment that in the Post-event test, a significant amount of time was saved by fully autonomous systems without interventions of the human supervisor commanding the robots for artifact detection and, thus, scoring. Such interventions include maintaining the high-bandwidth communication connection by stopping robots at the crossroads or sending robots to places where they may improve the estimate of the artifact location observed by another robot. Moreover, in the competition run, a part of the mission time was spent deploying the UAVs, leading to increased artifact scores.

Figure 26: Map coverage and error visualized for joined local maps received from all the robots deployed in the Post-event testing in the Final circuit environment. Ground truth is black, inlier points are colored green, and outlier points are red.

5 Conclusion

This report presents the description and field evaluation results of the developed framework for exploring underground communication-restricted environments with a heterogeneous multi-robot team. The feasibility of the developed solution has been examined within the DARPA SubT Challenge. The proposed method was further employed by the team CTU-CRAS-NORLAB in the field deployment within the Final circuit of the Systems track. Besides, the method deployed on the UGVs support achieving second place in the Final circuit of the Virtual track, showing the possibility of using the method in large-scale environments. In addition to examination of the developed framework in the Systems track under human supervision, it was also field-tested in the DARPA Post-event testing, where the developed method was deployed on three quadrupeds enabling them to fully autonomously explore the environment while using only a low-bandwidth communication system for coordination.

During the Systems track deployments, besides the technical issues related to the deployment of the framework, there were a few limitations that are a motivation for further research and improvements of the developed framework. The first observed limitation is the trade-off between avoiding exploring unknown parts of the environment and misclassifying a partially unknown environment. The second limitation is in bounding the local map resolution to the ability of the framework to plan through narrow corridors, which does not allow usage of the proposed solution in environments where the width of the corridors is close to the width of the robots. On the other hand, the framework was capable of real-time operation in complicated environments, supporting usage by a human supervisor and commanding through multiple communication systems in parallel.

Acknowledgments

We would like to thank rest of the members of the CTU-CRAS-NORLAB team who participated in the design and development of the UGV and UAV hardware platforms, simulations, testing and other support. Namely: Ruslan Agishev, Afzal Ahmad, Teymur Azayev, Tomáš Báča, Tommy Bouchard-Lebrun, Simon-Pierre Deschenes, Olivier Gamache, Alexandre Guénette, Daniel Heřt, Bedřich Himmel, Jakub Janoušek, Tomáš Krajník, Vít Krátký, Vladimír Kubelka, Tomáš Musil, Denis Ouellet, Martin Pecka, Pavel Petráček, Tomáš Petříček, Matěj Petrlík, Francois Pomerleau, Miloš Prágr, Tomáš Rouček, Vojtěch Šalanský, Martin Saska, Martin Škarytka, Vojtěch Spurný, Yurii Stasinchuk, Pavel Stoudek, Tomáš Svoboda, Arsen Tkachev, Maxime Vaidis, Matouš Vrba, Volodymyr Zabulskyi, Karel Zimmermann, and Martin Zoula.

The presented work has been supported by DARPA grant no. HR00112190014, - SUBTERRANEAN (SUBT) CHALLENGE - PHASE 3 and the Czech Science Foundation (GAČR) under research projects No. 19-20238S and No. 20-29531S.

References

A. Schang, J. Rogers, and A. Maio (accessed Oct 7, 2022). Map analysis (version 1.0) [source code]. `https://github.com/subtchallenge/map_analysis`.

Agha, A., Otsu, K., Morrell, B., Fan, D. D., Thakker, R., Santamaria-Navarro, A., Kim, S.-K., Bouman, A., Lei, X., Edlund, J., Ginting, M. F., Ebadi, K., Anderson, M., Pailevanian, T., Terry, E., Wolf, M., Tagliabue, A., Vaquero, T. S., Palieri, M., Tepsuporn, S., Chang, Y., Kalantari, A., Chavez, F., Lopez, B., Funabiki, N., Miles, G., Touma, T., Buscicchio, A., Tordesillas, J., Alatur, N., Nash, J., Walsh, W., Jung, S., H., L., Kanellakis, C., Mayo, J., Harper, S., Kaufmann, M., Dixit, A., Correa, G. J., Lee, C., Gao, J., Merewether, G., Maldonado-Contreras, J., Salhotra, G., Da Silva, M. S., Ramtoula, B., Fakoorian, S., Hatteland, A., Kim, T., Bartlett, T., Stephens, A., Kim, L., Bergh, C., Heiden, E., Lew, T., Cauligi, A., Heywood, T., Kramer, A., Leopold, H. A., Melikyan, H., Choi, H. C., Daftry, S., Toupet, O., Wee, I., Thakur, A., Feras, M., Beltrame, G., Nikolakopoulos, G., Shim, D., Carlone, L., and Burdick, J. (2022). Nebula: Team costar's robotic autonomy solution that won phase ii of darpa subterranean challenge. *Field Robotics*, pages 1432–1506.

Amigoni, F., Banfi, J., and Basilico, N. (2017). Multirobot Exploration of Communication-Restricted Environments: A Survey. *IEEE Intelligent Systems*, 32(6):48–57.

Amigoni, F. and Caglioti, V. (2010). An information-based exploration strategy for environment mapping with mobile robots. *Robotics and Autonomous Systems*, 58(5):684–699.

Balakirsky, S., Carpin, S., Kleiner, A., Lewis, M., Visser, A., Wang, J., and Ziparo, V. A. (2007). Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue. *Journal of Field Robotics*, 24(11–12):943–967.

Bautin, A., Simonin, O., and Charpillet, F. (2012). Minpos : A novel frontier allocation algorithm for multi-robot exploration. In *International Conference on Intelligent Robotics and Applications (ICIRA)*, pages 496–508.

Bayer, J. and Faigl, J. (2021a). Decentralized task allocation in multi-robot exploration with position sharing only. In *International Symposium on Swarm Behavior and Bio-Inspired Robotics (SWARM)*.

Bayer, J. and Faigl, J. (2021b). Decentralized topological mapping for multi-robot autonomous exploration under low-bandwidth communication. In *European Conference on Mobile Robots (ECMR)*.

Benkrid, A., Benallegue, A., and Achour, N. (2019). Multi-robot coordination for energy-efficient exploration. *Journal of Control, Automation and Electrical Systems*, 30(6):911–920.

Benton, J., Coles, A., and Coles, A. (2012). Temporal planning with preferences and time-dependent continuous costs. In *Twenty-Second International Conference on Automated Planning and Scheduling*.

Biber, P. and Strasser, W. (2003). The normal distributions transform: a new approach to laser scan matching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2743–2748.

Bouman, A., Ginting, M. F., Alatur, N., Palieri, M., Fan, D. D., Touma, T., Pailevanian, T., Kim, S.-K., Otsu, K., Burdick, J., et al. (2020). Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2518–2525.

Burgard, W., Moors, M., Fox, D., Simmons, R., and Thrun, S. (2000). Collaborative multi-robot exploration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 476–481.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. E. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376–386.

Butzke, J., Daniilidis, K., Kushleyev, A., Lee, D. D., Likhachev, M., Phillips, C., and Phillips, M. (2012). The university of pennsylvania magic 2010 multi-robot unmanned vehicle system. *Journal of Field Robotics*, 29(5):745–761.

Chung, T. (accessed Mar 10, 2022). DARPA Subterranean (SubT) Challenge. `https://www.darpa.mil/program/darpa-subterranean-challenge`.

Dang, T., Tranzatto, M., Khattak, S., Mascarich, F., Alexis, K., and Hutter, M. (2020). Graph-based subterranean exploration path planning using aerial and legged robots. *Journal of Field Robotics*, 37(8):1363–1388.

Delmerico, J., Mintchev, S., Giusti, A., Gromov, B., Melo, K., Horvat, T., Cadena, C., Hutter, M., Ijspeert, A., Floreano, D., et al. (2019). The current state and future outlook of rescue robotics. *Journal of Field Robotics*, 36(7):1171–1191.

Deschênes, S.-P., Baril, D., Kubelka, V., Giguère, P., and Pomerleau, F. (2021). Lidar scan registration robust to extreme motions. In *Conference on Robots and Vision (CRV)*, pages 17–24.

Dezső, B., Jüttner, A., and Kovács, P. (2011). Lemon–an open source c++ graph template library. *Electronic Notes in Theoretical Computer Science*, 264(5):23–45.

Faigl, J. and Kulich, M. (2013). On determination of goal candidates in frontier-based multi-robot exploration. In *European Conference on Mobile Robots (ECMR)*, pages 210–215.

Faigl, J. and Kulich, M. (2015). On benchmarking of frontier-based multi-robot exploration strategies. In *European Conference on Mobile Robots (ECMR)*, pages 1–8.

Faigl, J., Kulich, M., and Přeučil, L. (2012). Goal assignment using distance cost in multi-robot exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3741–3746.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33.

Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., and Stewart, B. (2006). Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, 94(7):1325–1339.

Fox, M. and Long, D. (2002). Pddl+: Modeling continuous time dependent effects. In *Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space*, volume 4, page 34.

González-Banos, H. H. and Latombe, J.-C. (2002). Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research*, 21(10-11):829–848.

Heppner, G., Roennau, A., and Dillman, R. (2013). Enhancing sensor capabilities of walking robots through cooperative exploration with aerial robots. *Journal of Automation Mobile Robotics and Intelligent Systems*, 7.

Hudson, N., Talbot, F., Cox, M., Williams, J., Hines, T., Pitt, A., Wood, B., Frousheger, D., Surdo, K., Molnar, T., Steindl, R., Wildie, M., Sa, I., Kottege, N., Stepanas, K., Hernandez, E., Catt, G., Docherty, W., Tidd, B., Tam, B., Murrell, S., Bessell, M., Hanson, L., Tychsen-Smith, L., Suzuki, H., Overs, L., Kendoul, F., Wagner, G., Palmer, D., Milani, P., O'Brien, M., Jiang, S., Chen, S., and Arkin, R. (2022). Heterogeneous ground and air platforms, homogeneous sensing: Team csiro data61's approach to the darpa subterranean challenge. *Field Robotics*, pages 595–636.

Husain, A., Jones, H., Kannan, B., Wong, U., Pimentel, T., Tang, S., Daftry, S., Huber, S., and Whittaker, W. L. (2013). Mapping planetary caves with an autonomous, heterogeneous robot team. In *IEEE Aerospace Conference*, pages 1–13.

Jain, U., Tiwari, R., and Godfrey, W. W. (2017). Comparative study of frontier based exploration methods. In *2017 Conference on Information and Communication Technology (CICT)*, pages 1–5. IEEE.

Katz, S. and Tal, A. (2015). On the visibility of point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1350–1358.

Kolvenbach, H., Valsecchi, G., Grandia, R., Ruiz, A., Jenelten, F., and Hutter, M. (2019). Tactile inspection of concrete deterioration in sewers with legged robots. In *Conference on Field and Service Robotics (FSR)*.

Lacaze, A., Murphy, K., Del Giorno, M., and Corley, K. (2012). Reconnaissance and autonomy for small robots (rasr) team: Magic 2010 challenge. *Journal of Field Robotics*, 29(5):729–744.

Miller, I. D., Cladera, F., Cowley, A., Shivakumar, S. S., Lee, E. S., Jarin-Lipschitz, L., Bhat, A., Rodrigues, N., Zhou, A., Cohen, A., Kulkarni, A., Laney, J., Taylor, C. J., and Kumar, V. (2020). Mine tunnel exploration using multiple quadrupedal robots. *IEEE Robotics and Automation Letters*, 5(2):2840–2847.

Murphy, R. R. (2014). *Disaster robotics*. MIT press.

Ohradzansky, M., Rush, E., Riley, D., Mills, A., Ahmad, S., McGuire, S., Biggie, H., Harlow, K., Miles, M., Frew, E., Heckman, C., and Humbert, J. (2022). Multi-agent autonomy: Advancements and challenges in subterranean exploration. *Field Robotics*, pages 1068–1104.

Olson, E. (2011). AprilTag: A Robust and Flexible Visual Fiducial System. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407.

Olson, E., Strom, J., Morton, R., Richardson, A., Ranganathan, P., Goeddel, R., Bulic, M., Crossman, J., and Marinier, B. (2012). Progress toward multi-robot reconnaissance and the magic 2010 competition. *Journal of Field Robotics*, 29(5):762–792.

Petráček, P., Krátký, V., Petrlík, M., Báča, T., Kratochvíl, R., and Saska, M. (2021). Large-scale exploration of cave environments by unmanned aerial vehicles. *IEEE Robotics and Automation Letters*, 6(4):7596–7603.

Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. (2013). Comparing ICP Variants on Real-World Data Sets. *Autonomous Robots*, 34(3):133–148.

Rouček, T., Pecka, M., Čížek, P., Petříček, T., Bayer, J., Šalanský, V., Azayev, T., Heřt, D., Petrlík, M., Báča, T., Spurný, V., Krátký, V., Petráček, P., Baril, D., M., V., Kubelka, V., Pomerleau, F., Faigl, J., Zimmermann, K., Saska, M., Svoboda, T., and Krajník, T. (2022). System for multi-robotic exploration of underground environments ctu-cras-norlab in the darpa subterranean challenge. *Field Robotics*, 2:1779–1818.

Saboia, M., Clark, L., Thangavelu, V., Edlund, J. A., Otsu, K., Correa, G. J., Varadharajan, V. S., Santamaria-Navarro, A., Touma, T., Bouman, A., et al. (2022). Achord: Communication-aware multi-robot coordination with intermittent connectivity. *arXiv preprint arXiv:2206.02245*.

Scherer, S., Agrawal, V., Best, G., C., C., Cujic, K., Darnley, R., DeBortoli, R., Dexheimer, E., Drozd, B., Garg, R., Higgins, I., Keller, J., Kohanbash, D., Nogueira, L., Pradhan, R., Tatum, M., Viswanathan, V., Willits, S., Zhao, S., Zhu, H., Abad, D., Angert, T., Armstrong, G., Boirum, R., Dongare, A., Dworman, M., Hu, S., Jaekel, J., Ji, R., Lai, A., Lee, Y., Luong, A., J., M., Maier, J., Picard, J., Pluckter, K., Saba, A., Saroya, M., Scheide, E., Shoemaker-Trejo, N., Spisak, J., Teza, J., Yang, F., Wilson, A., Zhang, H., Choset, H., Kaess, M., Rowe, A., Singh, S., Zhang, J., Hollinger, G., and Travers, M. (2022). Resilient and modular subterranean exploration with a team of roving and flying robots. *Field Robotics*, 2:678–734.

Schulz, C., Hanten, R., Reisenauer, M., and Zell, A. (2019). Simultaneous collaborative mapping based on low-bandwidth communication. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 413–414.

Schwarz, M., Beul, M., Droeschel, D., Schüller, S., Periyasamy, A. S., Lenz, C., Schreiber, M., and Behnke, S. (2016). Supervised Autonomy for Exploration and Mobile Manipulation in Rough Terrain with a Centaur-Like Robot. *Frontiers in Robotics and AI*, 3.

Stelzer, A., Hirschmüller, H., and Görner, M. (2012). Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain. *International Journal of Robotics Research*, 31(4):381–402.

SubT Ground Truth Datasets (2021) (accessed Oct 7, 2022). `https://github.com/subtchallenge/subt_resources`.

Thrun, S., Hahnel, D., Ferguson, D., Montemerlo, M., Triebel, R., Burgard, W., Baker, C., Omohundro, Z., Thayer, S., and Whittaker, W. (2003). A system for volumetric robotic mapping of abandoned mines. In *International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 3, pages 4270–4275.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006). Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692.

Tranzatto, M., Dharmadhikari, M., Bernreiter, L., Camurri, M., Khattak, S., Mascarich, F., Pfreundschuh, P., Wisth, D., Zimmermann, S., Kulkarni, M., et al. (2022a). Team cerberus wins the darpa subterranean challenge: Technical overview and lessons learned. *arXiv preprint arXiv:2207.04914*.

Tranzatto, M., Mascarich, F., Bernreiter, L., Godinho, C., Camurri, M., Khattak, S., Dang, T., Reijgwart, V., Löje, J., Wisth, D., Zimmermann, S., Nguyen, H., Fehr, M., Solanka, L., Buchanan, R., Bjelonic, M., Khedekar, N., Valceschini, M., Jenelten, F., Dharmadhikari, M., Homberger, T., De Petris, P., Wellhausen, L., Kulkarni, M., Miki, T., Hirsch, S., Montenegro, M., Papachristos, C., Tresoldi, F., Carius, J., Valsecchi, G., Lee, J., Meyer, K., Wu, X., Nieto, J., Smith, A., Hutter, M., Siegwart, R., Mueller, M., Fallon, M., and Alexis, K. (2022b). Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge. *Field Robotics*, pages 274–324.

Tranzatto, M., Miki, T., Dharmadhikari, M., Bernreiter, L., Kulkarni, M., Mascarich, F., Andersson, O., Khattak, S., Hutter, M., Siegwart, R., et al. (2022c). Cerberus in the darpa subterranean challenge. *Science Robotics*, 7(66):eabp9742.

Wermelinger, M., Fankhauser, P., Diethelm, R., Krüsi, P., Siegwart, R., and Hutter, M. (2016). Navigation planning for legged robots in challenging terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1184–1189.

Williams, J., Jiang, S., O'Brien, M., Wagner, G., Hernandez, E., Cox, M., Pitt, A., Arkin, R., and Hudson, N. (2020). Online 3d frontier-based ugv and uav exploration using direct point cloud visibility. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 263–270.

Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 146–151.

Yamauchi, B. (1998). Frontier-based exploration using multiple robots. In *Second International Conference on Autonomous Agents*, pages 47–53. Association for Computing Machinery.