arXiv:2207.04914v1 [cs.RO] 11 Jul 2022

Team CERBERUS Wins the DARPA Subterranean Challenge: Technical Overview and Lessons Learned

Marco Tranzatto[1 *] **Mihir Dharmadhikari**[2] **Lukas Bernreiter**[3] **Marco Camurri**[4]

Shehryar Khattak[1] **Frank Mascarich**[2] **Patrick Pfreundschuh**[3] **David Wisth**[4]

Samuel Zimmermann[1] **Mihir Kulkarni**[2] **Victor Reijgwart**[3] **Benoit Casseau**[4]

Timon Homberger[1] **Paolo De Petris**[2] **Lionel Ott**[3] **Wayne Tubby**[4] **Gabriel Waibel**[1]

Huan Nguyen[2] **Cesar Cadena**[3] **Russell Buchanan**[4] **Lorenz Wellhausen**[1]

Nikhil Khedekar[2] **Olov Andersson**[3] **Lintong Zhang**[4] **Takahiro Miki**[1] **Tung Dang**[2]

Matias Mattamala[4] **Markus Montenegro**[1] **Konrad Meyer**[1] **Xiangyu Wu**[5]

Adrien Briod[6] **Mark Mueller**[5] **Maurice Fallon**[4] **Roland Siegwart**[3] **Marco Hutter**[1]

Kostas Alexis[2]

Abstract

This article presents the CERBERUS robotic system-of-systems, which won the DARPA Subterranean Challenge Final Event in 2021. The Subterranean Challenge was organized by DARPA with the vision to facilitate the novel technologies necessary to reliably explore diverse underground environments despite the grueling challenges they present for robotic autonomy. Due to their geometric complexity, degraded perceptual conditions combined with lack of GPS support, austere navigation conditions, and denied communications, subterranean settings render autonomous operations particularly demanding. In response to this challenge, we developed the CERBERUS system which exploits the synergy of legged and flying robots, coupled with robust control especially for overcoming perilous terrain, multi-modal and multi-robot perception for localization and mapping in conditions of sensor degradation, and resilient autonomy through unified exploration path planning and local motion planning that reflects robot-specific limitations. Based on its ability to explore diverse underground environments and its high-level command and control by a single human supervisor, CERBERUS demonstrated efficient exploration, reliable detection of objects of interest, and accurate mapping. In this article, we report results from both the preliminary runs and the final Prize Round of the DARPA Subterranean Challenge, and discuss highlights and challenges faced, alongside lessons learned for the benefit of the community.

*[1] Robotic Systems Lab, ETH Zurich, [2] Autonomous Robots Lab, Norwegian University of Science and Technology — University of Nevada, Reno, [3] Autonomous Systems Lab, ETH Zurich, 4 Oxford Robotics Institute, University of Oxford, [5] Flyability, [6] University of California, Berkeley. Direct correspondence to Marco Tranzatto `marcot@ethz.ch`

1 Introduction

This paper reports the research activities and technological progress made by team "CERBERUS"[1] towards our winning participation in the Final Event of the DARPA Subterranean (SubT) Challenge. The SubT Challenge was a three year-long, $82 million [Montgomery, 2021] robotics competition organized and coordinated by the US Defense Advanced Research Projects Agency (DARPA). This challenge aimed to accelerate the research and development of novel technologies to support operations in complex and diverse underground settings, presenting significant challenges for military and civilian first responders. Operating in these environments can often be dangerous for humans; therefore, robotic systems are a viable alternative to provide rapid situational awareness to a small team of operators before possible entry into such unknown, high-risk, and dynamic subterranean areas. Possible use-case scenarios for these new technologies include - but are not limited to - search and rescue missions in collapsed mines, post-earthquake urban settings, and cave networks. In response to these challenges, we created CERBERUS: a system-of-systems involving walking and flying robots equipped with multi-modal perception capabilities, navigation and mapping autonomy, and self-deployed network communication-extender modules that could autonomously explore complex subterranean environments.

The SubT Challenge featured two competition tracks: Systems and Virtual. Systems track teams deployed their hardware and software solutions to compete in physical subterranean courses. In contrast, Virtual track teams developed software-only solutions evaluated in simulation. The underground scenarios for the Systems track included a) human-made tunnel networks extending for several kilometers and presenting multiple branches and vertical openings ("Tunnel Circuit", August 2019), b) multi-level urban underground structures with complex multi-storey layouts ("Urban Circuit", February 2020), and c) natural cave environments with complex, diverse morphologies and constrained passages ("Cave Circuit", August 2020, canceled event). Finally, a comprehensive benchmark mission combined all the challenges of the previous circuits ("Final Event", September 2021). The SubT Challenge brought together more than 300 competitors from 20 different teams, composed of individuals, startups, universities, and large companies that spanned 11 countries and 20 universities.

Alongside the particular operational goal of robotic subterranean deployments, DARPA initiated the SubT Challenge to stimulate development in four critical technical areas: Autonomy, Perception, Networking, and Mobility. Robotic systems were expected to show a high level of resilient autonomy, demonstrating the ability to map, navigate, and search in complex and dynamic environments with little to no human interventions (Autonomy). Additionally, they had to cope with grueling navigation challenges, operating under varying and degraded conditions including dust, fog, mist, water, smoke, darkness, obscured, and/or scattering environments (Perception). Moreover, the robots had to send information to the Human Supervisor while dealing with lack of line-of-sight, the effects of varying geological conditions, and wireless signal propagation challenges in subterranean environments (Networking). Finally, the systems were tasked to navigate mobility-stressing and dynamic terrain, including narrow passages, sharp turns, large drops/climbs, inclines, steps, falling debris, mud, sand, and water (Mobility). Competing teams were tasked with exploring and mapping unknown underground areas, with the additional requirement that only one team member, the Human Supervisor, was allowed to manage and interact with the deployed robots. Teams were scored based on their ability to localize specified objects, called "artifacts", representing features or items of interest applicable to subterranean settings. Artifacts for the Prize Round of the Final Event included human (mannequin) survivors, cellphones, backpacks, drills, fire extinguishers, vents, gas-filled rooms, helmets, ropes, and a SubT cube. The artifacts' surveyed positions were expressed with respect to a DARPA-defined coordinate frame. Each team had 30 min before the run to set up their Base Station (a computer used by the Human Supervisor to control the robotic agents) and for four team personnel (known as Pit Crew members) to set up and prepare the robots before the beginning of the mission. After the setup time, every team had 60 min to find as many of the 40 artifacts distributed along the course as possible, while having a limited number of available report attempts (to discourage false positives). A point was earned for each report that correctly identified an artifact's class and its position within 5 m of the object's ground-truth location.

[1] https://www.subt-cerberus.org/

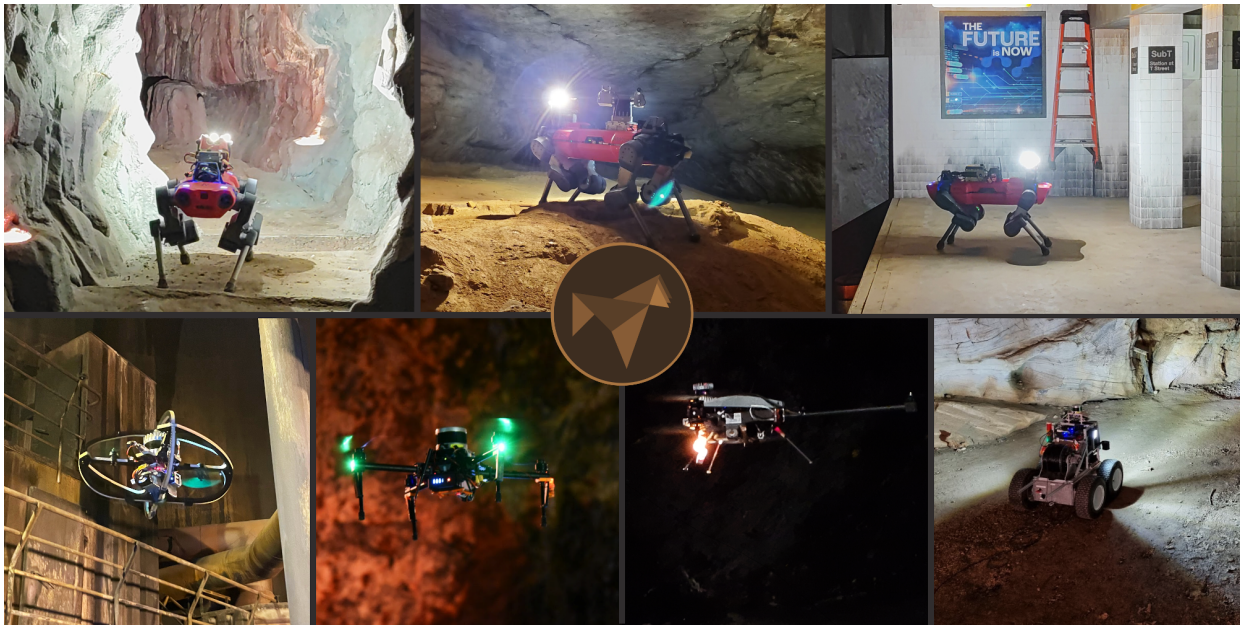


Figure 1: The types of robots developed by Team CERBERUS used for the DARPA SubT Challenge. Upper row: ANYmal C robots. Lower row (left to right): RMF-Owl, Alpha Aerial Scout, and Kolibri flying robots, alongside a tethered roving platform acting as communication hub for the other robots. All robots are presented in diverse subterranean environments. A video showing the Team's Prize Round is available at `https://youtu.be/QON8IFc8cjE`

Team CERBERUS, which stands for "CollaborativE walking & flying RoBots for autonomous ExploRation in Underground Settings," draws its name from Greek mythology where Cerberus was the three-headed dog guarding the gates of the underworld. The CERBERUS system-of-systems comprised a composite system of legged and aerial robots. Each agent featured our autonomy and multi-modal perception solutions. The legged agents, in addition, could deploy communication-extender modules to increase the wireless network range. Together, these features enabled resilient navigation, exploration, mapping, and search within complex, large-scale, sensor-degraded, and communications-denied subterranean environments. Moreover, a tethered roving platform assisted the team and acted as a communication hub. Figure 1 presents most of the types of robots Team CERBERUS developed for the SubT Challenge's Final Event. To gather the necessary expertise, Team CERBERUS involved an international partnership of the University of Nevada Reno, ETH Zurich, Norwegian University of Science and Technology, University of California Berkeley, University of Oxford, Flyability and Sierra Nevada Corporation. Team CERBERUS competed in the System Track of the SubT Challenge and was one of the DARPA-funded teams since the beginning of the competition. On September 23 2021, Team CERBERUS won the Prize Round of the Final Event of the DARPA Subterranean Challenge after three years of intensive developments and an exciting competition among top performers.

The remainder of this paper is structured as follows: related work is outlined in section 2; a detailed description of the CERBERUS robot and autonomy technologies is provided in section 3 followed by a comprehensive overview of the results during the DARPA Subterranean Challenge Final Event in section 4; lessons learned are outlined in section 5, while conclusions are drawn in section 6.

2 Related work

In order for our work to be put into perspective, in this section we attempt to summarize notable contributions in subterranean robotics and autonomy developed both before and outside of the DARPA SubT Challenge,

as well as some of work of other competing teams based on the currently available literature.

2.1 Autonomous systems in subterranean environments and related settings

Subterranean settings have long been enticing to the robotics community due to the uniqueness of the environments and the operating conditions [Murphy et al., 2009, Tardioli et al., 2019]. One of the main conclusions from the survey in [Murphy et al., 2009] was that robots with navigation, localization, and mapping capabilities developed specifically for the underground scenarios were required. The work in [Novák et al., 2017] described the concept of a reconnaissance mobile robot for inspecting underground coal mines including the motion, sensing, mapping, and communication modules. [Loupos et al., 2018] presented a robotic platform consisting of a robotized boom lift, a high precision robotic arm, advanced computer vision systems, a 3D laser scanner, and an ultrasonic sensor for autonomous tunnel inspection. Additionally, the authors in [Colas et al., 2013] proposed a path planning method to allow ground robots to navigate in 3D environments and demonstrated stair climbing. Departing from the need to fully reconstruct 3D maps of the environments, several works explored other directions to allow the robots to autonomously explore underground settings. The authors in [Silver et al., 2006] described a system based on the Groundhog robot platform which was capable of autonomous topological exploration of mine environments. On the other hand, the contribution in [Mansouri et al., 2019] presented a Micro Aerial Vehicle (MAV) system that used darkness contour detection to command its heading angle based on the direction of the underground tunnels. Another contribution presented in [Özaslan et al., 2016, Özaslan et al., 2017] described an MAV system that could fly along the center-line of tunnel-like dam penstocks using range and vision sensors. To mitigate the high bandwidth requirement of transmitting a high-resolution map from the robots to the operators, [Tabib et al., 2021] proposed a method for real-time occupancy reconstruction from Gaussian Mixture Models (GMM) with depth sensor observations. Furthermore, the work in [Qin et al., 2019] utilized a collaborative team of an Unmanned Aerial Vehicles (UAVs) and an Unmanned Ground Vehicles (UGVs) where the UGV performed fast autonomous exploration to construct 2.5D coarse maps and the UAV conducted 3D fine mapping of the environment. While most previous work focused on solving one or several challenges in the underground environments or related settings, the SubT Challenge pushed the robotics community to develop novel technologies in all four technical areas: Autonomy, Perception, Networking, and Mobility. For the SubT Challenge, fragile autonomy, frequent and demanding supervision or teleoperation, over-specialization to one or the other type of underground environments, sensitivity against sensor degradation, or dependence on always-available communications were not options. The teams were tasked to present a solution that could demonstrate efficient subterranean exploration and mapping with the highest degree of resilient autonomy across environment configurations or operational conditions.

2.2 Other teams in the SubT Challenge

In response to the SubT Challenge, several international teams were assembled, each developing highly advanced and complex solutions. Below we outline key elements of the solutions of the other teams competing in the challenge finals. As at the time of writing not all technical overviews are available, we provide a summary based on already published work of the teams and information provided during the Summit that took place after the Final Event.

Team Coordinated Robotics used custom wheeled, tracked and flying robots, leveraging several open-source mapping frameworks for the ground vehicles [Shan et al., 2020, Shan and Englot, 2018] and the aerial robots [Labbé and Michaud, 2019] depending on the specific sensor setup. To traverse through narrow areas, the robots employed a wall following mode or were driven manually by the Human Supervisor while frontier-based exploration mode was deployed to explore open areas. Furthermore, a mix of WiFi and Gigabit fiber ethernet devices running the advanced B.A.T.M.A.N routing protocol [Johnson et al., 2008], in addition to the droppable wireless nodes, created the communication system for the robotic team. For the robots to detect artifacts, YOLOv4 [Bochkovskiy et al., 2020] network was run with onboard camera images while the gas sensor and bluetooth detection were also utilized to detect artifacts. Team Coordinated Robotics was a

self-funded team.

Team CoSTAR developed a fleet of Spot legged robots, Husky wheeled robots, Balto RC cars, Telemax hybrid wheeled/tracked vehicles and custom flying as well as hybrid ground/aerial robots for the Final Event. Their approach was based on an uncertainty-aware autonomy framework, dubbed NEBULA [Agha et al., 2021], which constructed the belief space over the robot and world states and performed reasoning and planning in that belief space. NEBULA was an extensive architecture including several modules: (i) robust, resilient Simultaneous Localization and Mapping (SLAM) front-end [Palieri et al., 2020] and back-end [Ebadi et al., 2020] for large-scale environments, (ii) semantic understanding and artifact detection modules [Terry et al., 2020], (iii) uncertainty-aware traversability analysis as well as risk- and constraint-aware local planning [Fan et al., 2021, Thakker et al., 2021] modules, (iv) efficient global exploration and coverage motion planning in belief space using a hierarchical planning approach [Kim et al., 2021], (v) high-bandwidth and resilient network solution for a multi-robot system [Ginting et al., 2021], (vi) learning-enabled adaptation module [Fan et al., 2020a, Fan et al., 2020b]. The autonomy stack was implemented for several robot types including wheeled, legged, flying robots, and field-tested in various environments. Team CoSTAR was a DARPA-funded team from the beginning of the challenge.

Team CSIRO Data61's approach was presented in [Hudson et al., 2021] in which a team of BIA5 OzBot tracked robots, Spot legged robots, and Emesent Hovermap drones utilized the Wildcat SLAM system [Ramezani et al., 2022]. The robots explored the environments by choosing the most informative 3D frontier in an efficient manner [Williams et al., 2020]. Moreover, for the UGVs to navigate through challenging unstructured terrain, a common local navigation module optimized for negative obstacles was developed in [Hines et al., 2021] as well as a deep reinforcement learning approach to allow the tracked robots to traverse through narrow gaps in the environments [Tidd et al., 2021]. Additionally, the work in [Tychsen-Smith and Petersson, 2018] described a novel method used to obtain highly competitive localization accuracy with reasonable inference time for artifact detection and localization. A mesh network of radios formed by the modules on the ground robots, UAVs, and droppable nodes was utilized to share data between nearby robots and the Base Station when connected, as described in [Hudson et al., 2021]. Additionally, the artifacts were detected using onboard RGB cameras, and by measuring Bluetooth/WiFi signal strengths and gas concentration. The detection was then relayed back to the Human Supervisor for reviewing before submitting the artifact report to the scoring server (DARPA Command Post). Team CSIRO Data61 was a DARPA-funded team from the beginning of the challenge.

Team CTU-CRAS-NORLAB's multi-robot heterogeneous exploration system included Clearpath Husky wheeled robots, Absolem tracked robots, PhantomX Mark II crawling hexapods, and custom flying robots [Rouček et al., 2020, Rouček et al., 2021] in addition to several Boston Dynamics Spot, HEBI robotics LiLy, and custom SCARAB II legged robots as well as Superdroid Marmotte HD2 tracked robots. Their mapping solution was based on a variant of the Iterative Closest Point (ICP) algorithm as per [Pomerleau et al., 2013, Pomerleau et al., 2014]. The frontier-based exploration strategy described in [Bayer and Faigl, 2019] was deployed on the UGVs to explore the environments while the elevation mapping framework for the ground robots was described in [Bayer and Faigl, 2019, Bayer and Faigl, 2020]. Additionally, the state estimation and control pipelines used for the aerial robots were presented in [Baca et al., 2021] and the path planning as well as frontier-based exploration strategy were detailed in [Krátký et al., 2021]. Furthermore, three different communication systems were utilized: short-range link (WiFi), mid-range link (Mobilicom mesh network), and long-range link (droppable "Mote" modules) supported communication with different levels of reliability, bandwidth, and usage, as detailed in [Rouček et al., 2020]. Moreover, artifact detection was performed onboard every robot using a customized version of YOLOv3 [Redmon and Farhadi, 2018], then a Kalman filtering over multiple detections that was temporally consistent was utilized to estimate the final position of the detected artifact. Team CTU-CRAS-NORLAB was, for the first two competition events of the SubT Challenge, a self-funded team but became a DARPA-funded team due to their performance.

Team Explorer's entry to the Final Event of the SubT Challenge included a combination of custom wheeled UGVs, MAVs [Scherer et al., 2021] in addition to a Spot legged robot with custom payload. Their mapping framework, illustrated in [Zhao et al., 2021], employed an IMU-centric approach to combine the advantages of

tightly-coupled and loosely-coupled methods. Specifically, in the SubT Challenge, the LiDAR-inertial version of this framework was utilized. The planning module used a hierarchical framework, described in [Chao et al., 2021], including a global planner level which maintained data sparsely and computed coarse paths at the global scale, and a local planner which planned kinodynamically feasible paths in the local area for the vehicle. Additionally, [Yang et al., 2021b] developed another planning approach that dynamically updated a global visibility graph. This planner was capable of dealing with navigation tasks in both known and unknown environments as well as dynamic obstacles. For the robots to communicate with each other and with the Base Station, droppable radio nodes were deployed by the ground robots according to a cost function using distance, line-of-sight and RSSI conditions, as presented in [Scherer et al., 2021], or a predictive mapping approach, as per [Tatum, 2020], to maximize communications coverage. Moreover, a multi-modal artifact detection and localization system was developed in [Agrawal, 2019] to help provide situational awareness of the environments. Team Explorer was a DARPA-funded team from the beginning of the challenge.

Team MARBLE deployed Spot legged robots and Husky wheeled robots in the Final Event with the SLAM pipeline based on the tightly-coupled lidar-inertial odometry framework described in [Shan et al., 2020]. For volumetric mapping, the team used a modified version of [Hornung et al., 2013] for efficient map sharing as well as incorporating traversability and stairs semantic information. In the Final Event, a graph-based global planner combined with a local reactive controller [Ohradzansky et al., 2021] was deployed while the multi-agent coordination strategy and artifact detection pipeline were illustrated in [Ohradzansky et al., 2021]. Additionally, a custom-developed ROS transport solution based on UDP and a Meshmerize mesh network that prioritized reconnection times were utilized to create effective communication systems in subterranean environments as detailed in [Ohradzansky et al., 2021]. Team MARBLE was a DARPA-funded team from the beginning of the challenge.

Finally, team Robotika used only custom wheeled robots which were equipped with 2D LiDARs and Realsense RGBD cameras for SLAM, traversability estimation, and obstacle avoidance as well as LoRa and WiFi modules for communication purposes. Team Robotika was a self-funded team. Finally, it is worth mentioning that additional teams participated in earlier phases of the SubT Challenge and the above discussion is limited to those competing in the Final Event.

3 CERBERUS system-of-systems

This section outlines the CERBERUS' robots, their system design, and specifics with respect to the key technical areas that characterized the SubT Challenge: Mobility, Perception, Autonomy, and Networking. Additionally, the last sections show how the robots detected the artifacts placed in the environment and how the Human Supervisor interacted with the deployed agents.

3.1 Mobility

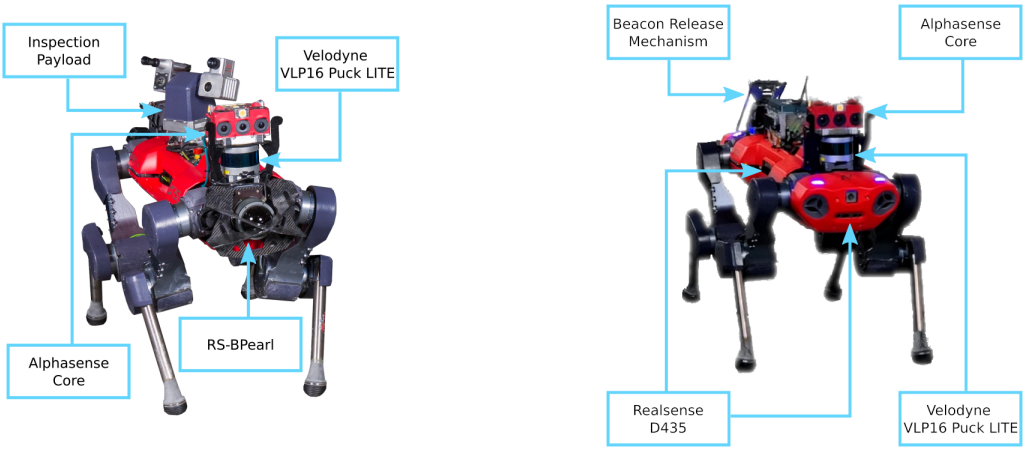
In this section, we summarize the main robotics platforms deployed by Team CERBERUS at the Final Event. We introduce the ANYmal C SubT and the subterranean aerial robots in terms of their hardware development and main software components. Finally, we present a roving robot used to support underground operations.

3.1.1 Subterranean walking robots - ANYmal C SubT

Team CERBERUS developed and deployed a team of legged robots specialized to operate in underground environments. In the Tunnel and Urban events, the team relied on the "ANYmal B300" quadrupeds by ANYbotics [Tranzatto et al., 2022], which were replaced with the "ANYmal C100" series quadruped from the same company for the Final Event. The ANYmal C100 series quadrupedal robot is a robust and agile legged system tailored for autonomous, long-endurance tasks in challenging environments. This robot is

water-proof and dust-proof (Ingress Protection (IP) 67), able to operate in humid and dusty conditions, and designed for robustness for long-term operations. It features dedicated software and hardware interfaces to make it possible to extend it for application-specific purposes. We customized this platform with several hardware and software modifications to create a specialized version: the ANYmal C "SubT".

The ANYmal C SubT platform weighed 55 kg including payload and could operate for 80 min while continuously walking. There were two types of ANYmal C SubT robots: the "Explorer" (Figure 2a) and the "Carrier" (Figure 2b) robots, which differed in their sensor configurations and mission role. The Explorers' goal was to proceed deep in the subterranean environments, while detecting as many artifacts as possible. The Carrier robots had fewer sensors and instead supported underground operations by carrying and deploying the communication-extender modules to increase the range of the wireless network.



(a) ANYmal C SubT Explorer

(b) ANYmal C SubT Carrier

Figure 2: ANYmal C SubT robots used in the Final Event.

Hardware modifications

Sensors - The ANYmal C SubT robots featured a common sensor suite located on one side of the platform for the Final Event. This suite included a "VLP16 Puck LITE" LiDAR by Velodyne and an "Alphasense Core" visual-inertial sensor by Sevensense (Figure 3a). The Alphasense Core encapsulated three monochrome and four color cameras, giving each robot a wide field of view (front, left, right, and upwards directions) to increase the chances of detecting artifacts while exploring unknown areas. Each camera has a 0.4 MP sensor and mounts a lens with opening angles of 165.4° x 126° x 92.4° (DxHxV). The Alphasense was fixed on a rigid, custom-made CNC milled aluminium mount to position the cameras precisely and tilt the side cameras to have clear field of view without including any part of the robot. To protect the unit against water and dust the entire camera array was enclosed in a sealed PA12 Multijet Fusion Case with flexible SLA gaskets around the cameras. Each sensor unit was equipped with four high power, air cooled LEDs. Each of the four LEDs pointed in one of the camera directions (left, right, front and upwards) illuminating the entire field-of-view. Additionally, each ANYmal featured a "SDC30" gas sensor by Sensirion and a Bluetooth sensor. In addition to the common sensors suite, the Explorers were mounted with a pan-tilt unit head (Inspection Payload) for artifact detection. This head featured a 10x optical zoom camera, a thermal camera, a microphone, and a spotlight. Its yaw and pitch angles could be controlled to inspect desired areas around the robot, where the main cameras could not detect potential artifacts due to distance or darkness. Each hardware modification was designed to respect the robot's Ingress Protection Code, thus resulting in the ANYmal SubT version to maintain the IP 67 rating.

Terrain mapping sensors - Reliable terrain perception is critical for safe navigation and locomotion, especially during autonomous missions. Therefore, terrain mapping sensors should be accurate and reliable in challenging environmental conditions and cope with different difficulties, such as darkness or reflective surfaces. The Explorer and Carrier agents presented different sensor configurations reflecting their mission roles.

Carrier robots used four "RealSense D435" active stereo sensors by Intel. This is a relatively lightweight and compact sensor (90 mm x 25 mm x 25 mm, 72 g), allowing each Carrier to additionally ferry on its back four communication-extender modules. On the other hand, the Explorers presented two "RS-BPearl" LiDARs by Robosense (100 mm x 111 mm x 100 mm, 920 g), each mounted on one end of the robot (Figure 3b). The RS-Bpearl is a dome-LiDAR that measures the reflection times of emitted laser pulses using a hemispherical scanning pattern. It offers a 90° field of view but presents a sparse point-cloud compared to other laser sensors (such as the "Ouster OS0" LiDAR). It is robust against sunlight and absorbent material, especially if compared to active stereo sensors. We chose it for the Explorer's configuration because of its sturdy performance in different underground settings, featuring dust, water, and reflective materials, thus making it a good fit for the role of these robots.

Parallel computing - The terrain mapping algorithms can be efficiently parallelized using a Graphical Process Unit (GPU), while visual object-detection also benefits from running neural-networks on a GPU. A "Jetson AGX Xavier" by Nvidia was used on the legged agents. The Explorer robots featured an on-board GPU inside their main body. In contrast, we developed an extra hardware module (Figure 3c) containing a GPU and additional electronics for the Carriers to ensure all the ANYmal C SubT robots had the same computational hardware configuration. We developed an air-cooled, water-proof, dust-proof, milled case to house the Jetson module and the additionally needed electronics.

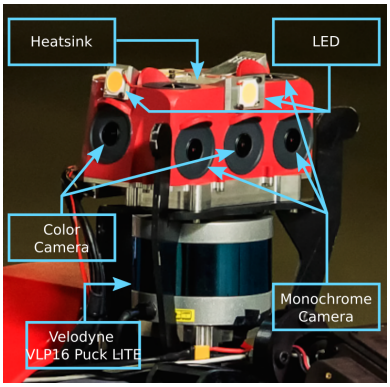
Beacon release mechanism - To deploy the communication-extender modules (also referred to as WiFi breadcrumbs or beacons) described in Section 3.4, the Carrier ANYmals featured a release mechanism (Figure 3d). It could carry four breadcrumbs, each held in place by an electromagnet. To deploy a beacon, the robot tilted its base and deactivated the electromagnet. The beacon slid along the low friction carbon fiber tubes, out of the mechanism, to land in an upright position on the ground.

Locomotion controller

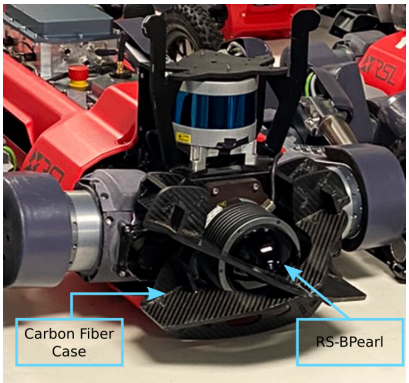
To walk stably on challenging terrain and ensure that the ANYmal robots could explore deep in the course, our locomotion controller had to fulfill the following conditions. First is the ability to walk on a variety of challenging terrains. The course contained tunnel, urban, and cave sections simultaneously, presenting different characteristics. Our controller was able to handle these diverse challenges posed by different settings without further input from the supervisor, i.e., it was not necessary to specify whether the controller had to use a specific gait to climb stairs or to locomote over rough terrain. Next is robustness. Both sensor malfunctioning and measurement degradation can occur during a mission. On the one hand, sensors themselves could stop working for several reasons, such as connection failure or physical damage, making new measurements unavailable. On the other hand, sensors' measurements can degrade due to external factors such as smoke, mist, fog, etc. The controller must be robust even under these conditions. We deployed a learning-based perceptive locomotion controller trained in simulation [Miki et al., 2022a] to satisfy these requirements. The robot learned to walk over randomly generated terrains within the simulation environment using reinforcement learning. It receives geometric information from an elevation map to walk smoothly over large steps or stairs. In addition to this, we used domain randomization during learning, such as the mass of the robot, the friction coefficient of the ground, and external forces so that it can respond to various situations that robots face in the real world. To enable the robot to respond to these cases, we added various distractors during learning to train the robot not to rely on map information when it becomes unreliable or unavailable. As a result, a single controller was able to handle all course areas during the competition. We also trained the body tilting action to deploy the wifi beacon on sloped or rough surfaces.

3.1.2 Subterranean aerial robots

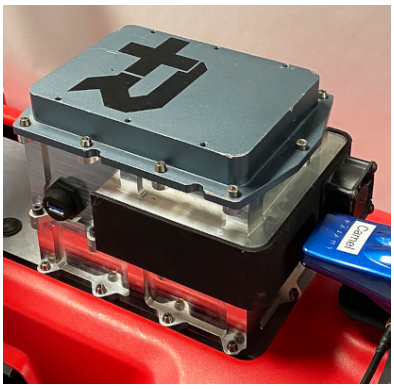
Alongside the team of legged robots, Team CERBERUS developed and deployed a team of "Aerial Scouts" to provide rapid exploration capabilities, especially in regions of the course which were inaccessible by ground robots. In the Tunnel and Urban events, the team relied on traditional multirotor systems and collision-tolerant designs. First, a set of systems based on the DJI Matrice 100 platform were developed and modified to host a 3D LiDAR sensor alongside color and thermal cameras. One of such platforms



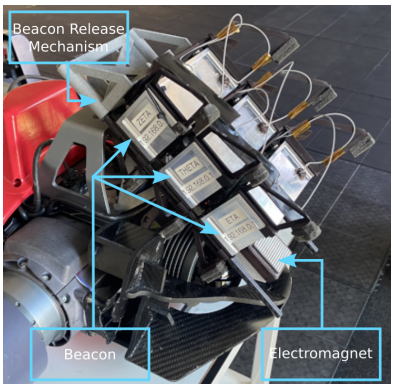
(a) Common sensor suite



(b) RS-BPearl LiDAR and its protection



(c) GPU Payload

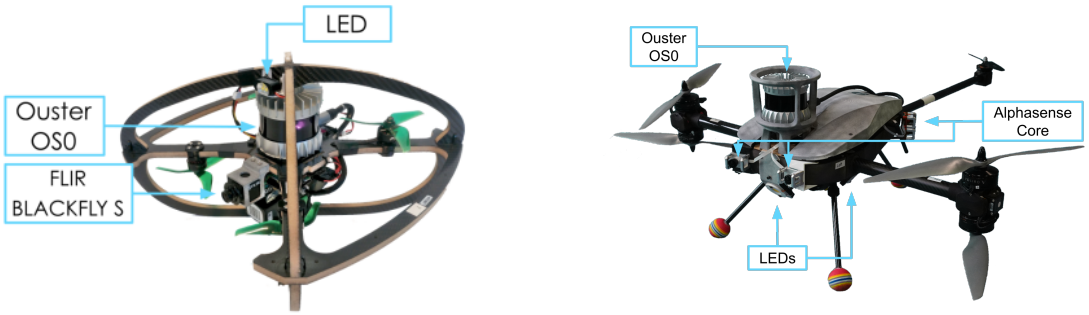


(d) Beacon release mechanism

Figure 3: Hardware customization for the ANYmal C SubT. (a) Common sensor suite included the VLP16 Puck LITE LiDAR sensor and the Alphasense Core unit. (b)RS-BPearl LiDAR for elevation mapping enclosed in its protective and lightweight carbon fiber case. (c) Jetson Xavier and additional peripherals mounted on top of the Carrier robots, inside an actively cooled case. (d) Beacon release mechanism with three loaded WiFi beacons held by electromagnets.

is depicted in Figure 5a. A collision tolerant platform called Gagarin, developed in collaboration with Flyability, was also used for the Tunnel and Urban events [Tranzatto et al., 2022]. Identifying the need for systems capable of navigating more constrained passages, we designed and developed a new collision-tolerant platform, named RMF-Owl [Petris et al., 2022], to traverse extremely narrow settings. In addition to RMF-Owl, we developed the larger and longer range Kolibri flying robot for the Final Event, although it was not deployed due to the narrow settings of the environment. The robots developed for the Final Event are depicted in Figure 4. Subsequently, we detail RMF-Owl and Kolibri, while the other two types of robots are documented in [Tranzatto et al., 2022].

The RMF-Owl was custom-designed to have a compact form-factor and carried a Khadas VIM3 Single Board Computer incorporating 4x 2.2 GHz Cortex-A73 cores, paired with 2x 1.8 GHz Cortex-A53 cores, alongside a Neural Processing Unit (NPU) as the main computation unit. This computer was used to run the entire autonomy stack with modified lightweight versions of our localization and mapping framework (c.f. section 3.2.2), the exploration planner (c.f. section 3.3.1) and device drivers for the various sensors alongside the networking components. The neural network for artifact detection was executed on the NPU to gain a speedup during inference (c.f. section 3.5). Its sensor payload included an "Ouster OS0-64" or "Ouster OS1-64" LiDAR (different robot versions were developed) featuring wide vertical Fields of View (FOVs) of 90° and 45° respectively and a horizontal FOV of 360°. Other sensors included an IMU, and a FLIR "Blackfly S" 0.4 MP color camera with a horizontal and vertical FOV of $[85, 64]°$, the latter used for object detection. The



(a) RMF-Owl (b) Voliro-T Kolibri

Figure 4: The new aerial robots developed for the Final Event.

outer dimensions of the robot were 38 cm x 38 cm x 24 cm and thus alongside its sensing suite was rendered suitable for going through staircases, vertical shafts, and narrow passages while detecting artifacts along the course. Additionally, to traverse and detect objects in dark environments, RMF-Owl was equiped with an off-the-shelf LED lighting system, increasing the visual range and enhancing the environment texture. The total weight of the RMF-Owl Aerial Scout, shown in Figure 4a, was 1.46 kg with a maximum flight time of 10 min.

The larger Kolibri, shown in Figure 4b, is based on a 5DoF multi-directional tricopter platform from Voliro, which we customized with full autonomy for long-range and vertical exploration missions. Its front propellers are fully-actuated, enabling it to pitch independently of translation and affording good flight stability near surfaces. With a takeoff weight of 6 kg, it carried a sensor payload similar to that of the ANYmals, using an "Alphasense Core" unit with three color and three monochrome cameras (each with 0.4 MP) for 270° artifact detection and multi-modal mapping. We paired these with the same high-power LEDs. However, to conserve power and weight for the aerial robot, the LEDs were flash synchronized with its cameras through the Alphasense Core, which also obviated the need for active cooling. It also carried a wide vertical FOV "Ouster OS0-128" LiDAR. To support the computing needs of our large sensor payload, the robot hosted a 16-thread AMD 4800U with 64 GB of RAM, as well as three Intel Neural Compute Stick 2 modules (one per color camera) for artifact detection. The maximum flight time of Kolibri was 22 min with the full competition sensor payload.

3.1.3 Subterranean roving robot

Finally, we developed a wheeled platform to act as a supporting agent to all the other robots and provide extended network connectivity into the course. Shown in Figure 5b, this robot was based on an Inspector Bots Super Mega Bot (SMB) robot platform, with a modified chassis to mount equipment for extending network connectivity, a variety of sensors, an onboard computer, light sources, and a battery to power the components. The 135 kg robot hosted a Zotac VR Go Backpack PC having a 6th generation Intel i7 CPU and an NVIDIA RTX 1070 GPU inside a cage-like structure. We manufactured a two-level sensor mount to house four "Blackfly S" color cameras by FLIR, two "Boson LWIR" thermal cameras by FLIR, alongside a "VN-100" IMU by VectorNav, and a "VLP16 Puck LITE" LiDAR by Velodyne. The platform also featured a 300 m long, ruggedized optical fiber cable, mounted on a controllable reel at the back of the robot. This cable was connected to the Base Station and the reel could automatically wind and unwind to prevent tangling into the wheels of the robot as it moved within the course. The wireless communication modules onboard this robot were connected to a 5.8 GHz panel antenna, placed at the front of the robot. Accordingly, through the combination of the long optical fiber cable and the high-gain antenna, this system acted as a network range extender and provided high-bandwidth connectivity to the Base Station. In addition to the role of a network extender, the robot featured the necessary sensors and perception software, allowing it to operate independently as a backup platform. This robot was teleoperated by the Human Supervisor and did not

(a) Aerial scout robot platform

(b) Roving robot platform

Figure 5: The DJI Matrice M100-based aerial scouts, and the roving robot were used throughout the competition for all three events (Tunnel Circuit, Urban Circuit, and the Final Event).

operate autonomously.

3.2 Perception

This section covers Team CERBERUS' perception pipeline, including onboard localization, multi-robot mapping, and terrain perception and classification. Firstly, we discuss the utilized sensor modalities with respect to how they were configured, synchronized, and calibrated. Next, we detail the onboard complementary multi-modal localization and mapping as well as the elevation mapping pipelines. Finally, we outline the multi-robot mapping approach combining the individual onboard estimations. It is noted that the competition necessitated that teams localize artifacts within a certain level of accuracy against groundtruth locations and report accurate maps. Alignment with the DARPA-defined coordinate frame (on which ground-truth is expressed) was achieved through the use of the "DARPA Starting Gate" which was at the entrance of the mission course. The DARPA Starting Gate offered multiple options for alignment including a) AprilTags, b) retro-reflective targetsm c) survey prisms and d) survey markers. Team CERBERUS performed alignment with the DARPA Starting Gate using only the AprilTags.

3.2.1 Sensor setup and calibration

Since heterogeneous sensor setups can impose significant challenges on a multi-modal and multi-robot mapping approach, all legged robots as well as the Voliro-T Kolibri were equipped with the same sensory system (cf. section 3.1.1 and section 3.1.2). This included the same cameras, lenses, synchronization logic and illumination hardware. In the context of robotic perception, this enables simpler parameter tuning as well as improved inter robot loop closure detection. Tailored to miniaturized system design and prolonged flight endurance for their overall size, the DJI Matrice 100 based aerial scouts and the collision-tolerant flying robots of our team utilized a LiDAR system, alongside a "Blackfly" color camera by FLIR and a "VN-100" IMU by VectorNav. One of the Matrice 100-based systems further integrates a "Tau2" thermal camera by FLIR.

Another important factor is the time synchronization between the sensors and the robots. The ANYmal robots where equipped with a VLP-16 Puck LITE LiDAR that timestamped pointclouds based on arrival time, whereas the Voliro-T featured an Ouster OS0 which was synchronized to the computer clock via Precision Time Protocol (PTP). Similarly, the Alphasense Core was also synchronized using PTP on each robot. For the DJI Matrice 100 based aerial scouts and the collision-tolerant flying robots, a software-synchronized approach was employed in which the offset was estimated among the time-of-arrival of sensor

packages.

Sensor calibration
Reliable sensor calibration is vital to most sensing-based algorithms. Our calibration efforts targeted mainly intrinsic camera calibration as well as extrinsic camera-to-camera and camera-to-LiDAR calibrations. Calibration-critical multi-sensory SLAM applications are, for example, M3RM (Section 3.2.4) and CompSLAM (Section 3.2.2). Furthermore good calibrations are important for the ray-casting of detected artifact locations into the volumetric map of the exploration planner (Section 3.5) as well as camera-based AprilTag detection for anchoring the robot's state estimation in a global reference frame.

Intrinsic calibration - To perform intrinsic camera calibration we used the visual-inertial calibration toolbox Kalibr [Furgale et al., 2013]. Given the wide-angle field-of-view of the cameras that are part of both the Alphasense sensor mount and the visual cameras onboard the multirotor aerial scouts and collision-tolerant flying robots of our team, we found a large enough rigid calibration target to be key to get good calibration results in relevant distances from the camera. We obtained good results with a grid pattern of Apriltags [Olson, 2011] with 6 rows, 6 columns and tag side lengths of around 10 cm. Furthermore, the Kalibr library provides a useful tool to examine camera focus. The method is based on calculating entropy in the image frequency domain [Kristan and Pernuÿs, 2004]. Intrinsic calibration of LiDARs and IMUs were provided by manufacturers directly and were considered sufficiently accurate to be used out of the box. Intrinsic calibration of thermal cameras is achieved with the use of a specialized heated calibration target.

Extrinsic calibration - For camera-to-camera extrinsic calibrations we also utilized the Kalibr toolbox. Alongside the tools for intrinsic camera calibration it allows for jointly estimating temporal offsets and sensor displacements using continuous-time batch estimation. It features options for single camera and IMU extrinsic calibration as well as multi-camera and IMU calibration. The latter is useful if camera FOVs are majorly overlapping and yields increased accuracy by performing co-optimization over all provided data. We considered using optimization based methods, such as presented in [Huang and Grizzle, 2020], to perform camera-to-LiDAR extrinsic calibration. In their work they show results using a 32-Beam LiDAR. However we observed that with a 16-Beam LiDAR the optimization problem tends to be insufficiently constrained. This is likely due to the sparsity of the LiDAR measurements. Accordingly, both for the robots with 16-Beam LiDARs and for our overall set of systems, we chose to use a self developed manual calibration method that allows to visualize projected LiDAR points in the camera image and to thereon dynamically modify extrinsic LiDAR-to-camera transforms.

3.2.2 Complementary multi-modal localization and mapping

Robust localization and mapping are among the core components of the robot autonomy pipeline to ensure reliable operation in complex subterranean environments. They are responsible for providing consistent and low-latency localization to facilitate low-level, real-time operations, such as robot control. In addition, they are also responsible for producing an accurate and informative map to assist high-level autonomy tasks, such as path-planning in complex large-scale environments.

In response, the robots of Team CERBERUS deployed an onboard complementary multi-modal localization and mapping approach (CompSLAM) [Khattak et al., 2020], that ensured reliable and independent robot operations without any dependence on communication with the Base Station. An overview of the proposed approach is shown in Figure 6. CompSLAM utilizes a hierarchical sensor fusion strategy which refines individual or multi-modal pose estimates as they propagate through the framework, ensuring that a robot always has an updated pose estimate and a map available. Furthermore, CompSLAM's modular design made it flexible to adapt to the sensing and compute variances across the robots of Team CERBERUS. Hence, the method allowed to take advantage of the complementary nature of multi-modal sensors for operation resilience, while maintaining the ability to scale cognizant to the available robot compute capabilities to ensure real-time performance across diverse environments and robot configurations.

To provide redundancy against cases of sensor data degradation and estimation failures, CompSLAM performs fusion of visual and thermal imagery, LiDAR depth data, inertial cues, and kinematic pose estimates (when available) in a hierarchical manner. For the fusion of visual, thermal and inertial data, CompSLAM utilizes an Extended Kalman Filter (EKF) based estimator that extends the work of [Bloesch et al., 2017a] to simultaneously operate on both visual and full-radiometric thermal images, as described in [Khattak et al., 2019, Khattak et al., 2020].

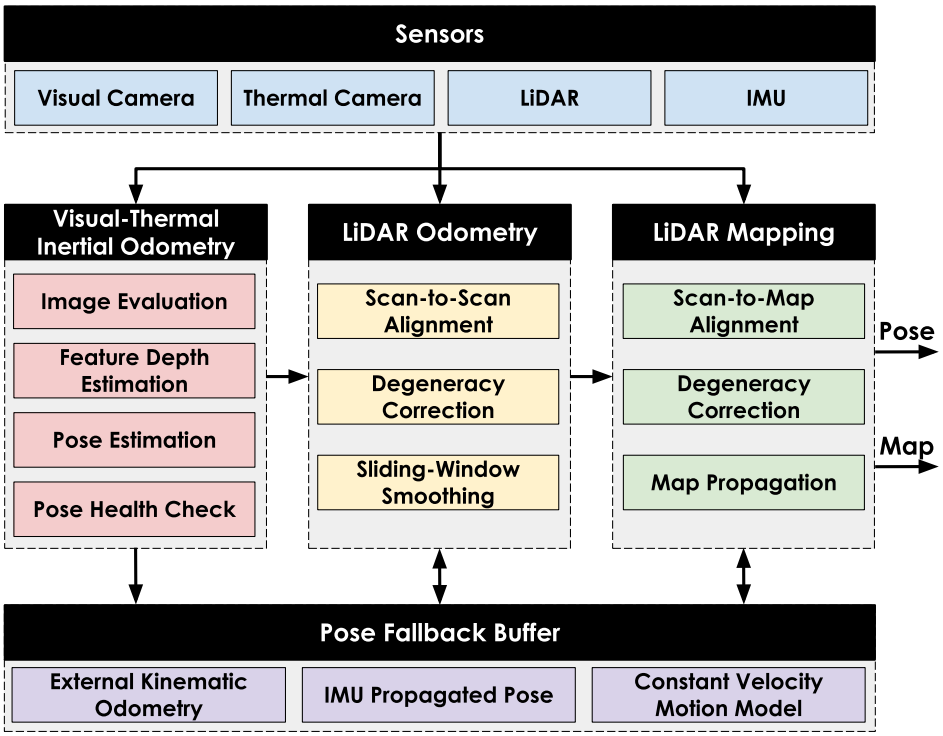


Figure 6: Overview of the onboard complementary multi-modal localization and mapping approach (CompSLAM).

The Visual-Thermal-Inertial Odometry (VTIO) estimator tracks and balances a fixed number of features across multi-spectral images by evaluating the quality of each image subject to a spatial and temporal entropy criterion [Khattak et al., 2019]. This results in a small number of features required to be tracked as part of the EKF state-vector, keeping the computational costs tractable for real-time operation. Furthermore, to maintain low computational costs while facilitating robust scale estimation in degraded environments, sparse depth from instantaneous LiDAR scans is exploited for feature depth initializations and updates. For features enclosed by at least two or more LiDAR scans, a local-flatness check is performed to ensure depth consistency. If passed, features are initialized with the median neighborhood depth acquired from LiDAR scan and associated with a smaller variance on the feature depth to improve the overall scale convergence of the estimator. For features initialized without LiDAR depth, if a LiDAR scan sweeps over a feature due to robot motion during the mission, the feature depth and variance are updated in a probabilistic manner [Graeter et al., 2018]. Although, sparse in nature, the instantaneous LiDAR scans are used instead of the onboard map for depth association to build redundancy into the system in case of a mapping failure.

For building the onboard map, the LiDAR scans along with the VTIO odometry are utilized by the Laser Odometry (LO) and the Laser Mapping (LM) modules. Before utilizing VTIO output as a prior, the LO module evaluates the VTIO health by ensuring that it is within the motion-bounds set by the robot controller. Furthermore, the covariance growth rate of the VTIO prior is evaluated to be within an acceptable margin by employing the D-Optimality criterion [Carrillo et al., 2012]. In case the VTIO health checks fail, the LO module reverts to a set of fallback solutions, in order, for the provision of prior - kinematic odometry (in case of ANYmal [Bloesch et al., 2017b]), IMU propagated pose or a constant velocity motion-model.

Using the obtained prior and consecutive LiDAR scans, the LO module performs a scan-to-scan alignment by minimizing point-to-line and point-to-plane distances, similar to the work of [Zhang and Singh, 2014]. The quality of scan alignment is checked by evaluating the minimum eigenvalues of the optimization Hessian matrix [Zhang et al., 2016]. If optimization degeneracy is detected, the ill-conditioned dimensions of the LO estimation output are replaced with the prior's estimates. The LO estimates are then fused with inertial measurements and smoothed using a sliding-window estimator scheme to improve pose consistency and gravity alignment. The sliding-window estimator is implemented as a factor-graph using the work of [Dellaert et al., 2017]. Furthermore, when the robot is stationary, zero-velocity factors are added to improve the robot pose and IMU bias estimation.

The refined LO pose estimate is then provided to the LM module which performs a scan-to-submap alignment to produce an onboard map. For computational efficiency and, in particular, to improve the speed of establishing point correspondences, the onboard map is divided into blocks of volume 10m^3. Each block is identified by a unique hash calculated from the location of each block's center with respect to the map origin. Furthermore, each map block is sub-divided into line and plane feature sub-blocks. Upon arrival of a LiDAR scan and associated LO pose estimate, all sub-blocks within the FOV of the current scan are identified and independent KD-Trees are built for each block in parallel to allow parallel search for point correspondences. Once a scan-to-submap alignment is performed, a health check similar to that of the LO module is performed to determine the quality of the LM output. In case LM optimization is ill-conditioned, a previous healthy estimate in the sensor-fusion hierarchy is searched and utilized for merging the current scan to update the onboard robot map.

3.2.3 Elevation mapping

An elevation map is a 2.5D representation where each grid cell holds the height of the terrain. We used this map representation for local navigation and perceptive locomotion of the quadrupedal platform. It is built in a robot-centric fashion from depth sensor readings and robot odometry. We developed a GPU-based software for efficient processing [Miki et al., 2022b]. In addition to point cloud registration it performs a visibility cleanup check to quickly remove map artifacts. The method performs raycasting from the sensor origin to each measurement point and clears cells if the ray passes below its stored elevation. To avoid issues with structures thinner than the map resolution, we do not clear a cell if it is currently observed by a measurement. The terrain map used in the SubT Challenge finals had dimensions equal to 8 m x 8 m with a resolution of 4 cm. These settings reflected the sensor limitations and represent a trade-off between accurately capturing the terrain's surface and keeping the computation cost to a reasonable level for onboard processing, thus making it possible to create a real-time terrain representation onboard the ANYmal's GPU utilizing the onboard depth sensor measurements. The elevation map was computed at 20 Hz for Explorer robots, given each RS-Bpearl publishing at 10 Hz, and at 45 Hz for Carrier robots, given each depth camera publishing at 15 Hz. The ANYmal's locomotion controller samples the terrain height in a circular pattern around each foot, as shown in Figure 7. This height information is used by the controller to perform terrain aware locomotion. We integrated a learning-based foothold score estimator [Wellhausen and Hutter, 2021], which gives a foothold score value based on the local geometry and was used by our navigation planner. The foothold score is indicated by black (low) and white (high) color in Figure 7. Furthermore, we enhanced the elevation map through improvements such as upper bound calculation similar to [Hines et al., 2021], which helps planning in the presence of occlusions.

3.2.4 Multi-robot mapping

Beyond individual onboard localization and mapping, Team CERBERUS further developed a solution for multi-robot mapping within the complex and large-scale underground environments. The long-term stability of onboard maps is a pertaining issue in SLAM. Especially given the often limited understanding of the environment, choosing how to fuse new information and decide what is adequate to keep is not trivial. A centralized server, accumulating all available onboard maps, can remedy this by providing a globally consistent environment estimate.

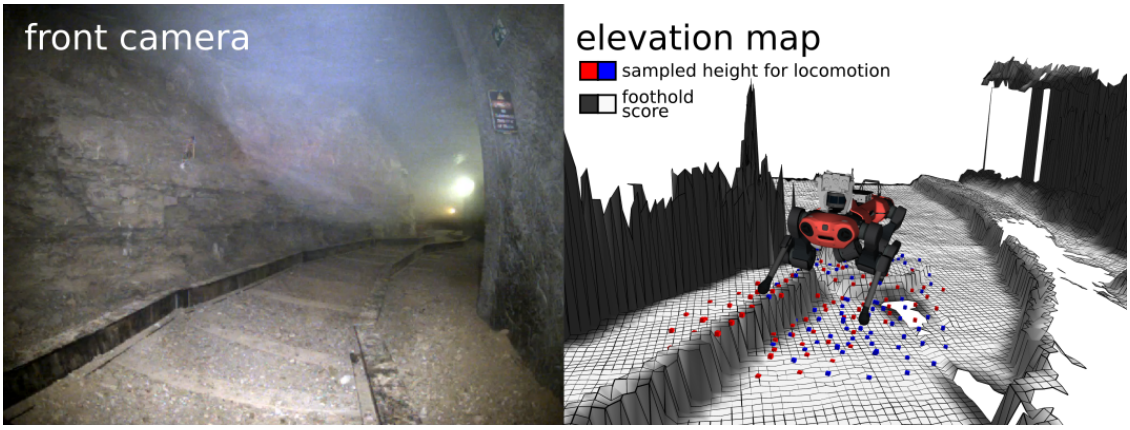


Figure 7: ANYmal's elevation map. The terrain map was built onboard the GPU and used by the learning based locomotion controller and the ANYmal navigation planner. The height information is used by the learning based controller to perform terrain-aware locomotion: blue and red dots indicate samples for left and right feet, respectively. The local planner utilizes the traversability layer (shown as foothold score) for path planning: white areas indicate patches considered as steppable, whereas black ones are not.

Centralized mapping approach

We designed and developed a multi-modal and multi-robot mapping (M3RM) framework using a centralized topology [Tranzatto et al., 2022]. Overall, M3RM comprises of two core components, the M3RM node that creates a factor graph using the onboard complementary localization and mapping solution, and the M3RM server that maintains and optimizes the multi-robot global map. An overview of our developed mapping approach is given in Figure 8.

Each M3RM node on each robot tracks visual features, i.e., BRISK [Leutenegger et al., 2011], using all camera images and triangulates the tracked features to landmarks using the onboard localization. The resulting visual map is then represented as a factor graph and transmitted as chunks (submaps) to the M3RM server at the Base Station. Before transmission, the point clouds are subsampled, compressed, and attached to the factor graph using their timestamp and extrinsic calibration. Accumulating all point clouds per submap, the required size is approximately 2 MB per transmission. The M3RM server used these point clouds to register spatially close poses with each other.

Finally, robots with an established connection to the Base Station transmitted submaps consisting of the onboard odometry estimates and subsampled sensor data to a centralized mapping server. A synchronization logic handles partial transmissions and connection losses and ensures that only completed submap transmissions will be integrated into the multi-robot map. Moreover, the submap transmissions can be batched and compressed to increase the efficiency and to take the network's bandwidth limits into account.

The M3RM server then builds a multi-robot factor graph using all the available sensor modalities and performs intra-/inter-robot loop closure detection using vision and LiDAR. Specifically, visual loop closure searches globally for matching descriptor candidates, while LiDAR loop closures are searched locally between spatially close poses and submaps. For the latter, additional LiDAR constraints are added to the factor graph by accumulating consecutive scans within a submap and aligning to spatially close submaps using ICP. Each finished submap transmission is processed on its own before being integrated into the global multi-robot map. Among other operations, this includes a visual landmark quality check, visual loop closure detection, LiDAR loop closure detection, and submap optimization. The processing of each submap is independent with respect to other submaps and thus, the M3RM server will run multiple operations concurrently. After processing the individual submaps, they are merged into the global multi-robot map which the M3RM server continuously processes and optimizes in a similar vein. The M3RM server runs indefinitely a fixed set of operations, i.e., visual loop closure, LiDAR loop closures, and optimization. However, in this case the

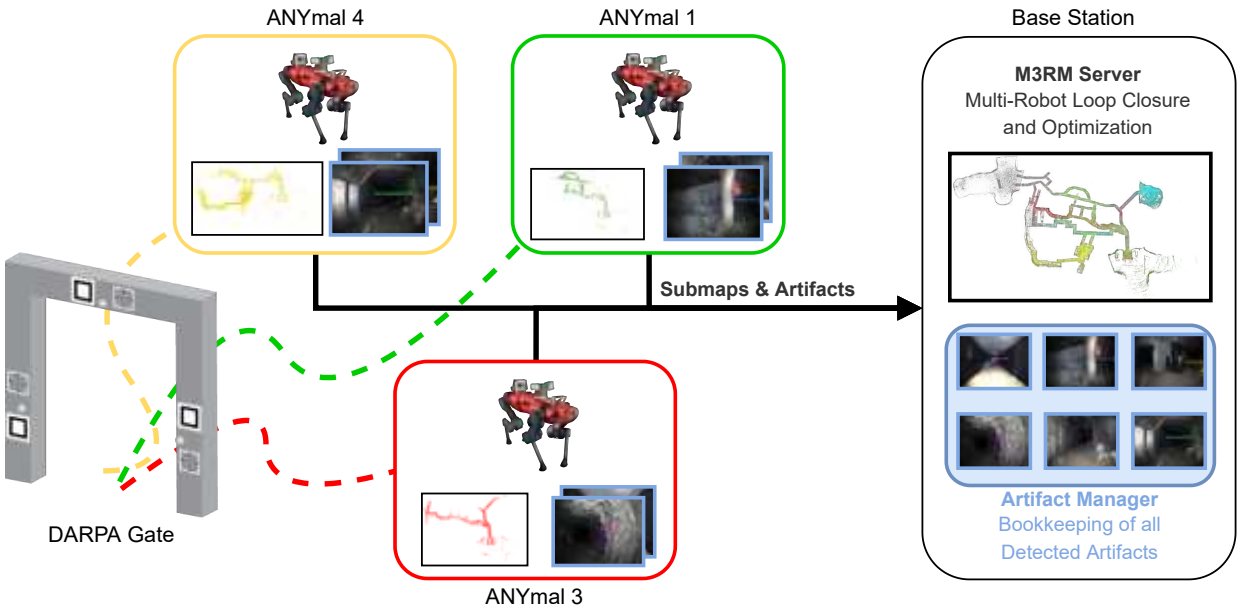


Figure 8: Overview of the employed centralized multi-robot mapping approach. Each robot aligns its estimations (poses and map) to the DARPA Gate and maintains its own onboard map independent of the other robots while exploring the environment. When in communication range to the Base Station, the onboard maps and the detected artifacts are transferred to the Base Station where the server merges the robot maps into the global multi-robot map. All artifact detections are accumulated at the Base Station and visualized to the Human Supervisor for verification.

operations consider the scope of the global multi-robot map.

Finally, the global multi-robot optimization is limited to a maximum of two minutes to guarantee fast iterations. This allows the M3RM server to integrate the most recent information that might contain crucial loop closures quickly and provide the most recent data to the Human Supervisor for ad-hoc mission planning and execution.

The M3RM server also enables to provide a localization feedback to the individual robots about their onboard position [Bernreiter et al., 2022] to circumvent inconsistencies between them. The robots and server can then exchange information in a bi-directional way to not only improve the onboard state estimation, but also serve as a catalyst for collaborative multi-robot planning.

Long-term operation
The plethora of sensor data being transmitted from the robots to the mapping server is filtered and subsampled to provide a high-quality and rapid global estimate of the environment. Nevertheless, the amount of data from multiple robots over a longer time span can still significantly slow the operation time. Therefore, the Human Supervisor can choose between multiple performance profiles at the mapping server to adapt the global multi-robot optimization to the current situation. In situations where all robots are not within connection range, the mapping server can increase its computational load, whereas in scenarios where robots have re-established connections should lower its capacities to provide rapid updates.

3.3 Autonomy

The autonomy solution of Team CERBERUS was tailored to the challenges of the subterranean environments. Our aim was to enable resilient autonomy with minimal human supervision in very large-scale settings with

difficult terrain, narrow passages, and multi-branching, multi-level topologies. To realize this, all robots at their core utilized a Graph-based Exploration Path Planner (GBPlanner2) [Kulkarni et al., 2022] for efficient exploration of the environment. The exploration planner was designed to work as-is across all of the team's robotic platforms, with the only change being selected robot-specific parameters and features. Furthermore, robot specific motion planning and path tracking algorithms were used to follow the path provided by the exploration planner or track the waypoints given by the Human Supervisor.

The autonomy architecture of the legged and aerial robots is shown in Figure 9. A coordination module, based on Behavior Tree (BT) (see section 3.3.3), manages the switching between different tasks of the ANYmal's autonomy stack. All the requests from the Human Supervisor are handled by this module. A robot specific Planner Control Interface (PCI) provides APIs for accessing the functionalities of GBPlanner2. The autonomy stack of the aerial robots is relatively simpler with the Human Supervisor requests being handled directly by the PCI.

3.3.1 Exploration planner

At the core of the CERBERUS autonomy lies the Graph-based Exploration Path Planner (GBPlanner2) [Kulkarni et al., 2022] that uniformly guides both legged and aerial robots. GBPlanner2 extends the work of GBPlanner1 [Dang et al., 2020] (used in the Tunnel and Urban Circuit events) to better handle slopes and uneven terrain for legged robots, as well as by providing features for reduced computation times for computationally constrained Micro Aerial Vehicles (MAVs). The planner uses an incrementally built volumetric map using the onboard depth sensors. In the Final event, all robots of Team CERBERUS used "Voxblox" [Oleynikova et al., 2017] as the volumetric mapping framework.

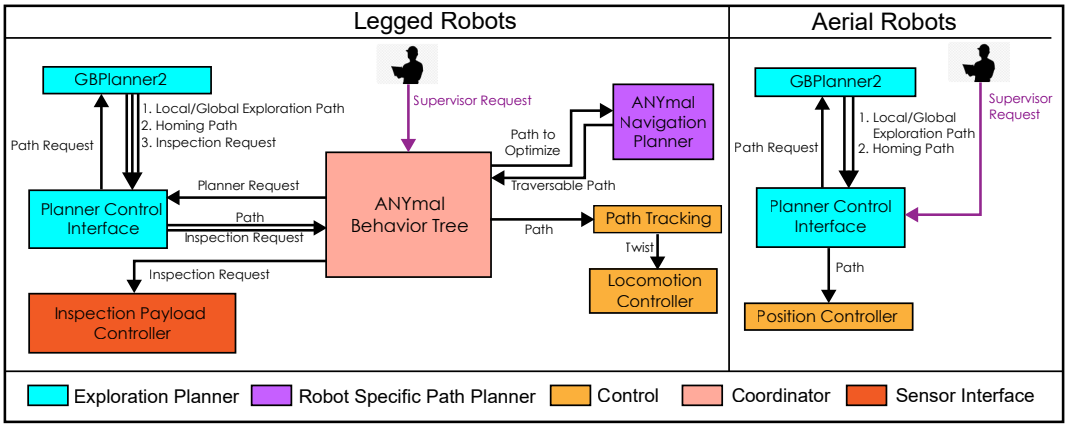


Figure 9: Overview of the Autonomy Stack of Team CERBERUS. The diagram shows a high level overview of the interaction between the sub-modules within the Autonomy Stacks for both legged and aerial robots.

GBPlanner2 operates in a bifurcated local/global planning architecture. The local planner guides the robot to efficiently explore within a local volume around the robot. It first samples a collision-free graph (called "local graph") inside a local subspace of the map (hereafter referred to as the "local bounding box") that is calculated based on the local environment geometry. For legged robots, each vertex and edge in the graph should not only be collision-free but also have traversable supporting ground under it. This is evaluated by projecting the sub-sampled edge (including its end vertices) downward on the map and checking if a) occupied voxels exist in the map below the projected points and b) the inclinations of the segments of the projected edge are within the permissible limit for the robot (Figure 10). To tackle negative obstacles, the planner retains certain vertices (called "hanging vertices") in the graph that are in free space but don't have supporting ground below them if at least one of the edges connecting to them has its inclination within the allowed limit. However, the edges connecting to these vertices are not commanded to the robot. Next, using Dijkstra's algorithm, the planner finds the shortest paths from the current robot location in this graph. The path that maximizes an information gain relating to the expected unknown volume that will be mapped by

traversing that path is selected and commanded to the robot. [Dang et al., 2020] provides further details on the formulation of the information gain, alongside further steps to refine the path quality. Additionally, to reduce the computation time, GBPlanner2 also provides the functionality to calculate this gain only at the end vertices of the shortest paths as well as to cluster these vertices and calculate the gain only for one vertex in each cluster.

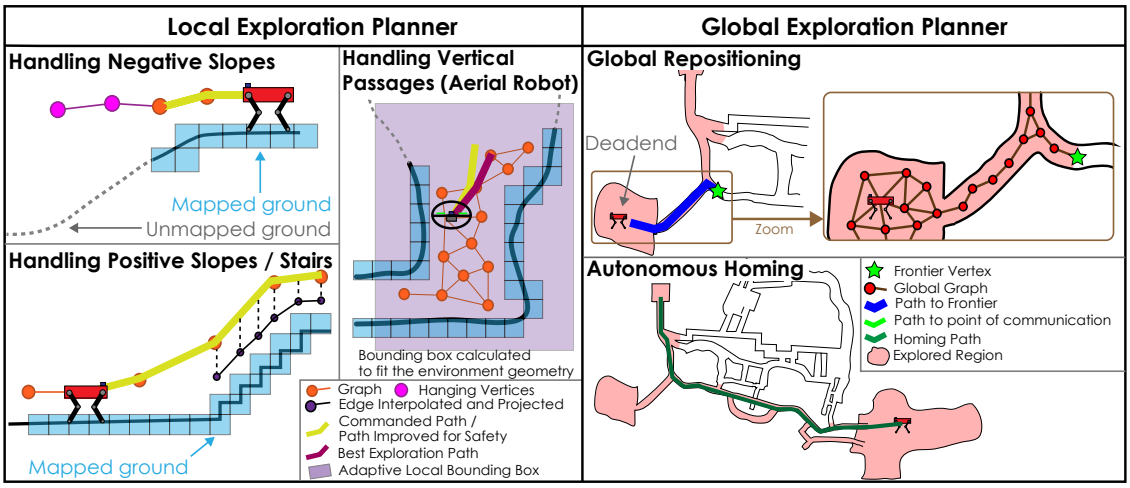


Figure 10: Graphical overview of the exploration planner's bifurcated local/global architecture. The main features involved in calculating the local and global exploration paths are shown.

Due to the complex topology and challenging terrain of the subterranean environments, the local planner might not be able to find a path leading to further exploration. In such a case, the global exploration planner is responsible to re-position the robot to unexplored areas of the map. This is done by exploiting an incrementally built sparse graph (called "global graph") that spans across the known map and stores high information gain vertices called the "frontier vertices" along with their corresponding paths. Furthermore, at each local planning iteration the global planner calculates a path from the current robot location to the home location and commands the same to the robot if the remaining endurance is not enough to perform one more local exploration planning step and return home. Additionally, the global planner also supports calculating paths to goal points provided by the Human Supervisor or the ANYmal's behavior tree. Figure 10 provides a graphical overview of the exploration planner. Similarly, visualizations of various components of GBPlanner2 on data are shown in Figure 11.

3.3.2 ANYmal navigation planner

To avoid local obstacles and follow the exploration path or the exploration waypoints provided by the Human Supervisor safely, we developed a navigation planner specifically for legged robots [Wellhausen and Hutter, 2021, Wellhausen and Hutter, 2022]. This navigation planner could locally adjust the exploration path to better cope with the properties of the surrounding terrain, given that legged robots have much higher mobility capabilities than wheeled or tracked platforms due to their ability to step over obstacles. Therefore, classic navigation approaches, which compute a single traversability value per terrain patch independent of robot orientation or motion direction, are limiting. Other teams which deployed both wheeled and legged systems deployed such a solution [Fan et al., 2021].

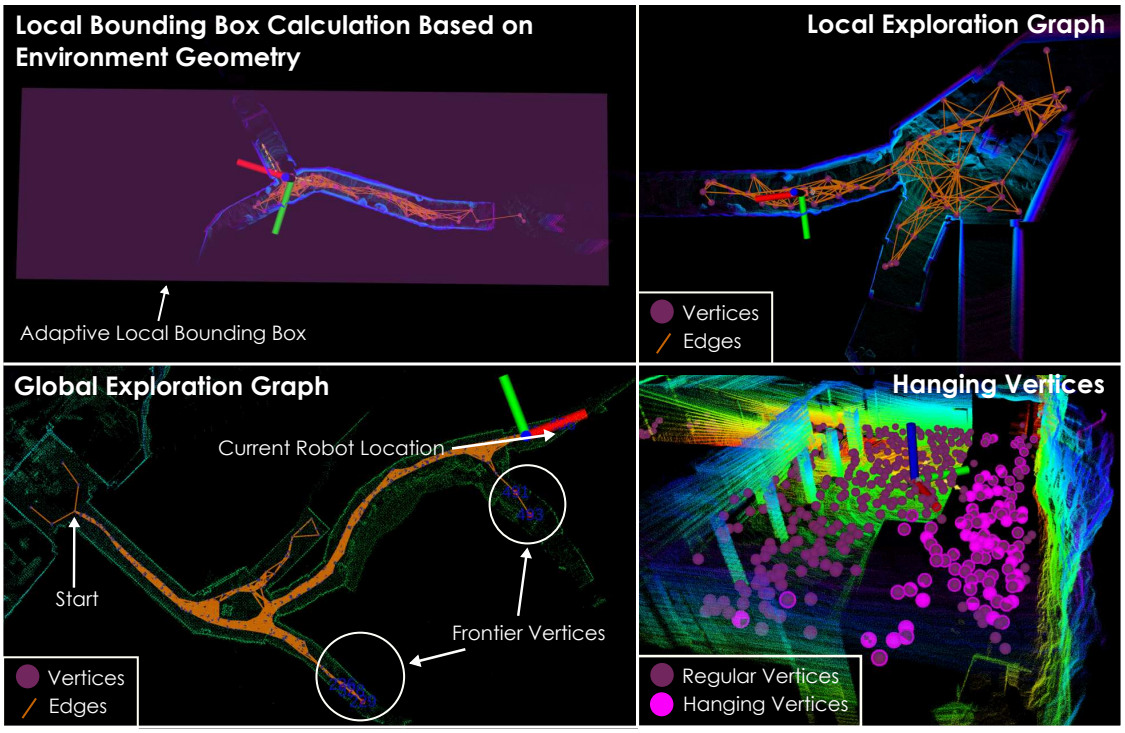


Figure 11: Visualizations of different components of the exploration planner on data. These include the adaptive local bounding box, the graph spanned by the local planner, the graph spanned by the global planner along with the frontier vertices, and the hanging vertices (feature for legged robot path planning).

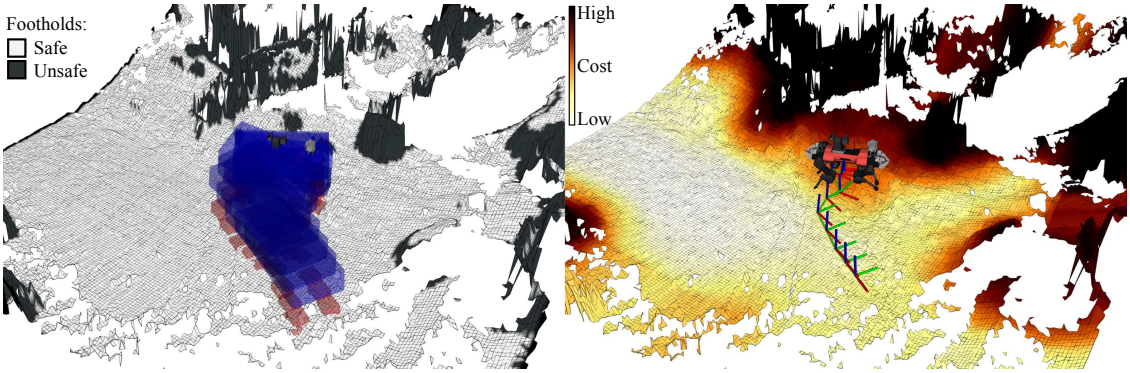


Figure 12: ANYmal navigation planner in the cave section during the Prize Round of the Final Event. (Left) Blue torso collision boxes need to be collision free while red reachability boxes need to be in contact with the environment. Black parts of the height map are considered invalid for footholds. (Right) A learned motion cost (white - low cost, black - high cost) is used to limit locomotion risk and optimize the path for both locomotion risk and traversal time.

Our planner is based on LazyPRM* [Hauser, 2015] with a custom sampling scheme, which extends poses sampled in 2D to 3D using map information, and favors sampling in regions with low planning graph node density [Wellhausen and Hutter, 2021]. It uses a reachability abstraction of the robot by decomposing it into a torso collision volume and one reachability volume per leg. Since a legged robot needs to touch the ground to locomote, we assume that a valid robot pose needs to have all reachability volumes in contact with the environment, while the torso remains collision-free. To avoid practically infeasible poses, like

walking on a wall, we train a very small convolutional neural network (CNN) on manually labelled data to predict a foothold score from a height map. We can then exclude geometry with low foothold score from collision checking for reachability volumes to disallow stepping on invalid terrain. Because the network only has 120 parameters, it can be trained on only 20 labelled height maps. This generally produces feasible paths, but since a shortest path cost function would have no notion of risk, caused by uncertainty in the environment, imperfect control and path following, and various other sources, we additionally use a learned motion cost [Wellhausen and Hutter, 2022]. To this end, we train a CNN to predict the motion risk, energy and time cost given a height map and a relative 2D goal pose [Guzzi et al., 2020, Yang et al., 2021a]. Data is generated in simulation by deploying a locomotion policy on randomly generated terrain with random directional commands. Energy consumption of the legged robots was of low concern during the Final Event, due to operation time greater than run's duration. However, a single navigation failure could be mission ending; thus we use a weighted sum of risk and time cost, with a five times higher weight on risk.

Although the ANYmal navigation planner plans towards a single goal pose, embedded into our autonomy stack, it receives a reference path from the exploration planner. We therefore iteratively try to plan towards the farthest pose in the exploration path, until we either succeed, or no valid goal pose can be found. Because we use a roadmap-based planner, these path queries can be performed rapidly once a planning graph has been built.

3.3.3 ANYmal behavior tree

The capabilities for coordinating different software modules to execute autonomous missions are provided by a behavior tree module running onboard the ANYmals. Moreover, the behavior tree receives commands from the Human Supervisor and executes recovery behaviors if it detects unexpected states during the mission execution. This module is based on an existing framework [2] and its main components are depicted in Figure 13. Specifically, the *Watchdog* block (Figure 14) triggers recovering actions if unexpected conditions are detected. It continuously checks the robot's position and triggers a backtracking of the traversed path, called "Recovery Homing," in case the robot does not move for a predefined amount of time. The *Process Operator Calls* block handles the commands given by the supervisor, assigning them a higher priority compared to the other onboard modules' requests. If the Human Supervisor allocated a time budget for exploration out of communication range, the *Homing Timer* block checks whether the given budget has elapsed and if so, it triggers the homing functionality of the exploration planner. If the Human Supervisor did not allocate a time budget, the *WiFi Safety* block (Figure 15) checks if the agent is in connection with the Base Station. If the connection is lost (based on a threshold on consecutive, failed pings), it commands the robot to backtrack its path until the connection is reestablished. The *Process Navigation Goals* block handles navigation goals created either by the Human Supervisor or the exploration planner and implements additional mechanisms to manage possible pitfalls. It forwards the assigned goals to the navigation planner to compute a terrain-aware path or directly to the path tracking module if the Human Supervisor decides to bypass the navigation planner. Possible pitfalls handled by this block include (a) the navigation planner cannot find a valid path to any pose in the reference path, and (b) the navigation planner considers the robot's starting position invalid (for example, due to drift in the odometry estimate). For the former case, depending on which module provided the request, either the supervisor gets informed about the inability to continue or an automatic replanning command to the exploration planner is issued. In the latter case, the module automatically resets the elevation map to remove any possible artifacts and issues again the last command that failed to execute. In either case, if the exploration planner provides a goal that cannot be achieved for a given amount of time, the robot is commanded to backtrack. While the robot backtracks its steps, the behavior tree triggers the exploration planner until a solution is found or the Human Supervisor stops the procedure. Lastly, the block *Process Inspections* receives inspection requests, applicable only for the Explorer robots, that are sent either by the exploration planner or the Human Supervisor and puts the other requests on hold until the inspection task is completed.

[2] `https://github.com/BehaviorTree/BehaviorTree.CPP`

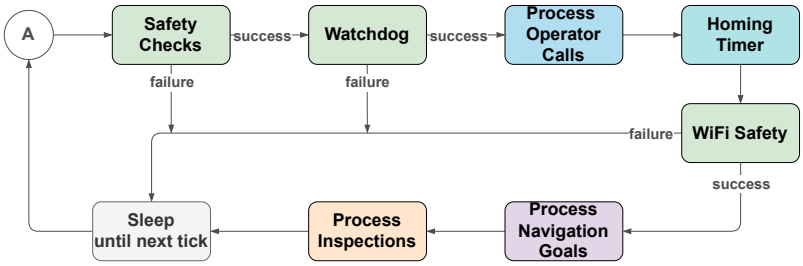


Figure 13: Overview of the ANYmal behavior tree module. Active blocks are displayed in green, blue, and purple. The grey block represents a sleeping block. Based on the given conditions, blocks can either return success and continue to the next block or fail. In case of failure or after reaching the last active block, the behavior tree sleeps until the subsequent execution.

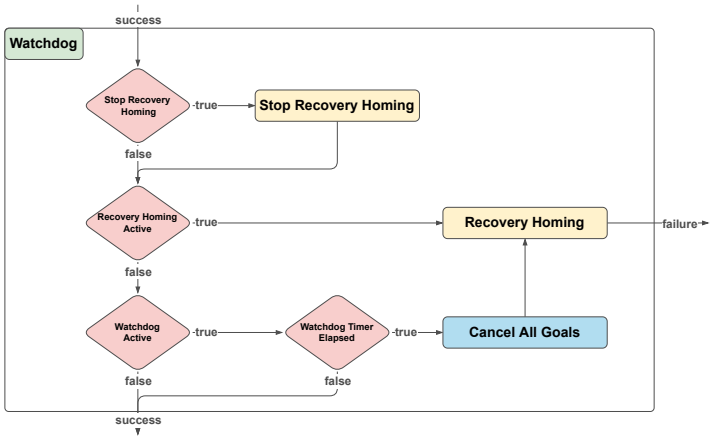


Figure 14: Watchdog module of the ANYmal behavior tree. This module triggers recovering actions if the agent is detected as not moving for a given amount of time.

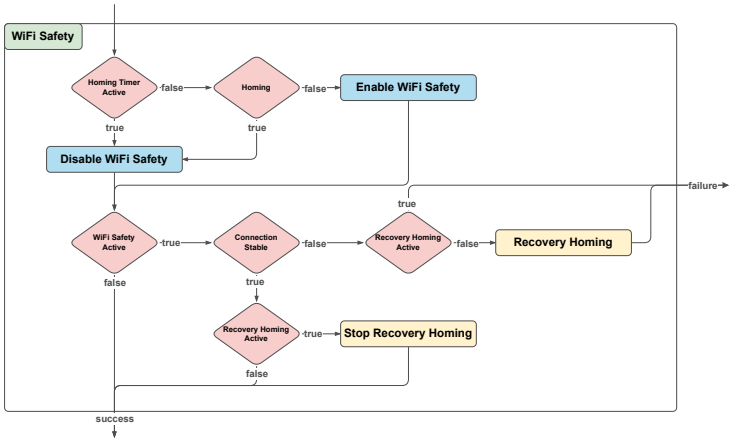


Figure 15: WiFi Safety module of the ANYmal behavior tree. This module checks if the agent is in connection to the Base Station, and if the agent is not assigned a time budget for exploration out of communication range, it commands the robot to backtrack its path until the connection is reestablished.

3.4 Networking

While the autonomy system of each robot is crucial to achieve subterranean exploration with minimal human supervision, simultaneously maintaining a reliable, decentralized wireless network is critical transmit the gathered information (such as the map, artifact locations, and robot poses) from the robots to the Base Station so as to provide situational awareness to the Human Supervisor. Wireless communication is challenging in underground settings due to the unpredictability of the signal propagation and the constrained environment geometry. Similar to the system described in our previous work, [Tranzatto et al., 2022], the CERBERUS system-of-systems employed an ad-hoc wireless mesh network, composed of several physical "nodes," spread of extended distances ensuring connection to the robots exploring deep in the unknown.

The network operated in the 5.8 GHz range and was composed of the Base Station, the legged and roving robots, which acted as nodes of the mesh, alongside the flying robots which acted as clients. Furthermore, Carrier ANYmals could ferry and deploy additional communication-extender nodes, called "breadcrumbs," to expand the reach of the wireless mesh network incrementally upon human command. The mesh network constituted the team's backbone communication infrastructure, which could also be accessed by the aerial scouts, acting as clients. Figure 16 shows an overview of the communication architecture. In the following part, we highlight the main differences of the system deployed in the SubT finals as compared to our intermediate publication.

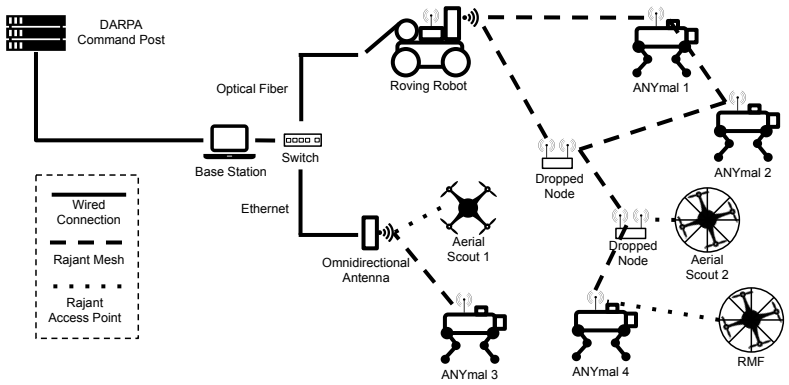


Figure 16: Block-diagram showing the networking infrastructure used during the Final Event. The Base Station, the quadrupedal robots, the roving robot, and the dropped communication-extender breadcrumb nodes created an ad-hoc mesh network, which established the backbone communication infrastructure. The roving robot was also equipped with a fiber optical cable to maintain a connection to the Base Station. Aerial scouts used a standard (802.11) WiFi network, acting as clients connecting to the backbone infrastructure.

Mobile agent nodes
The legged robots and the roving robot featured a commercial off-the-shelf radio ("Rajant BreadCrumb DX2") that acted as a mesh point node and a WiFi access point. Each radio formed a mesh network when used in conjunction with other Rajant's BreadCrumb systems. In this mesh network, the DX2 radios connected directly, dynamically, and non-hierarchically to as many other DX2 radios as possible and cooperated to efficiently route data between clients. Moreover, each radio operated also as a WiFi access point, broadcasting a WiFi network with the same SSID. Aerial robots did not have a DX2 radio or their own - due to their limited payload capability and size constraints. Instead, they connected to the closest existing WiFi access point. In our previous setup for the Tunnel and Urban Circuit events the aerial robots could not connect directly to the legged robots; at the Final Event this limitation was removed.

Base station nodes
Our approach had two solutions for communications at the Base Station. The primary setup was a "Rajat BreadCrumb ES1" radio in the Staging Area, attached to an omnidirectional antenna to connect with the robots before entering the competition course. The ES1 radio can mesh with other DX2 radios. Second,

the Base Station could be connected to a roving wheel robot via a 300 m-long optical fiber cable that could unwind as the rover drove in the course while carrying a DX2 radio. This dual configuration allowed the robots in the Staging Area to connect to the ES1 radio when the roving robot was deployed deep in the course and out of wireless reach. Compared to our setup of the Tunnel and Urban events, the Rajant radios offered the possibility to connect both an ES1 and a DX2 to the Base Station via wired connection without creating any network loop, which was not possible with the previous technology.

Deployable breadcrumb nodes
Carrier ANYmal robots ferried additional WiFi breadcrumbs around the course and deployed them to incrementally expand the reach of the wireless mesh network. Carriers could transport four wireless breadcrumb modules and deployed them by lowering their torso and using an electromagnetic release mechanism (see Figure 17b), as described in Section 3.1.1. Each breadcrumb was a water and dustproof self-contained unit, consisting of a 2 h battery pack, a battery management system, a magnetic switch (to switch the module on after releasing it), an LED to aid the recovery of the module after a mission, a WiFi patch antenna, and a "BreadCrumb DX2" radio (see Figure 17a). The patch antenna was installed on a compliant, 3D printed part that could be folded to allow the stacking of multiple beacons when loaded in the release mechanism.

Once the electromagnet holding a beacon was deactivated, the antenna mount would release tension, helping the module to slide down a carbon fiber tube and open in the upright position. The modules had four landing feet, with padding material on the bottom to absorb impacts when falling to the ground after release. Each beacon weighed 245 g and was 120 mm x 60 mm x 50 mm in size with the closed antenna mount. When standing on the ground, the antenna extended 17 cm upwards. We did not automate the deployment of these beacons. During the Final Event, the Human Supervisor made the expert decision about where and when to deploy the breadcrumb nodes as wireless signal propagation is often unpredictable in subterranean scenarios.

Nimbro network for multi-master ROS networking
Above the physical layer, we employed the Robot Operating System (ROS) (version 1) as the middleware for message passing between processes running on each robot, as well as across robots on the same network. Each robot ran their own onboard *rosmaster* process, which monitored all internal communication connections. Given our use case scenario with multiple robots and the Base Station, each with their *rosmaster*, we used the Nimbro Network[3] software package to allow multiple such instances on the same network.

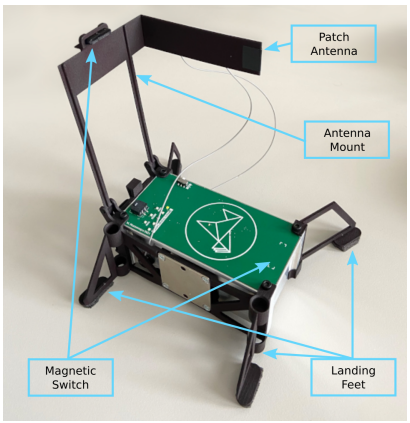
This software package was configured to select individual data streams which were transmitted between the agents and the Base Station. We additionally implemented an option to distinguish between essential and non-essential topics. Essential topics included the robot's heartbeat and telemetry, artifact reports, and mapping data. Non-essential topics included camera streams and other data not strictly needed to be sent to the Base Station for the mission execution. All robots were allowed to stream essential topics to the Base Station continuously and could stream the non-essential ones only if selected by the Human Supervisor. This option, developed for the finals, prevented unselected robots from sending not-essential and bandwidth-hungry data when not specifically desired.

One possibility we considered was to use ROS2 and its native support for multi robots systems and Quality of Service (QoS) to manage the bandwidth requirements between robots. We decided against mixing ROS1 and ROS2 for simplicity.

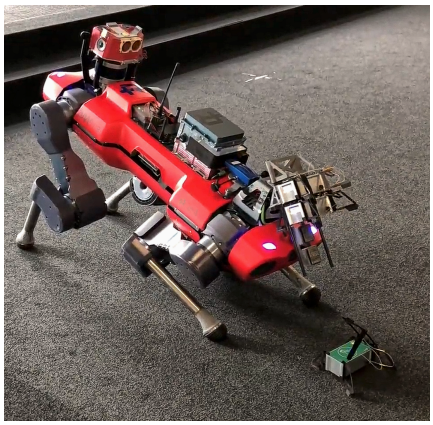
3.5 Artifacts scoring

Each robot performed artifact detection onboard using a trained neural network similar to [Redmon and Farhadi, 2018]. The trained object detector provided bounding boxes identifying the correct class of the artifact from image streams collected by the cameras onboard the robots. Because of their small size or nature, the Phone, SubT Cube, and CO_2 gas artifacts were detected using dedicated sensors that measured

[3] `https://github.com/AIS-Bonn/nimbro_network`



(a) WiFi breadcrumb.



(b) WiFi breadcrumb deployment.

Figure 17: Communication-extender modules and ANYmal C SubT Carrier. (a) Assembled WiFi breadcrumb module. A WiFi patch antenna is attached to a compliant foldable mechanism. When loaded on the robot, the antenna is folded down and under tension. When being deployed, the compliant structure pushes the module out of its magazine. On the ground the antenna unfolded into position for better reception. (b) Carrier ANYmal tilting its torso and deploying a WiFi breadcrumb on the ground.

the Bluetooth strength and CO_2 gas concentration. Multiple detections of the same artifact were fused to refine its estimated position. When sufficient confidence was reached, a single report was sent to the Base Station over the communication network.

Training

To train the object detector, we collected images using various camera sensors in urban basements, abandoned tunnels, mines, and natural caves, with lighting conditions ranging from low ambient lighting to complete darkness, depending only on the robot's onboard light sources. Datasets were acquired in the presence of obscurants and with a wide variety of background colors and textures. Different types of cameras and lenses were used on the robots when collecting data which contributed to a diverse common training dataset. We observed that training the detector with images obtained using different sensors at different resolutions improved the detection accuracy.

During the training process, special care was taken to identify conditions where the detector failed and to collect images for the detector to perform better in similar scenarios. Figure 18 shows some of these cases. Examples included the detection of artifacts against similarly-colored backgrounds, presence of motion-blur, obscurants, and artifacts located distant from the robot and against cluttered surroundings. These cases were identified through continuous field testing and evaluation on datasets. The collected images were manually filtered to remove repetitive images to prevent over-fitting. The images that had significant amounts of obscurants or extreme motion blur were identified and removed. In total, 40 007 images containing artifacts were labeled manually. Table 1 shows the number of labeled instances per artifact. Despite using labeled images for the "Cell Phone" in the training data, we used Bluetooth as the primary method to detect the artifact. In addition to these, we collected a set of 384 images containing no artifacts, but instead containing challenging environments and specific conditions that were found to cause false-positive detections. The training set consisted of 90 % of all the images while the remaining 10 % were used for validation. The neural network was trained on this set for up to 16 000 epochs, and the best performing parameters for training were chosen based on their results on the validation dataset.

Inference

We adapted the trained neural networks to run on dedicated hardware on each robot. For the DJI Matrice 100-based aerial scouts and Kolibri platforms we used the Intel Movidius Neural Compute Stick 2 to perform inference at 5 Hz per camera stream. For the RMF-Owl, the weights were converted from 32-bit floating-

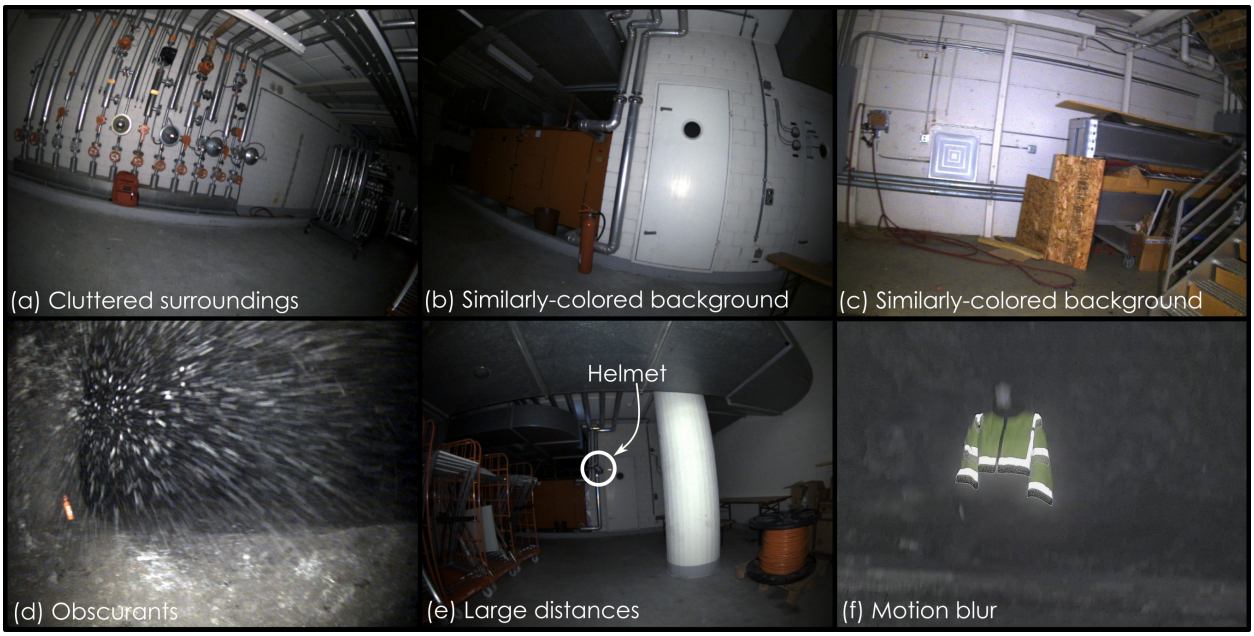


Figure 18: Specific examples where the detector regularly failed were collected to improve performance. Images were collected in cluttered surroundings (a-c) with distracting colors in the background or clutter similar to the artifacts to improve detection accuracy. Images were also collected in the presence of obscurants (d), with artifacts at large distances (e), and with the camera in motion to induce blur (f), as would be observed by the robot in real-life deployments.

Training Images for the Artifact Detector	
Artifact Class	**Number of Labels**
Survivor	6477
Fire Extinguisher	5357
Drill	4580
Backpack	3415
Vent	5350
Helmet	6053
Rope	5219
Cell Phone	3556
Total	40 007

Table 1: Number of labeled images used to train a neural network using YOLOv3 for each class of visible artifact.

point numbers to 8-bit integers to run on the NPU of the Khadas VIM3 computer at a rate of 3 Hz. ANYmal C used a Jetson AGX Xavier and NVIDIA's TensorRT SDK. This module used 16-bit floating-point numbers and batch processing, achieving an inference rate of 3 Hz for each of the four color cameras, while running alongside the elevation mapping pipeline on the Jetson. For the roving platform, the dedicated onboard GPU comprising of a GeForce 1070 graphics card from NVIDIA ran four instances of the object detection module, one for each of the onboard color cameras, and achieved an inference rate of 5 Hz per camera stream.

Detection filtering

For the operator, the object detector overlaid rectangular bounding boxes around the object with the probability of a correct detection as shown in Figure 19. Because detections could be noisy or inconsistent detections, we filtered the artifact position estimate. Individual rays were cast into the volumetric map built by the exploration planner for each pixel inside the bounding box using the camera parameters. To estimate the position of the object, the median point of intersection of these rays with the map was calculated. A virtual sphere was constructed, centered at this median point. Further measurements from subsequent images that estimated the artifact's location within this sphere were considered to be detections of the same object. The sphere center's position was updated by averaging all the accepted detections. With each new detection, a separate binary Bayes filter recursively updated the probabilities for each artifact class. Once the probability for an artifact class exceeded a predefined threshold, the process was stopped, and the updated center of this sphere was reported to the Base Station as the position of the artifact. Subsequent detections of the artifact on the same robot were ignored to prevent duplicate reports. If the robot was out of communication range of the Base Station, the artifact's class and location was stored on the robot and was transferred to the Base Station once the communication was re-established.

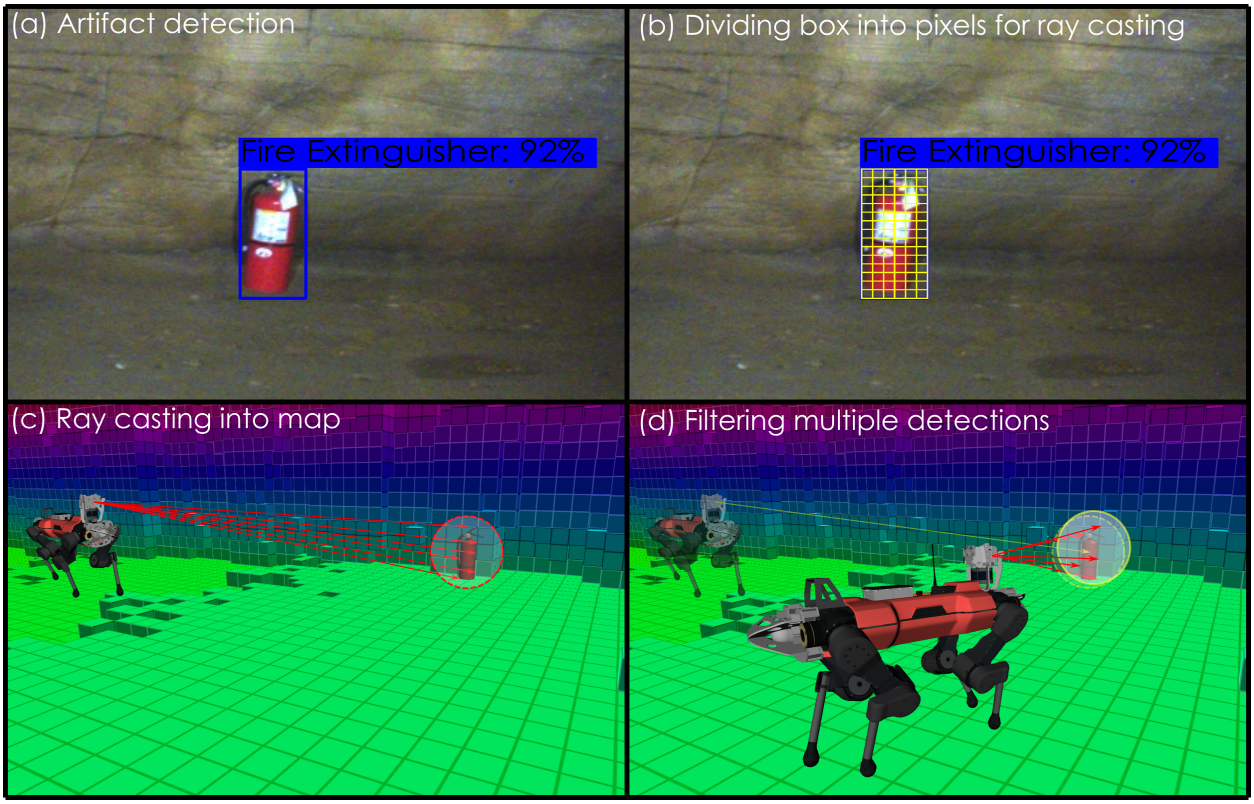
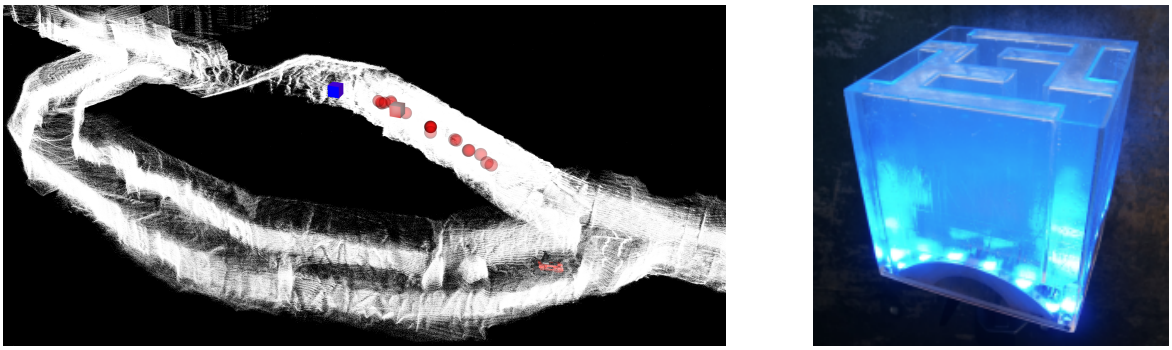


Figure 19: Artifact detection methodology employed by Team CERBERUS for artifacts detected by YOLOv3. (a) The artifact is detected with the bounding boxes around it and (b) the area inside the bounding box is divided into pixels. (c) For each of these pixels, rays are cast into the volumetric map of the robot to locate the artifact. (d) Multiple detection from the same robot are filtered to refine the position of the artifact.



(a) Clustering of measurements (b) The SubT cube artifact

Figure 20: Cube artifact detection. (a) The raw Bluetooth measurements are shown as red spheres, located on the positions where the robots sampled them. A brighter color indicates a higher signal strength at the marked position. The red cube shows the cluster position, whereas the blue cube displays the artifact's ground truth position. Because the robot didn't actually pass by the cube, its position estimate was not precisely accurate, nonetheless our estimate was still inside the 5 m scoring radius resulting in a successful artifact report. (b) The SubT cube artifact.

Bluetooth and CO_2 detection

Bluetooth and CO_2 readings were processed by the same software module by normalizing the raw sensor measurements to a predefined range. Bluetooth measurements were clustered by their SSID identifier, whereas CO_2 measurements were clustered by distance. When a new raw measurement above a certain threshold was received, a new measure cluster was created. Each cluster's center was continuously updated using the largest measured values. A cluster's center was reported to the Human Supervisor when enough measurements about a predefined threshold were received. Both the SubT Cube and the Phone artifact were detected by Bluetooth localization; visual detections generally unreliable in the latter case. An example of clustered measurements used to obtain an estimate of a SubT Cube artifact position is shown in Figure 20.

Position optimization

M3RM (Section 3.2.4) continuously estimated the optimized position of detected artifacts. Additionally to estimate the global artifact position estimated by CompSLAM (Section 3.2.2) onboard the robot, each artifact report contained the position of the artifact in the occupancy map, expressed in the robot frame at the time of detection, as well as the respective timestamp. For each artifact, M3RM then estimated the position of the reporting robot at the detection timestamp. The global optimized position for each artifact was calculated by transforming the position of the artifact with respect to the robot frame into the global frame using the optimized pose of the robot. The position estimate for each artifact was updated after each full optimization cycle of M3RM.

Inspection payload

On the ANYmal Explorer robots we employed the payload developed by ANYbotics' for inspection. This featured a 10x optical zoom camera, a thermal camera, a directional microphone, and a spotlight. The unit can pan and tilt its sensor suite to provide a range of 270° for the pan joint and 180° for the tilt joint. The purpose of the moving sensor unit was to complement the static cameras of the Alphasense Core, which have a large field of view but cannot scan the distant surroundings and did not have thermal imaging. In contrast, the inspection payload's zoom camera had a smaller field of view but could spot far-off artifacts. Its thermal camera captured the thermal signatures of the survivor and the vent artifact.

Object detector - We used the same image detector for the zoom camera images as for the Alphasense Core images, without any modifications. We implemented a dedicated detector for artifacts with thermal signatures (Survivor and Vent). For each received thermal image, the detector clustered pixels that were in the same temperature range as specified by DARPA for the Survivor artifact or 10 °C above the ambient

temperature (defined as the average of all temperature values of the last three images). If enough adjacent pixels were in a similar temperature range, a thermal artifact was considered detected and sent to the Human Supervisor for further inspection.

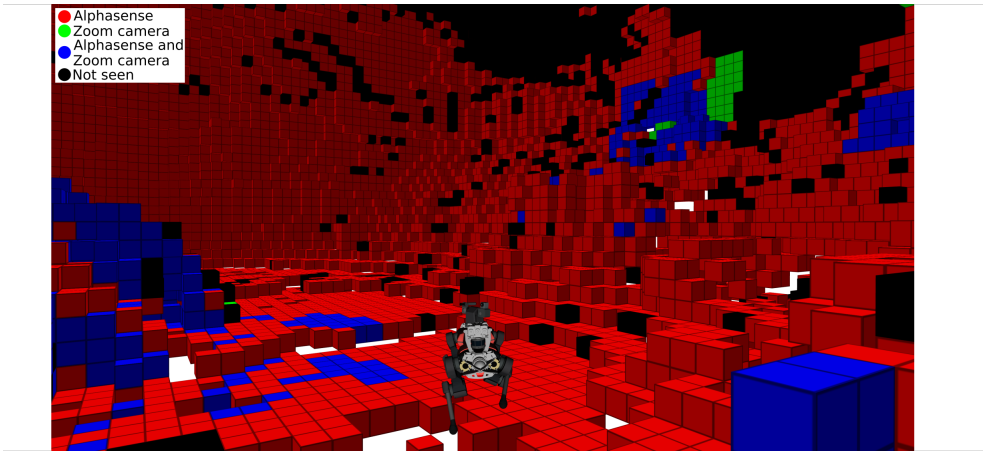
Scanning operation - For any requested point to scan, the inspection payload's controller adjusted the yaw and pitch angles so that the inspection module's sensor suite pointed to the desired location. It changed the zoom level of the camera based on the distance to the area containing the point to scan so that if any artifact were present in the scanned area, it would have similar relative dimensions to the camera's resolution. Adjusting the zoom level based on distance was possible by performing ray-casting in the global volumetric map created by the exploration planner. The inspection payload featured four different operation modes: "Operator Point Scan", "Operator Full Scan", "Autonomous Point Scan", and "Autonomous Full Scan". The Human Supervisor could trigger either "Operator Point Scan", to request the inspection of a given position or "Operator Full Scan" to request the scanning of the whole area (composed of multiple points scanned in sequence) surrounding the robot. The other two modes, "Autonomous Point Scan" and "Autonomous Full Scan", had the same functionalities as their "Operator" versions but were triggered by an autonomous scanning algorithm.

Autonomous scanning - The autonomous scanning algorithm relied on the volumetric map built by the exploration planner to compute points to scan with the inspection payload. Areas in the map closer than 5 m and within the field of view of one Alphasense Core's color cameras were marked as seen. Afterwards, the algorithm computed unseen areas with respect to the inspection payload's joint space, which was set to $-135°$ to $135°$ for the yaw joint and $-10°$ to $30°$ for the pitch joint, with a resolution of $3°$ for both joints, resulting in a total number of 1170 potential scan points. For each possible scan point, we performed ray-casting to determine where each ray intersected the global map. We marked each intersection with an identifier indicating whether it was unseen, and in that case we calculated the zoom camera's distance to that point. Both the zoom camera's distance to the nearest voxel and the already inspected area were considered to select the least number of scan points while still covering the entire workspace. An example of the selection of new points to scan is depicted in Figure 21, where it can be seen that no points are selected on already seen parts, and fewer scans are performed when the voxels are closer to the zoom camera. If the "Autonomous Full Scan" mode was enabled by the Human Supervisor, the scanning algorithm checked if the unseen area exceeded 50 % of the reachable workspace and, in case, sent a request to the ANYmal's behavior tree to execute a full scan of the area. The behavior tree module then paused the currently running action, stopped the robot, scanned all points provided by the autonomous scanning algorithm, and resumed the previously running action. If the unseen area was less than 50 % of the reachable workspace, the "Autonomous Point Scan" was executed, where the selected points were scanned while the robot was walking.

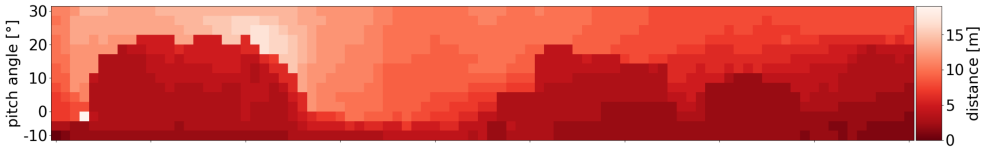
3.6 Single human supervisor

A characteristic aspect for the SubT Challenge was the requirement to have a single operator, called the "Human Supervisor", as the only person permitted to use wireless (or other) communications to interact, coordinate, and manage the deployed robots during the competition run once they had entered the course. The single operator requirement was intended to minimize the amount of human intervention and pushed the competing teams to develop a high level of autonomy for remote mapping and navigation. The Human Supervisor was allowed to operate at a Base Station external but close to the course's entrance. The Base Station was a computer that served as an interface between the supervisor and the robots. Moreover, this computer directly connected to the DARPA Command Post. Each team had to send regular map updates and artifact reports to the Command Post. Artifact reports were automatically evaluated and scored if the reported positions were within five meters of the objects' ground truth.

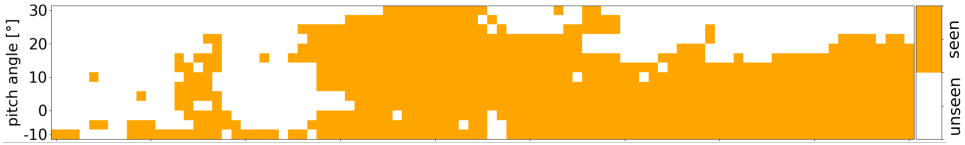
Our experience in the first two competition events showed that errors made by the operator were a significant fraction of mistakes during the runs (due to the general pressure the competition, controlling multiple robots via several computers and not enough practice operating in unknown environments). These findings are similar to those reported by several teams in DARPA Robotics Challenge [Atkeson et al., 2016]. In the



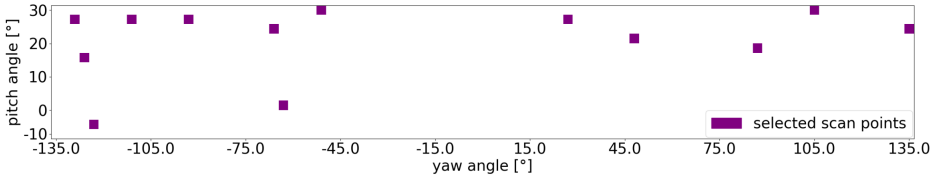
(a) Volumetric map classified based on seen voxels.



(b) Robot's distance to voxels.



(c) Already scanned area



(d) Scan points with seen area

Figure 21: Selection of scan points for the ANYmal C SubT Explorer's inspection payload. (a) Visualization of the volumetric map created by the exploration planner: red voxels were seen by the Alphasense Core's color cameras, green voxels were already scanned by the zoom camera, and the blue voxels were inspected by both Alphasense Core and inspection payload, black voxels are considered as unseen. Voxels in 3D are converted to a 2D image. (b)-(d) The x and y axes represent the inspection payload's workspace. (b) Distance from the zoom camera to the nearest voxel for each potential scan position. (c) Already inspected areas by the Alphasense Core and the zoom camera are marked in orange and unseen areas in white. (d) Final selected scan positions, displayed in purple, are obtained considering the already scanned area and the distance to the nearest voxel.

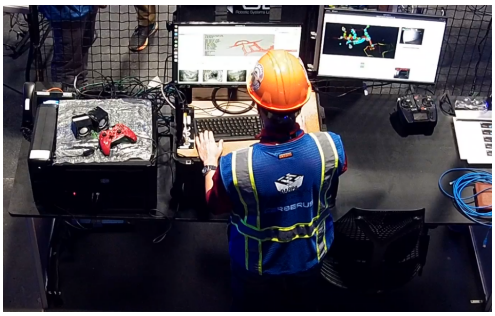
Tunnel and Urban events, our Human Supervisor operated different computers to interact with different robots and to handle mapping and artifacts information. Building on the lessons learned from those events, we developed an improved version of the human-robot interface before the Final Event. A single computer was used to control all the robots and to aggregate mapping information and artifacts reports. The supervisor used two main software modules to interact with the deployed robots: a Mission Control Interface (MCI) and a Monitor and Artifacts Reporting Interface (MARI), exposed through a Graphical User Interface (GUI) (see Figure 22). Both interfaces ran on the same computer at the Base Station and used a multi-master architecture with the Nimbro Network software package (described in section 3.4) to communicate with the deployed robots.

Mission control interface
Through the MCI (Figure 22c,22d) the operator could select and interact with a specific robot. Specifically, the supervisor could (a) give high-level commands, (b) visualize throttled and compressed image streams to either inspect an area or manually report artifacts, (c) request specific sensor data to evaluate the current situation better ("on-demand" requests). The supervisor had several options for high-level command and control. We include here a list of the most relevant ones: start autonomous exploration; define the exploration bounding box to restrict the area of interest; start homing procedure; cancel current task; relocate to a selected frontier and start exploration from there; scan surrounding area or selected point with the inspection payload (ANYmal C Explorer robots specific); move to a desired waypoint. Only the selected robot in the MCI was allowed to send image streams to the Base Station, as specified in section 3.4. Allowing only one robot to stream images helped the supervisor to focus only on one robot per time and also decreased the wireless bandwidth utilization. Finally, the operator could request additional sensor data to better evaluate the robot's current state and surrounding area. These data included: CompSLAM map, LiDAR scans (finer resolution than the CompSLAM map) and elevation map snaphosts (legged robot specific).

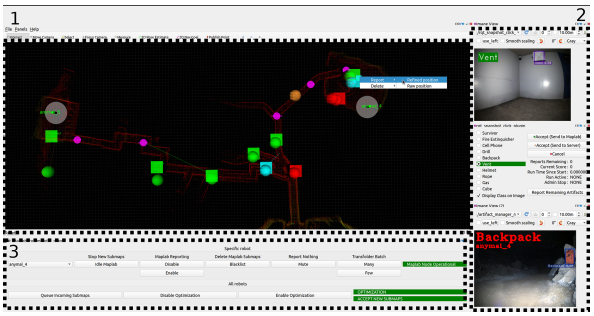
Figure 22c shows the MCI for the ANYmal C robots and highlights three main panels: an RVIZ view (1) showed essential data such as a frame representing the robot's pose in the map and the requested sensor data by the supervisor; a bottom panel (2) displayed the current image streams sent by the selected agent; a side RQT panel (3) exposed control buttons to the supervisor to interact with the selected robot. Similarly, Figure 22d shows the MCI for the aerial and roving robots. Two panels are highlighted in this figure: an RVIZ view (1) similar to that of ANYmal; a side RQT panel (2) containing the control buttons for interacting with the robot.

Monitor and artifacts reporting interface
The MARI (Figure 22b) was the main interface to provide situational awareness to the Human Supervisor. This GUI was divided into two main parts, one for artifacts reporting and one for monitoring of the mission state. The artifacts part contained a panel notifying the supervisor of all new automatic artifact detections. If a robot sent a new artifact report to the Base Station, the Human Supervisor was notified with a sound. The associated camera image was shown for context, annotated with the artifact's bounding box and class. The operator could either accept, reject or modify (class type) the artifact report. The monitoring part presented a top-down view of the global map created during the mission and some extra markers. Cylindrical markers indicated the last received positions of all the robots. Spherical markers in the map showed the locations of the accepted artifact reports. The operator could click on each of these artifacts' interactive markers and either send them to the DARPA Command Post to try to score a point or delete them. Moreover, in the monitoring part, the operator could visualize where each robot was and what was its primary mode of operation (exploration, waiting for commands, homing, etc.). The goal of this view was to provide high-level information and help the operator's decision-making process.

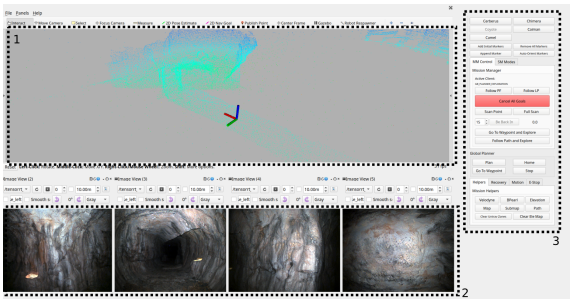
Figure 22b highlights the MARI's three main panels: an RVIZ view (1) showed the global map, the last received position of each deployed robot and the locations of the accepted artifact reports, represented by sphere and cube pins. Spheres represented unoptimized locations reported by the agents, whereas cubes represented optimized positions. Each artifact report could therefore have one or two associated pins, based on whether its position had been already optimized or not. Every artifact report's pin was an interactive marker that the supervisor could use to report its location to the DARPA Command Post or delete it from



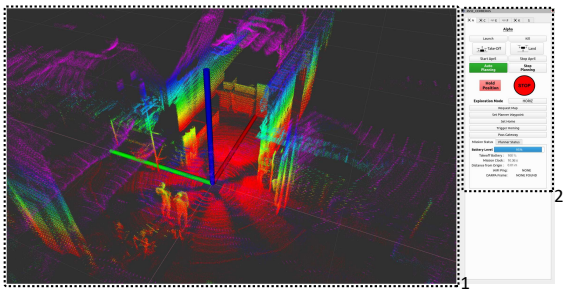
(a) Base Station



(b) Monitor and Artifacts Reporting Interface (MARI)



(c) ANYmal Mission Control Interface



(d) Aerial and roving robots Mission Control Interface

Figure 22: Base Station setup for the single Human Supervisor. (a) Setup of the Base Station at the Final Event with a single computer and two screens, showing the two main GUIs used to supervise the deployed robots. (b) Monitor and Artifacts Reporting Interface (MARI) was the main interface to provide situational awareness to the Human Supervisor. (c-d) Mission Control Interfaces (MCI) used by the Human Supervisor to provide high-level commands to a selected robot, specialized based on agent's type.

the map. Artifacts that were not reported yet were colored blue if the optimized location was available and orange otherwise. After a report attempt, the associated pins turned green if a point was scored and red in case of an unsuccessful report. The locations of deployed breadcrumb nodes were indicated in purple; a side panel (2) showed incoming artifact detections (top). The user could accept or reject the detection and manually change the artifact class in case the detected class was not correct (middle). Upon acceptance, the interactive marker of the artifact was added to the global map (1). When the user selected an interactive marker in the global map, its corresponding detection image was shown in the lower right corner; (3) allowed enabling and disabling of specific robots in the optimization as well as changing several settings of it.

Modes of operation

Team CERBERUS envisioned a squad of highly autonomous robots coordinated by human reasoning to drive the exploration of unknown environments. As a result, the deployed systems worked in a "Supervised Autonomy" mode. Supervised meant that the single Human Supervisor used the information available at the Base Station to take high-level decisions and assign tasks to each robot toward one collective exploration strategy. Each system then undertook the execution of its task autonomously. Specifically, at each moment, the operator evaluated the current high-level situation (global map, position, and mode of operation of each robot) through the MARI. If the supervisor wanted to get more information (camera streams, sensor data) of a specific robot, then they selected that agent in the MCI and requested the additional information.

Furthermore, through the MCI, the supervisor could assign or change each robot's task. First, a robot could be tasked to explore a specific unknown area while ensuring it remained within communication range. If it went beyond the communication range after a certain step, it would backtrack to a point with connectivity. Second, a robot could also be tasked to explore unknown regions away from the communication range, given a set time budget after which it would return to the connection range. A third mode allowed the Supervisor

to command the robot to reach a frontier close to unmapped areas and trigger autonomous exploration. Last, as a last resort, the operator could provide a high-level waypoint in the map and request the robot to find an admissible path to reach it.

The part of the Human Supervisor shifted from an "operator" role during the Tunnel Event, telling each robot where to go and how to go there, more and more toward a "supervisor", coordinating a group of robots without focusing on the motion of a single agent. This mode of operation incorporated our vision of "Supervised Autonomy" for the SubT Challenge.

4 Results

This section focuses on the results from the deployment of Team CERBERUS in preparation of and during the Final Event of the SubT Challenge.

4.1 Individual experiments by subteams and preparation

We performed a series of field tests to evaluate the performance of our walking and flying robots system-of-systems and prepare for participation in the Final Event. Reflecting the geographical organization of our team, we executed several field deployments spanning four different countries (USA, Switzerland, Norway, and UK). We explored different types of underground mines, subterranean urban infrastructure, and cave environments.

4.1.1 Field deployment preparation

From the beginning of the SubT Challenge, we continuously improved our field deployment and software testing procedures to ensure the robots were ready to execute periodic mock-up missions. We employed different tools and strategies to achieve that. On one hand, we used modern development tools such as "git" for version control and used Continuous Integration (CI) pipelines for automated testing of the individual software modules. While these software development approaches were important, field deployment is robotics specific. Our approach to field testing changed according to our experience in the three years of the competition. As CERBERUS was a diverse international team with expertise across countries and continents scheduling of major field testing events was of paramount importance to integrate disparate contributions and build a team spirit. The following paragraphs contain the most relevant findings we gathered during this time.

Weekly shakeout
Regardless of the frequency of the field deployments, we executed a general functionality test of core systems once a week. We referred to this event as a "weekly shakeout". We introduced this procedure after the Tunnel Circuit event and committed to testing the overall functionalities of our robotic systems once a week in a more straightforward but representative enough scenario. Weekly shakeouts usually took place on Fridays and involved a subset of all the robots. The involved members updated the software the robots used for testing. First, team members performed a functionality tests in the lab, where they verified the functionality of the basic features (autonomous planning, SLAM, artifacts detection and reporting, etc.). After this first step, the team and robots moved to a more representative but still easy to access location, such as a university basement featuring multiple rooms, corridors or a garage. The testing team verified that the main software and hardware components worked during a full mission mock-up. Even though these environments were relatively small compared to the actual scale of the SubT Challenge environments, we could execute mission runs lasting circa 20 min and collected valuable data about the overall status of our deployed solution.

Field deployment

To gather as much experience in the field as possible and to develop a unifying robotics solution for the different and diverse types of underground settings (tunnel-like, urban, and cave networks), we tested in various subterranean environments in preparation for the finals. Figure 23 shows a selection of the different environments we tested in, whereas Table 2 includes the list of extended (daily) field deployments. We divided the field deployment into two main phases: "pre-deployment" and "deployment".

Pre-deployment - This phase started roughly one week before a scheduled field test. Firstly, the team agreed on which new features had to be tested in that deployment. Secondly, we did an integration test of the new desired features one day before the field test as part of a mission shakeout. Each robot had to pass the shakeout individually to be cleared for the field test. If the team detected a malfunction during the shakeout phase, they could either try to solve it or decide to use a previous, tested version of the same module, without the detected issue. If the team performed either a software fix or a downgrade, all the robots had to switch to the same software version and be tested again. Once every robot passed the shakeout test, it was marked as "mission-ready" and allowed no new hardware or software updates. With these constraints, the team ensured that on the field deployment day, they were ready to go to the selected underground location and start testing without any hindrances due to unforeseen issues related to the latest software or hardware updates. After all the robots were ready and packed, we discussed and agreed on the field deployment schedule and wrote down a testing plan.

Deployment - We scheduled at least one field deployment in an underground setting per month in a different location to avoid overfitting to a specific site (underground mine, urban environment, cave networks). Each field deployment included several mock-up missions in a configuration as similar to the SubT Challenge runs as reasonably possible. Specifically, the team set up a mock-up gate with four Apriltags so that each robot could execute the "DARPA-frame" detection before starting its mission. Moreover, we prepared the Base Station, where the Human Supervisor coordinated the mission runs, as required during an actual competition event. Finally, we distributed the artifacts along the course to verify that the robots could detect them and report their locations. Before each mission test, the Pit Crew members prepared the robots with the same modalities used for the official missions. In the year before the Final Event, we had two people training as the Human Supervisor. Each field deployment happened despite the status of the features under development, and its primary goal was to assess the overall mission performance. Each executed mission was a snapshot of the system's readiness in a competition-like environment and provided essential information to the team. Furthermore, each field test contributed to the team's readiness for the final event.

System launch
Before each mission, the Pit Crew launched the software in each robot using ssh-shell sessions running *tmux*[4]. Tmux is a terminal multiplexer used to create persistent sessions on each robot where the launched software kept on running even if the original ssh-shell session disconnected. Moreover, we could save the complete output history of each onboard session to file for post-processing analysis and debugging of every mission.

Onboard data recording
Robots recorded extensive data during their mission executions for post-mission analysis. These data included mission-related information (such as robot state and odometry, onboard map, planned paths, etc.), as well as raw sensor data (LiDAR packets, camera streams, etc.) and other onboard compute information (for example, CPU, memory, GPU usage, and temperature). Given the complexity of the software running onboard, we had to carefully select which data to record since recording all the available information for possibly more than 60 min (one entire mission duration) was not feasible due to onboard memory limitation and processing overhead created by the recording process itself. Therefore we focused on the information needed to perform post-mission analysis and debugging and privileged, when possible, "raw information" required to reconstruct the robot state in post-process.

Mission data analysis
After every mission test, we thoroughly analyzed the onboard sessions' logs and recorded data from each

[4] `https://github.com/tmux/tmux`

Field Deployments				
Location	**Environment**	**Country**	**Period**	**Deployments**
Wangen a. d. Aare - ARCHE	Urban	Switzerland	2019-2021	15
Bergwerk Gonzen	Tunnel	Switzerland	2019-2020	12
Versuchsstollen Hagerbach	Urban/Tunnel	Switzerland	2021	6
Bergwerk Seemühle	Tunnel/Cave	Switzerland	2021	4
Menznau Bunker	Urban	Switzerland	2019	3
Baustoffe Schollberg	Tunnel	Switzerland	2021	2
TRJV Mine	Tunnel	USA	2018-2019	12
Comstock Mine	Tunnel	USA	2018-2020	6
Black Diamond Mine	Tunnel	USA	2019	1
Wampum Underground	Tunnel	USA	2019	1
Applied Research Facility	Urban	USA	2019-2021	12
Moaning Caverns	Cave	USA	2020	1
Great Saltpetre Preserve	Cave	USA	2021	1
Løkken Mine	Tunnel/Cave	Norway	2020-2021	4
Corsham Limestone Mine	Tunnel	UK	2019-2021	6

Table 2: Overall robotics field deployment of Team CERBERUS in preparation for the DARPA SubT Challenge. Every reported deployment counts as a full day, including several mission mock-ups and tests.

robot. This procedure included writing down a summary report highlighting the mission's main outcome and any detected issue that required further analysis. Additionally, we generated a series of plots using the onboard recorded data for each test. We found the tool *plotjuggler*[5] to be extremely helpful for this purpose. For example, by visually inspecting these plots and looking for an unexpected trend, we could detect if any silent bug was introduced (such as a memory leak, abnormal CPU or GPU usage, etc.).

4.1.2 Qualitative results from preparation tests

Different challenges were expected from the diverse environments of interest (tunnel-, urban- and cave-like) including a) geometric self-similarity, b) visual obscurants (e.g., smoke and dust), c) lack of visual texture, d) multi-level settings, and e) wide or extremely narrow environments. This guided our selection of a representative set of specific environments for conducting field tests. We expected challenges in terms of their complexity, traversability, effect to communications, scale and perception. To this end, we conducted field deployments in a set of environments that can be categorized as tunnel-, urban- or cave-like in preparation for the challenge. Table 2 presents a complete list of all the field deployments conducted over the course of the competition (2018-2021) divided by category. A representative subset of these environments are shown in Figure 23, with a top-down view of the 3D maps and an image, respectively. The environments presented in Figure 23-(a-e) are representative of tunnel-like environments, while Figure 23-(f) and Figure 23-(g,h) are representative of cave-like and urban environments respectively. In terms of perception, these environments presented challenges to our entire exteroceptive sensor suite albeit not necessarily simultaneously. For LiDAR, Figure 23-(a, e) presented a sudden change in scale, Figure 23-(c) multiple levels and Figure 23-(a, h) geometric self-similarity while for a visual camera, Figure 23-(a, b, f, g) presented darkness, Figure 23-(a, b) heavy airborne dust and for a thermal camera, cave environments like Figure 23-(f) presented thermal flatness.

[5] `https://github.com/facontidavide/PlotJuggler`

Figure 23: A subset of different underground environments used during the "deployment" phase of field-tests during our participation throughout the SubT Challenge. The first and third rows represent a top-down orthographic view of the reconstructed pointcloud maps of the environments while the second and fourth rows capture a third-person view in these environments. The locations of the image captures are presented in the corresponding maps by a white hollow circle. The challenges presented by each of the environments are discussed in section 4.1.2

4.2 Final event

The SubT Challenge's Final Event took place at the Louisville Mega Cavern in Kentucky on $21-24$ September 2021 and involved eight teams for the Systems track and nine teams for the Virtual track. For the Systems track, as side from Team CERBERUS, the following teams competed: CSIRO Data61, MARBLE, Explorer, CoSTAR, CTU-CRAS-NORLAB, Coordinated Robotics, and Robotika. The competition course included elements from all the subdomains (Tunnel, Urban, and Cave) combined into a unified, integrated environment. The course[6] had a single entrance, composed of an inclined tunnel and a steep turn, followed by an opening containing three different access breaches leading to different sections for each subdomain. Multiple connections and crossover points existed between the subdomain areas. Thus, the final course had a complex, heterogeneous, multi-level morphology exposing the teams to all the different challenges that each subdomain offered. Accordingly, the Final Event represented a grueling challenge combining the mobility, perception, autonomy and communications challenges envisioned by DARPA. In the leaderboard of the Systems track's Prize Round, which took place on September 23, both teams CERBERUS and CSIRO Data61 scored 23 points, followed by teams MARBLE (18 points), Explorer (17 points), CoSTAR (13 points), CTU-CRAS-NORLAB (7 points), Coordinated Robotics and Robotika (both with 2 points). As both CERBERUS and CSIRO Data61 were tied with 23 points each, the tiebreaker rule had to be invoked. This rule denoted that the team which reported its last score the earliest would win: CSIRO Data61 reported their final point with under 30 s remaining, whereas CERBERUS reported our last artifact with more than 1 min left. Team CERBERUS won the DARPA Subterranean Challenge as well as the associated $2 million (USD) prize reward.

4.2.1 Preliminary runs

Two test runs on September 21 and 22 2021, called "Preliminary Runs", preceded the Prize Round and served as a testing bench for the teams before the scoring final run. Each Preliminary Run lasted 30 min and included 20 artifacts. The competition course was re-configured between each Preliminary Run and before the Prize Round. Each team's score during these runs was not used to determine the final ranking in the Prize Round, which was based solely on the Prize Round result. Teams were also not allowed to use any prior map information or other data collected from the preliminary runs.

The afternoon before each Preliminary Run, some team members gathered to discuss the deployment strategy for the upcoming run (how many robots to use, which robot to deploy first, etc.). In the meantime, the other members finalized data analysis and required fixes in software and hardware. Upon agreement on the deployment strategy, each subteam (walking, flying, autonomy, multi-robot mapping) worked together to finalize the system configuration for the run. This work included software updates on the agreed modules and final verification of each system, to be completed within the same evening. Selected team members checked each robot (for instance, to verify that all the LEDs turned on, sensor glass was cleaned, plugs correctly connected, etc.) and subsequently tested our overall system in a small mock-up mission to verify that all the significant functionalities were correctly executed. This mock-up mission resembled the tests performed during the "weekly shakeout" sessions highlighted in section 4.1.1.

Preliminary run - day one
Team CERBERUS deployed three ANYmal C SubT agents and the roving robot during the first Preliminary Run. Our overall strategy for this run was a trade-off between testing our robotic solution in the course while being cautious enough not to damage the agents by following more risky paths in terms of traversability characteristics. Specifically, we used rather conservative settings for the ANYmal's navigation planner to prevent the robots from tackling dangerous but still feasible routes.

We deployed an ANYmal C SubT Explorer, two Carriers, and the roving robot. The Human Supervisor assigned a different area of exploration to each ANYmal based on the information he received from the first deployed robot. The supervisor then drove the roving robot in the course and parked it at the intersection

[6] https://s.ntnu.no/subt-finals-course

with the three subdomain openings to extend the wireless coverage of the agents and support underground operations. The supervisor tasked the Explorer with the autonomous exploration of what he perceived as the Tunnel subdomain area, accessible from the course entrance. Moreover, he allowed this agent to keep on exploring even if it went beyond the communication range. Afterward, the supervisor assigned the first Carrier robot to explore the Urban section. While this Carrier proceeded further, the third ANYmal robot was deployed and set to explore the Cave subdomain. The Explorer autonomously navigated through the Tunnel section and reached the intersection with the Urban area, where one of its feet got stuck in a railway track and caused an unexpected shutdown of the agent due to over-current demands while trying to proceed forward.

The Carrier sent into the Urban area also explored this section autonomously, apart from two occasions where the Human Supervisor explicitly provided waypoints. In the first case, the exploration planner provided waypoints to climb a short flight of stairs, but the ANYmal's navigation planner did not plan a path due to the conservative settings mentioned at the beginning of this paragraph. In the second case, a long descending flight of stairs connected the Urban and the Tunnel environment, but its bottom part was never seen by the LiDAR sensor placed on top of the Carrier, due to its limited field-of-view. Thus, the exploration planner could not provide any path to descend the staircases. Once the Human Supervisor realized that the robot could not proceed further, he provided a waypoint to command the robot downstairs directly. Lastly, the third deployed robot autonomously navigated in the narrow passages of the Cave section. During the first Preliminary Run, Team CERBERUS scored 7 points and led the Systems' preliminary round tie with team CSIRO Data61. Importantly, we learned that the environment of the DARPA Subterranean Challenge final event started with a narrow tunnel. We therefore identified that flying robot deployment from the staging area could be both risky and of limited value.

Preliminary run - day two
For the second Preliminary Run, we decided to start with the same settings as the first run and apply a few modifications to evaluate further which configuration to use for the scored run of the day after. First, we did not allow any robot to explore beyond the communication range for this run. This constraint was an extra safety feature on top of the conservative tuning from the first day: we wanted to make sure to have all the robots available for the Prize Round. Second, we deployed the exploration planner with a modification to the global planning stage. Since the planner had a binary constraint on the inclination of the edges in the graph, we observed that the planner could provide paths passing areas too steep for the robot to traverse safely but satisfy the inclination constraints according to the low resolution volumetric map (0.2 m). The issue was prominent in the global planner due to the static nature of the global graph. Hence, a modification was made that calculated k shortest paths towards the goal in the global graph and commanded the one having the least inclination value on its steepest edge. We also re-tuned the detection thresholds for Bluetooth and gas artifacts.

We deployed two ANYmal C SubT Explorers, one Carrier, and the roving robot. The Human Supervisor assigned each ANYmal to autonomously explore different sections: the first Explorer to the Cave environment, the second Explorer to the Urban environment, and the Carrier to the Tunnel area. The deployed robots explored less unknown space than in the previous run due to the additional constraint of maintaining wireless communication with the Base Station. Team CERBERUS scored 6 points in this run, achieving third ranking in the overall Preliminary Round (determined based on the sum of the team's two runs).

4.2.2 Prize round

The Prize Round took place on September 23, 2021 and consisted of one scored run, lasting 60 min, where each team had to find as many of the 40 artifacts distributed along the course, with the total number of allowed artifacts reports equal to 45. Each team was competing on the same course, one after the other, with particular caution by the DARPA team to ensure that team personnel would not communicate with each other to prevent gaining knowledge from another team's run. This was achieved by "sequestering" team personnel and systems from their garage area until they completed their scored run. As it turned out, the

course was further modified from the two Preliminary Runs, again without the knowledge of the competing teams, with the addition of dynamic obstacles such as a hidden mechanism that would close an opening after a robot stepped in or a falling object from the ceiling triggered by a moving robot passing close by.

The afternoon before the Prize Round, team members discussed the final software and hardware configuration and agreed on the deployment strategy. We decided to use settings similar to the ones used during the first Preliminary Run because we were confident about the overall performance of our robots with these configurations. Moreover, we extensively discussed whether to deploy some last software modifications (in the planning and SLAM pipelines) that could have improved some detected issues, but we decided not to. We believe that these conservative decisions, avoiding last minute (and not extensively tested) changes was crucial for the overall performance during the scored run. We learned during the previous years of this competition how complex robotic systems are and how everything must be extensively tested before deploying in the field. Afterwards, we tested all the robots with a "shakeout" procedure, similar to that mentioned above.

At the Prize Round, we prepared the following robots for deployments: four ANYmal C SubT robots (two Explorers, ANYmal 1 and ANYmal 2, and two Carriers, ANYmal 3 and ANYmal 4), one RMF-Owl aerial robot, one DJI M100-based Aerial Scout, and the roving robot. First, the two Explorers were deployed: the Human Supervisor utilized online feedback regarding the map of the area and assigned ANYmal 2 to explore the Cave section and ANYmal 1 to explore the Tunnel section. When the first two robots were exploring their areas autonomously, the supervisor teleoperated the roving robot in the course and parked it close to the entrance of the three different sections, with its directional antenna pointing toward the Tunnel and Urban openings. Afterward, the first Carrier robot, ANYmal 4, was deployed and assigned by the supervisor to explore the Urban environment. While the three ANYmal robots autonomously navigated their given bounding boxes, the last Carrier robot, ANYmal 3, was deployed and sent to the Cave area.

With 18 min left in the run, the team decided to deploy the RMF-Owl to explore the higher area of the cavity in the cave section. However, after deploying, the robot went up exploring the upper part of the staging area. This was caused due to the fact that there was no provision to add 3D "no-go zones" within the staging area (to indicate to the planner that this area is out of consideration) and an existing but deprecated feature of our software for crossing the gate before starting the exploration was not used. The robot was manually landed by the safety pilot and not deployed in the remaining of the run. During the complete run, the Human Supervisor coordinated the mission execution by checking what each deployed robot was doing (autonomous exploration, frontier repositioning, etc.) and, if needed, requested additional actions. Most of the time the supervisor focused on processing the incoming artifact reports from the different robots. At the end of the 60 min scored run, Team CERBERUS achieved 23 points and used all the 45 report submissions available.

Figure 24 and Table 3 show an overview of Team CERBERUS' Prize Round. Figure 24 details in which mode each robot operated at any given moment in time: mode "Supervisor" indicates that the Human Supervisor directly provided a waypoint in the robot's proximity and asked the agent to go there (directly or querying the ANYmal navigation planner to find a traversable path). Mode "Autonomous" indicates that a robot was executing a given task completely autonomously and could, if necessary, decide to take different actions (for example, switch from exploration to homing, execute frontier repositioning, etc.). Mode "No Operation" shows when a robot was not moving / not operational. An agent was considered out of communication range ("Out of Comms") if it could not ping the Base Station for more than 3 s. A correct artifact report (artifact class and position) sent from a robot to the Base Station is marked with a dark green star in the robot's timeline. The Human Supervisor filtered the reports received by the different agents and decided which artifact (class and position) to report to the DAPRA Command Post. Successfully scored artifacts are marked with yellow stars in the Human Supervisor's timeline. Since each robot operated in the course solely based on the information collected onboard and on the commands received by the Human Supervisor, it was not aware of the data collected by the other agents. Therefore, if two robots crossed the same section, they might have reported the same artifacts to the Base Station (hence the number of "Reported Artifacts" exceeds "Scored Artifacts").

Prize Round Statistics								
Robot	**Run Time**	**Distance**	**Scored Artifacts**	**Operational**	**Autonomous**			
					Explore	Home	Reposition	Recover
ANYmal 1	57 min	240 m	8	31.1 %	75 %	6.9 %	4.5 %	13.6 %
ANYmal 2	60 min	687 m	8	69 %	60.6 %	6.6 %	25.8 %	7 %
ANYmal 3	47 min	311 m	1	34.1 %	53.4 %	11.9 %	20 %	14.7 %
ANYmal 4	50 min	500 m	6	57.5 %	81.3 %	12.9 %	0 %	5.8 %

Table 3: Statistics of the winning Prize Round of Team CERBERUS in the DARPA SubT Challenge. For each robot, the following information are provided: "Run Time" indicates for how long each robot operated in the course; "Distance" holds the travelled distance by each agent; "Scored Artifacts" indicates how many artifacts of the total scored ones were reported by the agent; "Operational" indicates the operating percentage with respect to the agent's time in the course (Run Time); "Autonomous" holds the percentages, split by type, when the agent was in autonomous mode - percentage with respect to the operational time.

The Figure 24 also shows our overall strategy: we decided to let the robots go out of communication range for a limited amount of time in favor of getting them back in connection range, receiving their artifacts reports, checking the area they explored, and finally assigning them a new task. The assigned budget to each agent for exploration could be decided at the moment, but during the Prize Round, it ranged from 5 min to 13 min. A given time budget meant that the robot could explore for that time, and then it had to come back to the starting position, thus resulting overall in a time allotment for the assigned task with a duration double the allocated budget. We believe that such a strategy, involving short and frequent autonomous tasks, with the advantage of the Human Supervisor interacting with the robots at a reasonable frequency, paid off compared to letting every agent go for longer autonomous missions and waiting for them to come back only towards the end of the run. However, our approach still implied a slightly heavier load on Human Supervisor who had to cope with different agents at the same time while checking the artifact reports. Another insight about the mission can be noted in the plot between 15 min and 33 min, when the Human Supervisor did not score any points. Specifically, during this time, the supervisor had to control the deployed robots and, at the same time, check a higher number of artifact reports sent by ANYmal 4. This agent sent to the Base Station 30 CO_2 artifact reports between 10 min and 15 min. These reports accumulated in FIFO list, called artifacts queue, at the Base Station and had to be analyzed by the supervisor. Such a high number of reports for a single artifact type, detected in a short period, was caused by erroneous tuning of the threshold used to check the CO_2 concentration. We set a low detection threshold for CO_2 concentration (1500 ppm, in contrast to the given official DARPA specification of 3000 ppm), intending to detect the presence of gas leaks even if not particularly close to the source and relying on the supervisor to decide further where to send the agent in this case. Unfortunately, such a low value created many reports, even though measurements above the set threshold were grouped based on their positions. In fact, within each group, the estimated clustered position of the artifact was allowed to be updated if new measurements were received. Therefore, the more the robot moved close to the gas leak source, the more measurements were above the threshold, and the more often the clustered position was updated and reported to the Base Station. The plot shows only one "Reported Artifact" marker, despite of how many times its clustered position was updated. A similar problem, even if less severe, occurred between 37 min and 42 min when ANYmal 2 sent to the Base Station 11 CO_2 artifact reports. Finally, the plot shows that in the last 7 min of the competition, the supervisor focused only on deciding which of the already received artifacts to report to the Command Post, thus he did not send any additional exploration goals to the agents.

Artifacts scoring

The artifacts during the Prize Round were detected by the robots using their onboard artifact detection

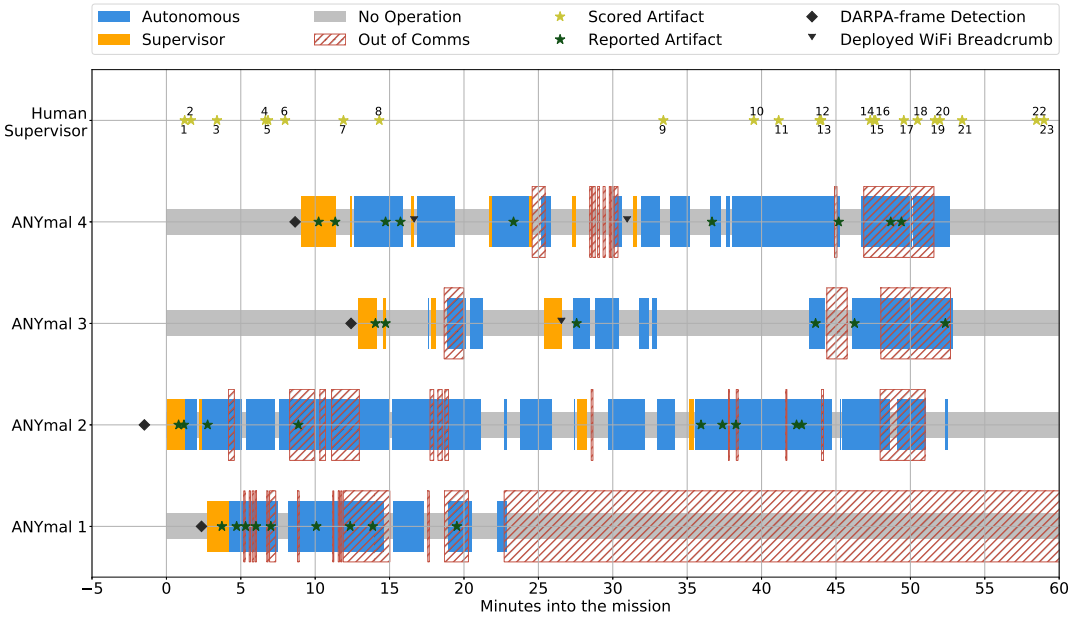


Figure 24: Mission analysis of the winning Prize Round of Team CERBERUS in the DARPA SubT Challenge. The plot shows in which mode each robot operated ("Autonomous", "Supervisor", and "No Operation") and when major mission events occurred ("Reported Artifacts", "DARPA-frame Detection", and "Deployed WiFi Breadcrumb"). An agent was considered out of communication range ("Out of Comms") if it could not ping the Base Station for more than 3 s. An extra line is used for the Human Supervisor to highlight when artifact reports sent to the DARPA Command Post were scored.

framework. For each artifact, the detections were filtered and reported back to the Base Station. The Human Supervisor could see the detections on the MARI as shown in Figure 22b, which was then used to accept, or reject the reports or to change the class of a detected artifact. Team CERBERUS detected 23 artifacts during the Prize Round. These artifacts were detected by the four ANYmal robots, were initially localized based on the CompSLAM solution onboard each robot and the M3RM framework was (at instances) utilized to correct their position estimates on the Base Station. The Human Supervisor added each incoming artifact report from the robot to the map in the MARI and reported the artifacts visualized in the map. For some artifacts reported initially, the Human Supervisor chose to utilize the M3RM solutions, while others were reported using the onboard CompSLAM estimate. Since the M3RM framework required more time to optimize the position as the map grew, this was used in the beginning, whereas later in the mission, the onboard estimate was used.

Figure 25 chronologically numbers the artifacts that were correctly reported to the DARPA Command Post visualized alongside the ground-truth pose of the artifacts. The ground-truth locations of the artifacts were made available by DARPA only after the completion of the Final Event. The ground truth map of the environment provided by DARPA is shown in green. The figure also shows the images captured by the onboard sensors of the respective robots at the time of detection. The Human Supervisor reported the refined positions of the artifacts numbered 6, 9 and 15 as optimized by the M3RM, while the artifact numbered 12 was identified by observing the image streams on the MARI. Artifact localization was initiated onboard the robot for this instance by the supervisor by clicking on the artifact inside the image.

Inspection payload

The main purpose of the Explorer robot's inspection payload was to scan areas classified as being far away from the Alphasense Core's color cameras. Due to the narrowness of the final course, most of the explored sections resulted to be within a close range from the color cameras. Therefore the inspection payload executed only single point scans during the Prize Round. Overall, 25 scan points (cf. section 3.5) for ANYmal 1 and 156 for ANYmal 2 were targeted. ANYmal 2 was inside wider areas than ANYmal 1, explaining the higher number of scans. Scans in narrow passages were either due to imperfect ray-casting in narrow and tangled sections or because the inspection payload captured the area prior to the Alphasense Core. For each scanned point, the camera images obtained with the zoom camera were processed by another instance of the same neural network used for the Alphasense Core's camera. The thermal image samples were processed by the thermal detector at 3 Hz. Figure 26 shows an example of multi-modal artifact detections using the ANYmal's inspection payload module.

Evaluation of the multi-modal and multi-robot perception system

The onboard complementary multi-modal localization and mapping, CompSLAM, and the global multi-robot mapping approach, M3RM, were deployed on all the legged robots during the Prize Round. Each robot individually estimated its state and the map of the environment and, when in communication range, transferred all its information, including submaps, back to the Base Station. However, during the Prize Round the M3RM approach had significant difficulties due to the texture-less entrance of the urban part of the competition course as the primary modality of M3RM was visual imagery. Consequently, for M3RM the majority of the visual constraints in this particular area were incorrect which lead to skewed maps. Hence, M3RM required a significantly larger amount of time to converge to the correct solution. Furthermore, the computational burden prohibited its reliable utilization after a certain point during the duration of the Prize Round. Therefore, the Human Supervisor mainly relied on the onboard CompSLAM pose estimates for the mapping and artifact reporting. Figure 27 shows the maps and paths from CompSLAM for all robots as recorded during the Prize Round (top row). Furthermore, robot map and paths post-processed and aligned with the DARPA-provided ground-truth map are also shown in the bottom portion of Figure 27 to provide a qualitative measure of performance.

To give a quantitative evaluation of the onboard localization during the mission as well as the onboard and global mapping performance that could have been achieved either given different tuning parameters or more time (especially for M3RM), both CompSLAM and M3RM were post-processed and are compared with ground-truth robot trajectories. The ground-truth robot trajectories were created by registering individual

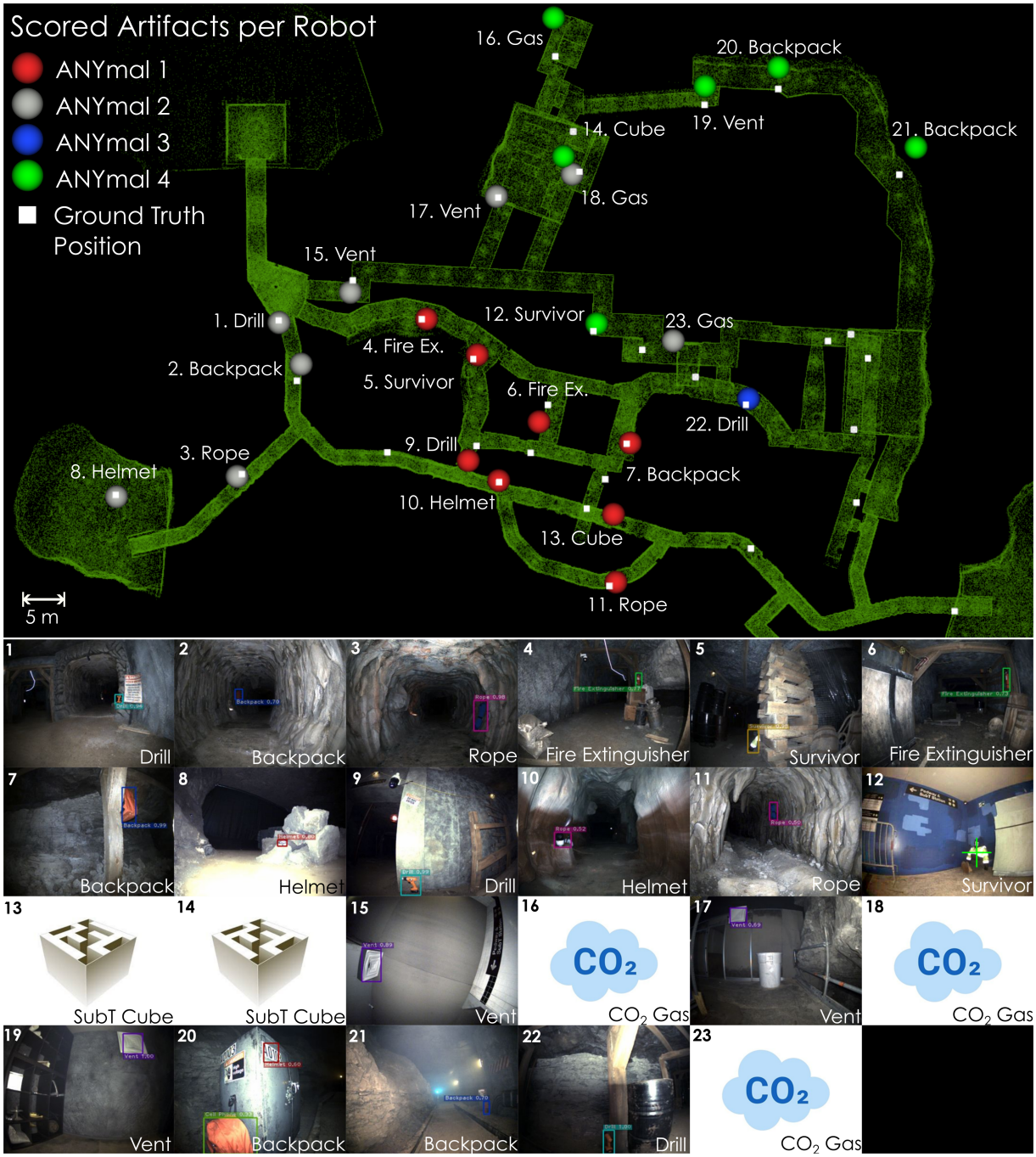


Figure 25: Scored artifacts during the Final Round. The colored circles represent the locations of the artifacts detected by each robot against the ground-truth (in white). The images of artifacts numbered 10 and 20 show erroneous classes. The supervisor accepted the reports from the robot, changed the class types and then sent the revised reports to the DARPA Command Post. The locations of fourteen artifacts were estimated using the robot's onboard CompSLAM output, while those of three artifacts were refined using the M3RM (artifact numbers 6, 9 and 15). Five artifacts involved Bluetooth or CO_2. Finally, the Human Supervisor manually detected one artifact (artifact number 12) while checking the image streams sent from one agent. A video showing the Team's Prize Round is available at `https://youtu.be/QON8IFc8cjE`

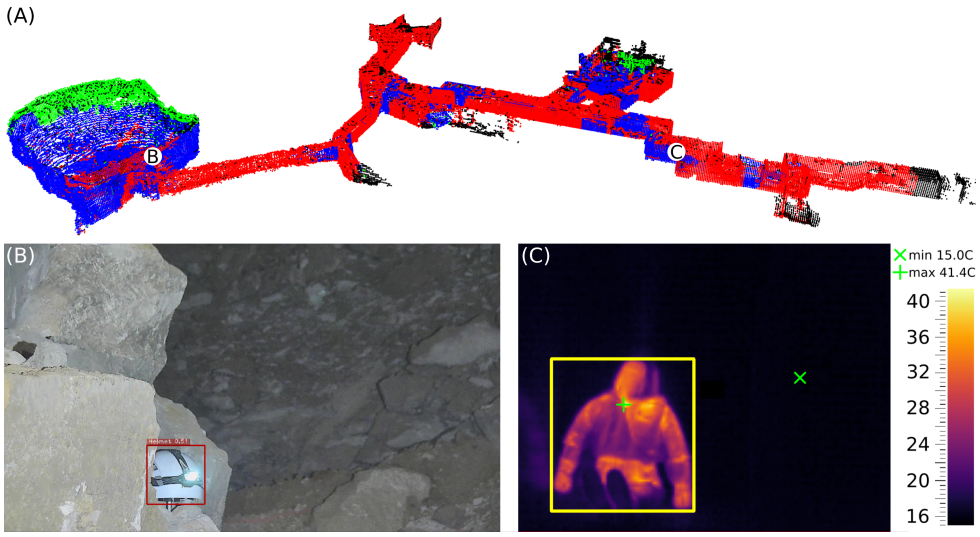


Figure 26: ANYmal's inspection payload artifact detection. (A) The scanned area after the run for ANYmal 2 (Explorer robot). Black points are unseen, red points were only scanned by the Alphasense Core colored cameras, and green points only by the inspection payload zoom camera. Blue areas were scanned by both sensors. Artifact detections performed with multi-sensor modalities: (B) visual detection of an helmet with the zoom camera and (C) thermal detection of a survivor. The letters B and C in image (A) display the point of the respective detections.

robot pointclouds to the DARPA-provided ground-truth map, as described in [Ramezani et al., 2020]. Table 4 presents the Relative Pose Error (RPE) metric for the CompSLAM (onboard) localization during the run and post-processed by comparing against individual ground-truth trajectories. During the competition, as the scale of environment was unknown, the down-sampling leaf-size of pointclouds was increased from 0.1 m to 0.2 m as a safeguard against excessive onboard computational demand, which later given the environment size, was deemed unnecessary. To measure the performance difference, the down-sampling leaf-size was restored for the post-processed results which increased the accuracy of the onboard approach, as demonstrated in Table 4. Since the global multi-robot mapping approach, M3RM, sparsifies the robot poses by using key-frames as compared to estimating the full robot trajectory as done by CompSLAM, the RPE for M3RM cannot be calculated accurately and hence is omitted from comparison.

Similarly, Table 5 contains the Absolute Pose Error (APE) metric for the CompSLAM approach, during the run and post-processed as well as the M3RM (global) mapping approach post-processed. Combining the information from all individual robots leads to obtaining more constraints and hence better constraining the solution and building a more extensive factor graph. Despite the bigger optimization problem with more information, the M3RM solution was able to reduce the onboard drift due to intra- and inter-robot loop closures. In particular, the upward-facing cameras of the ANYmal's sensor rig provided many valuable loop closure detections, especially in passages that were traversed in different directions. As a result, for all final ANYmal runs, the M3RM approach improved the trajectory estimates of the onboard localization and mapping solution and significantly corrected the poses in terms of the APE.

4.2.3 Finals discussion

In this section, we review selected highlights of individual components of our system as well as core challenges that were faced during the Final Event of the DARPA SubT Challenge.

Exploration planner highlights
The Final Event course of the SubT challenge presented settings, particularly challenging for autonomy.

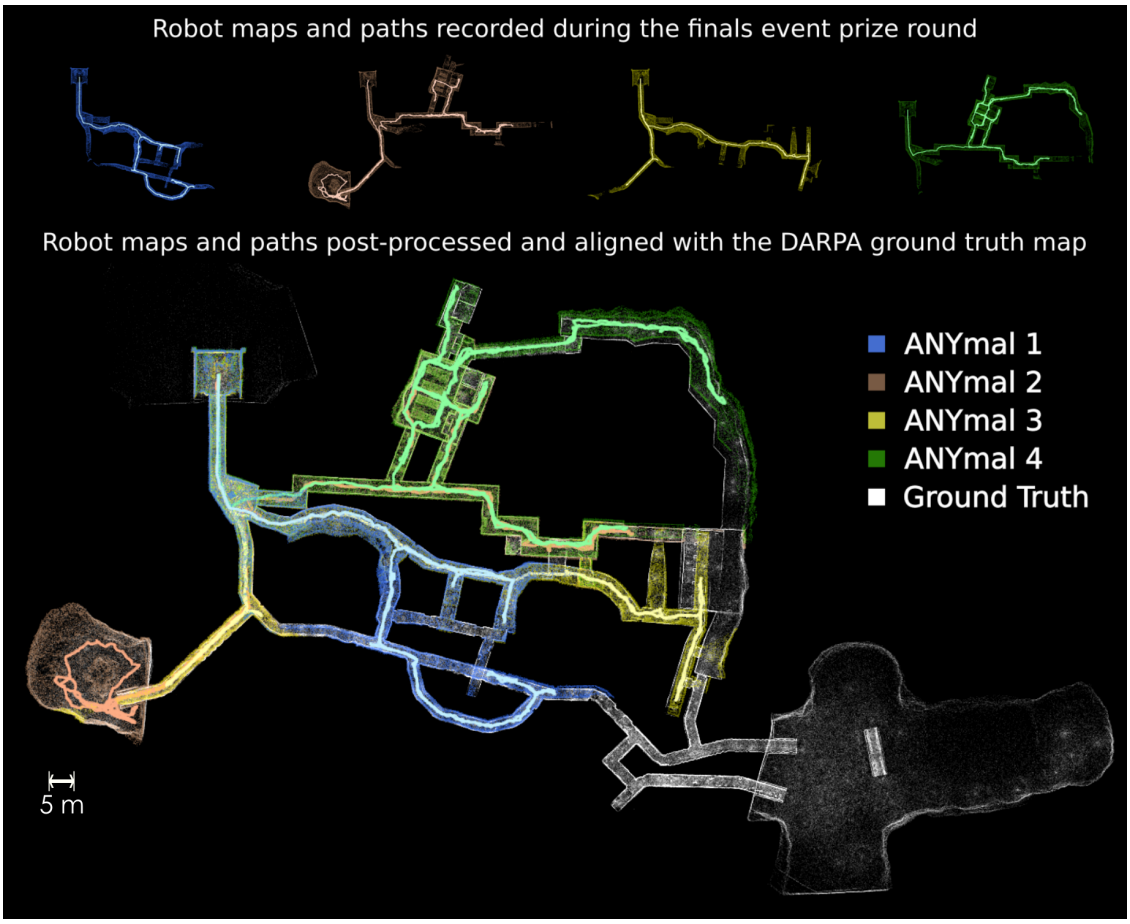
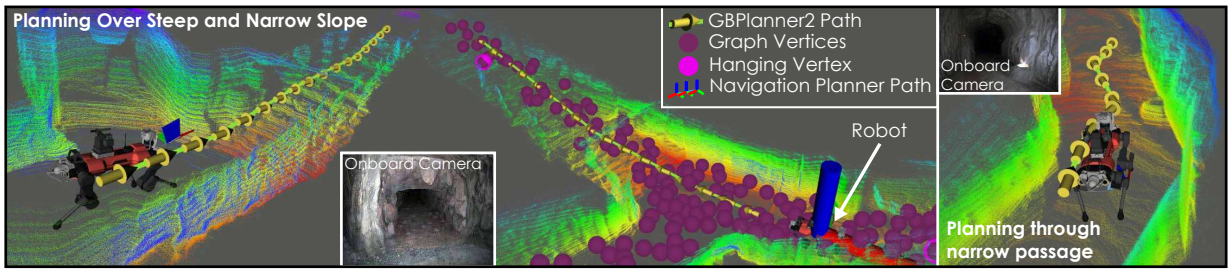


Figure 27: Robot paths and maps generated by CompSLAM and recorded during the Prize Round are shown (top row). The paths and maps were post-processed and aligned with the DARPA provided ground-truth map post competition to provide a qualitative overview of the mission performance (bottom). A video showing the Team's Prize Round is available at `https://youtu.be/QON8IFc8cjE`

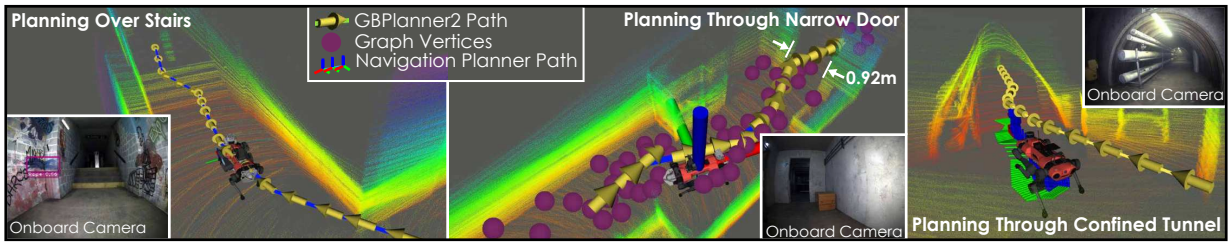
These include stairs, steep slopes, narrow openings, confined passages and big open rooms, and an overall large-scale of the environment. The Graph-based exploration planner was able to provide safe and efficient exploration paths in real-time. The improvements in GBPlanner2 for the ground robots enabled the planner to plan over short staircases and steep slopes. The team had done certain tuning and modification in the planner a few weeks before the competition to enhance its ability to plan through extremely narrow (1 m x 1 m cross-section) openings. This paid off in the Final Event as the course involved several narrow, constrained passages which would have been difficult for the exploration planner. A few notable instances of GBPlanner2 are shown in Figure 28. The figure shows the paths provided by GBPlanner2 in the presence of the challenges mentioned above in the Cave, Urban, and Tunnel sections of the Final Event course during the Prize Round. The figure also shows an instance of global repositioning.

ANYmal mobility highlights

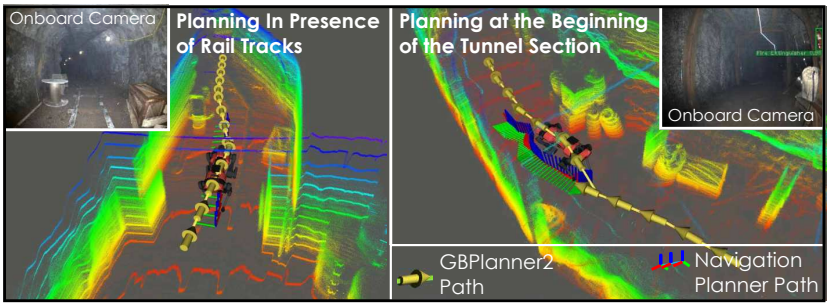
The combination of our perceptive locomotion controller and the navigation planner allowed the four legged robots to safely walk in the most challenging parts of the course (see Figure 29a). Moreover, the navigation planner was able to stop the robot from hitting obstacles which GBPlanner2 missed, as shown in Figure 31. In conditions challenging for the localization and mapping pipeline, the height map can become corrupted due to unstable or too imprecise pose estimates. Corrupted terrain maps could lead to a complete failure to plan for the ANYmal navigation planner. If the robot pose was considered invalid with respect to the terrain map (which should be impossible), the ANYmal's behavior tree module triggered a reset of the height map.



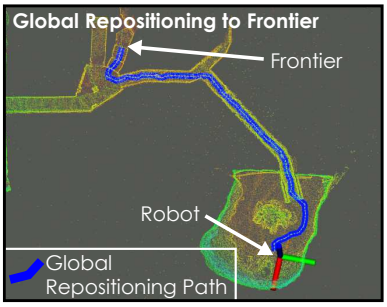
(a) Cave Section



(b) Urban Section



(c) Tunnel Section



(d) Global Repositioning

Figure 28: Highlights of the deployment of the exploration planner in the Prize Round of the Final Event. (a), (b), and (c) show instances of local exploration in the Cave, Urban, and Tunnel sections of the course respectively. The figure shows the performance of the exploration planner in the presence of short stairs, steep slopes, narrow openings and passages. It is noted that in (b), even though the exploration planner planned a path up the staircase, ANYmal did not walk up the stairs due to conservative tuning of the navigation planner. The path commanded by GBPlanner2, refined traversable path by the navigation planner, and the local exploration graph (edges are not shown for better visibility) can be seen along with the front camera view of the robot. (d) shows an instance of global repositioning. The robot had finished exploring one part of the Cave section and the Human Supervisor provided a goal point at the beginning of the Tunnel section. GBPlanner2 calculated a path towards the goal along the global graph.

Prize Round Evaluation - mean RPE and standard deviation				
Robot	**During Run**		**Post-Processed**	
	CompSLAM		**CompSLAM**	
	Rotation [°]	Translation [m]	Rotation [°]	Translation [m]
ANYmal 1	1.21 (0.76)	0.07 (0.05)	**0.69** (0.36)	**0.03** (0.02)
ANYmal 2	0.85 (0.54)	0.05 (0.04)	**0.54** (0.34)	**0.04** (0.03)
ANYmal 3	0.73 (0.41)	0.05 (0.03)	**0.40** (0.20)	**0.04** (0.02)
ANYmal 4	0.89 (0.61)	0.06 (0.04)	**0.56** (0.31)	**0.04** (0.03)

Table 4: Comparison of the mean Relative Pose Error (RPE) from the DARPA SubT challenge Final Event of the CompSLAM approach during the run and post-processed across all the deployed robots.

Prize Round Evaluation - mean APE and standard deviation						
Robot	**During Run**		**Post-Processed**			
	CompSLAM		**CompSLAM**		**M3RM**	
	Rotation [°]	Translation [m]	Rotation [°]	Translation [m]	Rotation [°]	Translation [m]
ANYmal 1	3.71 (2.02)	0.60 (0.37)	2.45 (0.67)	0.72 (0.41)	**1.59** (0.46)	**0.25** (0.13)
ANYmal 2	4.55 (1.07)	1.81 (0.84)	3.97 (0.40)	1.29 (0.90)	**0.96** (0.30)	**0.36** (0.28)
ANYmal 3	1.53 (0.68)	0.22 (0.29)	**0.89** (0.50)	0.23 (0.43)	2.30 (1.02)	**0.20** (0.34)
ANYmal 4	3.26 (1.02)	1.24 (0.67)	2.22 (0.79)	1.00 (0.71)	**2.16** (0.55)	**0.24** (0.17)

Table 5: Comparison of the mean Absolute Pose Error (APE) from the DARPA SubT challenge Final Event between CompSLAM (during the run and post-processed) and the global M3RM (post-processed only) localization and mapping approaches.

This allowed us to recover from mapping failure, as shown in Figure 29b. Furthermore, the use of learned motion costs showed to be valuable in producing safer, more efficient paths. Figure 29c shows how the navigation planner prefers to circumnavigate high railroad tracks instead of crossing them. In cases where crossing rails is necessary the robot rotates before stepping onto the rails such that it can walk straight on them, with the rail between its legs.

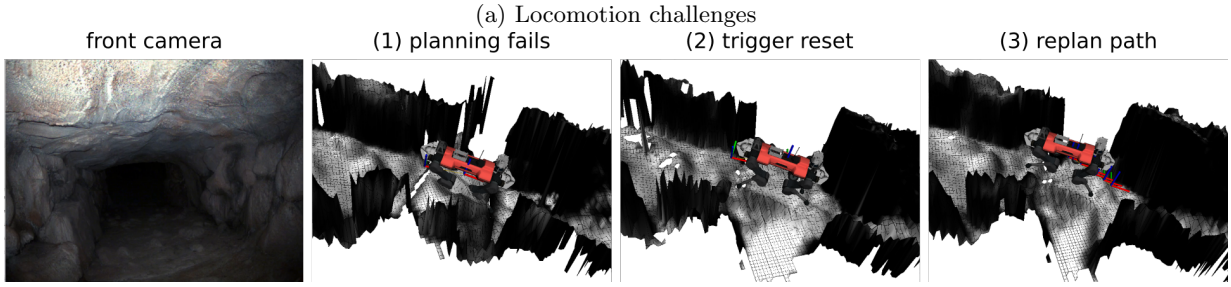
Operator's interface challenges

The usage the GUIs, as presented in section 3.6, significantly improved the capabilities of the Human Supervisor to manage the overall mission and interact with the single agents, compared to the Team's setup at the previous SubT events. Despite the improvements achieved by having only two user interfaces, the context switching of selecting a different robot in the MCI was the principal Human Supervisor's limitation. Whenever he chose a new robot in this interface, for example to task the agent with a different exploration goal, he often had first to query the robot for additional information. Specifically, every time a new robot was selected, the supervisor first had to request the robot's information, such as its local CompSLAM map or specific sensors data, wait for the agent to stream them and then decide on a new exploration goal for the agent. These data were robot specific and had to be requested for every selected robot. In the case of bandwidth-demanding data streamed over a wireless connection shared with several agents, the whole process could take up to several seconds to complete. Therefore, selecting a new robot to control included a fixed context switching, to be paid in terms of no-operation time for the supervisor, limiting how quickly he could supervise different agents. This constraint, combined with the request to check for artifacts reports and overall mission status in the MARI, overloaded the Human Supervisor. A single GUI to both control a selected robot as well as display an overview of the mission alongside the artifacts reports would have been beneficial and released some of the cognitive load from the Human Supervisor.

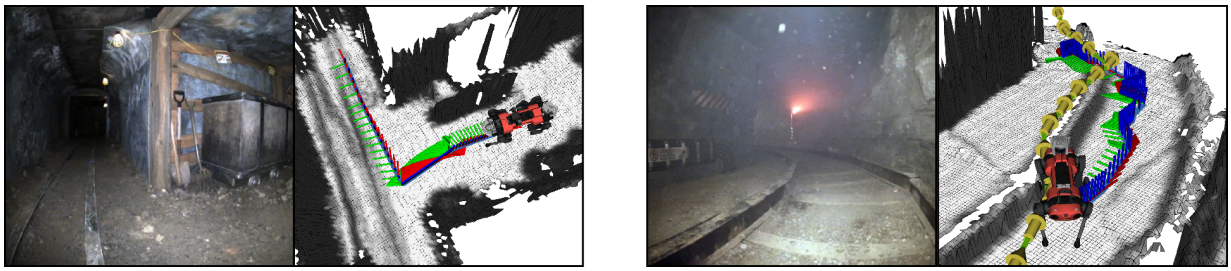
Communication challenges

The combination of a tethered rover and deployable wireless breadcrumb modules helped the team's overall

(a) Locomotion challenges



(b) Automatic reset of the elevation map



(c) ANYmal navigation planner instances

Figure 29: Highlights of ANYmal's mobility in fully autonomous mode during the Prize Round. (a) Locomotion challenges included obstacles, rough surfaces, and railway tracks (left); slopes and constrained passages (center); rough terrain (right). (b) Automatic safety mechanisms acted during the mission to ensure the robot could deal with unexpected situations, for example, with challenging localization conditions leading to corrupted height maps, like in the case shown in part (1), where drift in the robot's odometry made the elevation map move upwards up, touching the agent's knees, thus preventing the ANYmal navigation planner from finding a traversable path due to an invalid start position. If such a condition was detected, the terrain map was automatically reset and a new planning iteration triggered. (c) The learned motion cost within the ANYmal navigation planner allowed us to circumnavigate obstacles like high rails, instead of crossing them. Poses represented as RGB axes were produced by the navigation planner, whereas poses displayed as yellow arrows were created by the exploration planner. This also led to safer behavior when the robot had to walk on rails, turning before stepping onto the rails to then walk straight.

performance during the mission execution by extending the wireless network's reach. Despite the development done to create ruggedized and reliable breadcrumb nodes, we deployed only three of them at the Final Round (given the eight available modules, four per Carrier robot). We identified two main reasons for using just a few modules: the increased level and reliability of the autonomy of the agents and the cognitive overload of the Human Supervisor. The former implied that the supervisor could reliably task each agent to autonomously explore an assigned area, given a time budget, and wait for the robot to come back into communication range to then receive the artifact reports. While the robot was out of communication range, the supervisor could focus on other important matters, such as coordinating the other robots or analyzing other artifacts reports. These important matters are also the cause of the cognitive load on the Human Supervisor, that couldn't really focus on if and where to deploy the communication breadcrumbs of each Carrier robot. Deploying more breadcrumb modules could have helped the Human Supervisor stay connected with the robots even when further away from the Base Station, thus allowing to send artifacts reports sooner. This could have been possible only if an automatic mechanism to decide when to deploy a breadcrumb node had been implemented for the Carrier robot.

Planning challenges

The Final Event of the DARPA SubT Challenge included dynamic obstacles blocking certain parts of the course. The team faced challenges due to these dynamic obstacles. The global re-positioning and autonomous homing (through GBPlanner2 or backtracking of the traversed path) parts of the autonomy stack were not designed for handling dynamic obstacles. In the Prize Round two of the robots encountered the dynamic obstacles, one in the Tunnel section and the other in the Urban section. While ANYmal 1 was performing global repositioning, the path was blocked by a dynamic obstacle. It attempted to follow the path for some time after which the watchdog module of ANYmal's behavior tree triggered backtracking of the previously traversed path. However due to the obstacle ANYmal 1 was stuck for the remaining of the round. This was due to two factors happening at the same time: the backtracking of the path command can be stopped only by the Human Supervisor but in that section of the course there was no WiFi connection with the Base Station, therefore the supervisor did not realize the robot was still functioning but stuck in an obstacle. Similarly, ANYmal 4 encountered a dynamic obstacle while backtracking the traversed path to the home location. ANYmal 4 was stuck in a similar fashion as ANYmal 1, but still in connection range. Therefore, the Human Supervisor could stop the recovery behavior and trigger GBPlanner2 again to explore another area. Figure 30 shows both the instances where the robots encountered the dynamic obstacles.

In the Urban section of the final course ANYmal 4 encountered two construction cones creating a narrow gap through which the robot could not fit through. However, due to the relaxed tuning of GBPlanner2, it planned through that gap on multiple occasions. The navigation planner correctly prevented the robot from passing through that gap. This demonstrates the usefulness of the synergy between an optimistic exploration planner and a conservative robot navigation planner. However, in our autonomy system, the two planners were loosely coupled. It meant that if the navigation planner detected that the path from the exploration planner could not be tracked, the exploration planner would be re-triggered. There was a lack of feedback from the navigation planner to the exploration planner which caused the exploration planner to plan through that gap multiple times slowing down the mission. Figure 31 shows one instance of this.

In the tight underground environments encountered during the Final Event, the 2.5D height map representation used for planning became problematic. We used a filter which allows measurements at larger height to be included into the height map, as distance increases. It was tuned to work on slopes and stairs, but still allowed us to pass through low openings when the robot was close to them. This caused artifacts in the elevation map in areas with low ceiling height, as shown in Figure 32(top). While these artifacts were removed by a visibility cleanup check using ray-casting as the robot got closer, they still restricted space considered traversable by the planner and therefore slowed progress.

During data analysis after the competition, we discovered that there was a non-constant delay between the navigation planner publishing a path, and the path tracking module which we observed to be up to 500 ms. This, in combination with height map issues and the nonsmooth nature of sampling-based planning, caused ANYmal 2 to get stuck on a scaffolding pole in the Urban section for about 20 s. The time series of this event

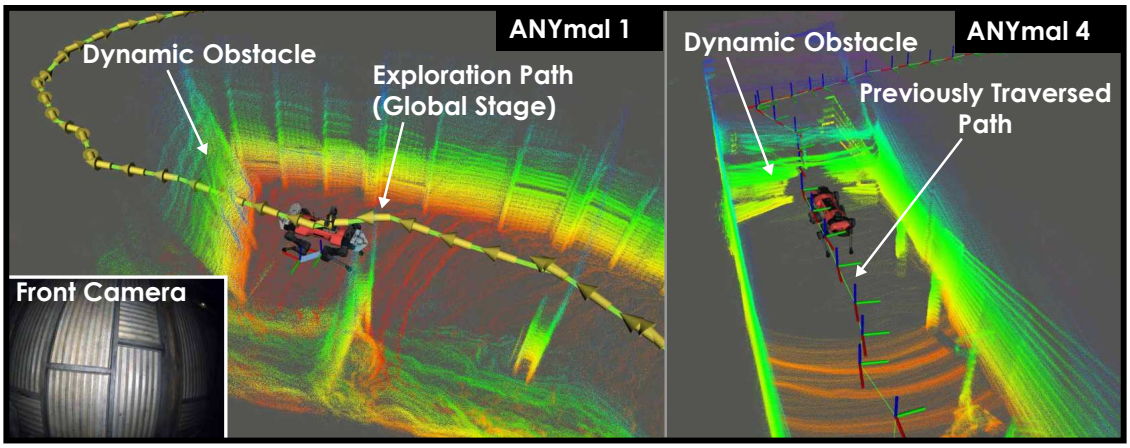


Figure 30: ANYmal 1 and ANYmal 4 encountering dynamic obstacles in the Tunnel and Urban section respectively. The dynamic obstacle blocked the path of ANYmal 1 while performing global repositioning. It attempted to follow the path for some time but could not move due to the presence of the obstacle. Therefore after some time, the watchdog module of ANYmal's behavior tree triggered and it started backtracking the previously traversed path. However, due to the obstacle blocking the path ANYmal 1 initially took, the robot was stuck for the remaining of the round. This was due to two factors happening at the same time: the backtracking of the path command can be stopped only by the Human Supervisor, but in that section of the course, there was no WiFi connection with the Base Station therefore the supervisor did not realize the robot was still functioning but stuck in an obstacle. ANYmal 4 was backtracking its traversed path to the home location when it encountered the dynamic obstacle. The robot was stuck for some duration after which the local layer of GBPlanner2 was manually triggered by the Human Supervisor.

is shown in Figure 32(bottom): When the robot approached the pole, the initial plan was to circumnavigate it on the right, which was slightly longer than on the left. Just as the robot started moving, a new path was published, which passed the pole on the left. This caused the robot to move so close to the pole that it could not be perceived and disappeared from the height map. Since the robot had followed the original path to the right, the next path went right again and got the robot stuck on the pole. Because the pole was now missing from the height map, all new paths went straight through it. After another 20 s, the robot had drifted to the left of the pole and finally got unstuck.

Perception challenges

The course designers of the DARPA's organizational team introduced several challenging conditions to directly and indirectly influence the performance of robot perception. Figure 33 summarizes the key challenges faced during the exploration of the course environment. Throughout the course, several areas were completely dark and dull. Hence, the visual cameras had to rely on the onboard illumination. The effective range of the onboard LEDs was approximately 7 m which was enough for close passages such as tunnel and urban systems. However, in large open-spaces such as the cave environment it only illuminated the nearby structure and effectively missed regions further away. Such areas posed a challenge for visual odometry estimation as features were only detected tracked in the near vicinity of the camera, typically on the planar ground directly in front of the robot.

Furthermore, the course featured many narrow passages, some with smooth planar surfaces, e.g. the corridor of the urban area. Such areas posed a challenge for the LiDAR odometry, mainly due to two reasons. First, the narrow corridors significantly reduced the observation range of the already sparse VLP16 Puck LITE, making the estimation of ground and ceiling planar surfaces at longer ranger difficult. Second, the smooth planar walls created a featureless self-similar environment along the principal direction of the corridor. Accordingly, this rendered the underlying LiDAR pose estimation optimization briefly ill-conditioned, thus enforcing the use of fallback solutions for the correct robot pose and map estimation. Another crucial type for perceptual degradation was the introduction of obscurants such as smoke and dust, as well as dynamic

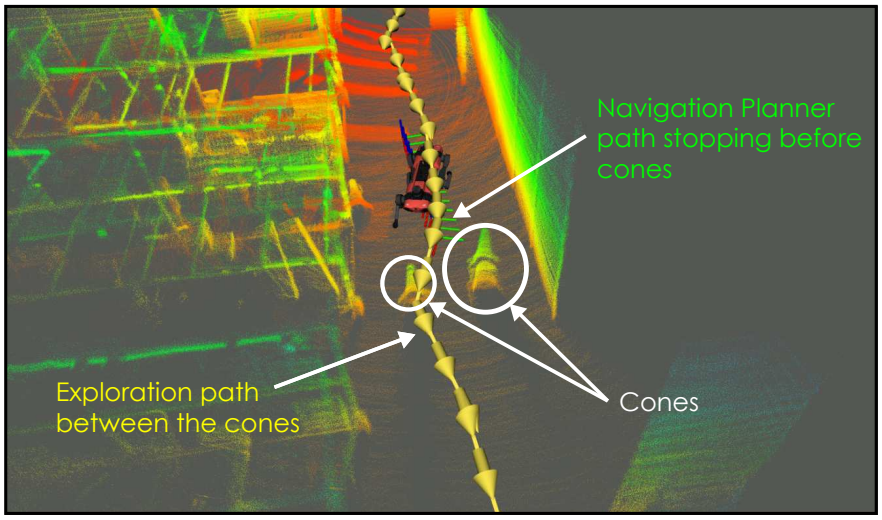


Figure 31: An instance where ANYmal 4 encountered a non-traversable narrow gap formed by two construction cones. The exploration planner attempted to plan through the gap, but the navigation planner prevented the robot from going through it as it was not safe for the robot to traverse through it. This shows the usefulness of the synergy between the two planners, but also highlights the lack of feedback from the navigation planner to the exploration planner.

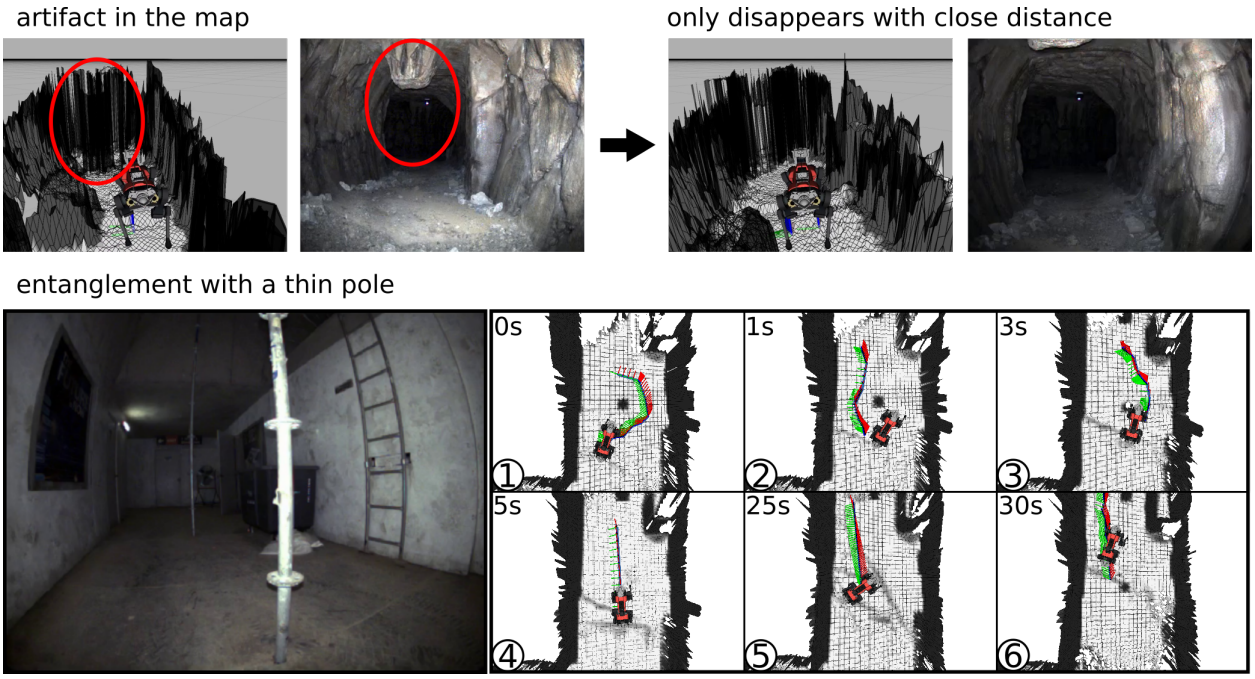


Figure 32: ANYmal navigation challenges. (top) To be able to capture slopes or stairs, the upper limit of depth measurements included into the elevation map was set above robot height. This produced artifacts in the elevation map when low ceiling points are projected into the 2.5D grid. These artifacts were only removed when the robot got closer, thanks to visibility cleanup and rejection based on distance, which slowed down planning. (bottom) A 500 ms time delay between computing a path and starting to follow it lead to imperfect path following. As a result, one robot briefly got stuck on a narrow scaffolding pole in the urban section.

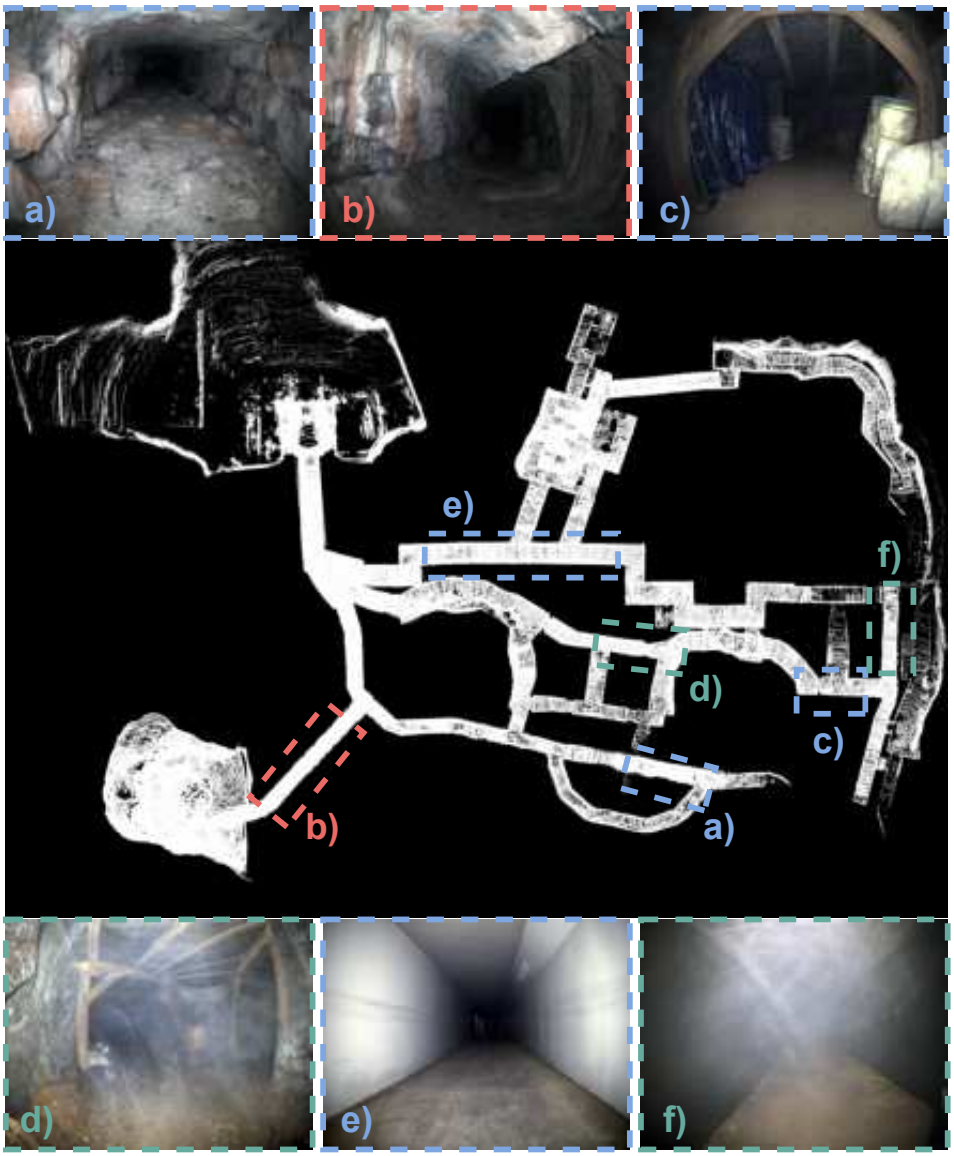


Figure 33: Illustration of a selection of different types of degradation encountered during the DARPA SubT Final Event. Blue denotes completely different kinds of environment; red denotes narrow passages; green areas denote smoke. Displayed images are taken from the onboard cameras of the deployed robots.

elements, e.g. trap-doors, into the course. Different areas in the course featured motion-triggered smoke machines that produced artificial smoke visible in the visual camera and LiDAR sensors. The artificial smoke introduced many incorrect feature detections for visual tracking and led to the loss of good visual landmarks in the scene. Additionally, the artificial smoke might lead to incorrect measurements for the LiDAR sensors for which the configuration of only using the strongest return signal per beam is a possible remedy. Especially during the Prize Round, where our robots were exposed to multiple sections filled with smoke, our onboard CompSLAM successfully penetrated such conditions and offered robust estimates.

Moreover, the trap doors made it particularly difficult for the registration-based approaches, e.g., scan-to-map or the LiDAR loop closures to spatially close poses, since the current pointcloud differs significantly from the previously acquired pointcloud.

As mentioned before, the M3RM performance significantly decreased due to the degraded entrance to the urban environment. Generally, the entrance to the urban area featured a long white and texture-less corridor with no salient regions nor structure. Thus, putting a significant burden on the visual perception pipelines for the M3RM approach as the lack of good visual features and wrong associations lead to incorrect visual landmarks in the factor graph that resulted in erroneous estimation.

Additionally, the narrow passages not only during the entrance of the urban area but also throughout the course yielded many overexposed scenes since the onboard lighting system was not adaptively controlled. As a result for M3RM, in many situations, the employed detection and description of BRISK features resulted in only a handful of good landmarks since most of them were directly discarded due to having a too short feature track. Similar to the urban entrance, the factor graph then becomes less constrained and more sensitive to the other terms, such as the IMU and especially its biases.

It is evident that a keypoint detection and description using BRISK has limited performance and efficiency in subterranean environments. Therefore, greatly decreasing the M3RM pipeline as the visual landmarks were primarily used for global loop closure. An experimental processing of the Prize Round using different feature types such as Superpoint [DeTone et al., 2018] showed that the multi-robot map significantly improves. Figure 34 compares two visual maps, one built with BRISK and one with Superpoint features.

5 Lessons learned

During the three years of the SubT Challenge, our Team gathered a considerable amount of information and insights concerning how our research and development efforts toward efficient subterranean exploration should progress. The main point is that subterranean environments present extreme diversities and pose multiple challenges for robotic autonomy. Our approach toward the exploration of these environments involved, since the beginning of the completion, one unifying, versatile, and scalable solution across all types of underground environments. This section provides overall conclusions for lessons learned that we hope can inform other teams, research groups, technology developers, system operators, and stakeholders that share similar goals and interests.

5.1 Lessons learned regarding the robotic technologies

Legged and flying robots: Dexterous and versatile robotic mobility is the key enabling factor to exploring and searching such environments. In our experience, this can be achieved in a robotic systems-of-systems fashion involving multiple legged and flying robots that complement their skills to conquer any terrain and navigate any environment. Quadruped systems offer a versatile and unifying way to negotiate diverse terrain challenges, including rough rocky terrain, large drops/climbs, inclines, steps, stairs, mud, sand, and/or water. They can outperform similar sized tracked or wheeled vehicles in terms of mobility and represent, in our experience, the main robot category to be used when long-term and large-scale (subterranean) exploration

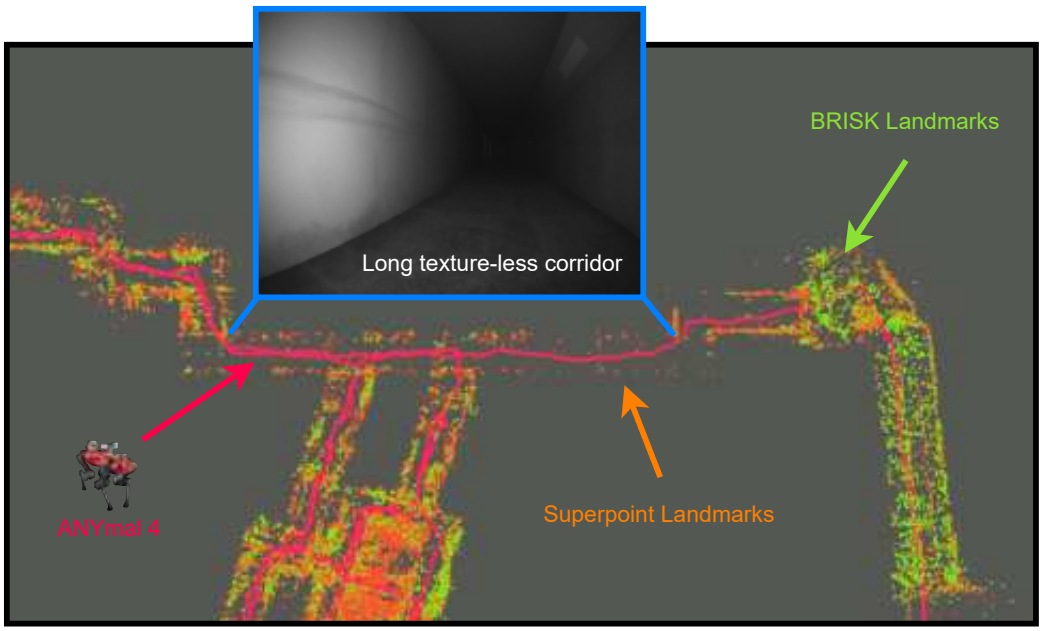


Figure 34: Visual map of the Prize Round of ANYmal 4. The map only contains landmarks with a minimum of three observations. Thus, landmarks detected using BRISK are barely present in the corridor. Despite the lack of structure, Superpoint still detects several good landmarks in this challenging environment.

is required. Simultaneously, flying systems proved, through our years of experience in the challenge, their significance as the only platform that can naturally be unaffected by the complexities of terrain and navigate fully vertical passages. However, aerial systems typically suffer from endurance and payload limitations, and thus should be used alongside ground robots. We believe that continuing research in legged and flying systems-of-systems, with enhanced walking dexterity, improved flying agility, and the possibility of marsupial integration of ground and flying systems, is the key to conquering the diversity of underground environments.

Resilient multi-modal perception is the core enabling factor: Despite historic progress in the field of SLAM, the subterranean environments presented a number of challenging conditions that significantly degraded robot perception. Consequently despite an overall resilient performance - primarily based on our onboard localization and mapping - certain corner cases and points of failure were identified. Systems had to be made resilient in their ability to enable operation even through the most degraded perceptual conditions. Multi-modal robotic perception fusing complementary sensor data capable of penetrating diverse conditions of perceptual-degradation, and presenting different cases of ill-conditioning (e.g., LiDAR data are not affected by lack of visual texture which challenges visible-light cameras but it can be challenged in conditions of geometric similarity which is irrelevant to visual systems), was essential in order to minimize possible breaking points in the robots' ability to localize and map. However, our developed solutions faced a set of challenges. CompSLAM, running onboard our robots, employed a hierarchical sensor fusion architecture which entails that we had to define a set of decision functions and heuristics. Despite its satisfactory performance for the purposes of the SubT Challenge, this approach may present limitations in terms of generalizability. Furthermore, our multi-robot pipeline (M3RM) was severely challenged both in terms of computational requirements and with respect to the resilience of its solution against conditions of visual degradation.

Resilient autonomy technologies are key to explore and search subterranean environments: Subterranean exploration requires high-degrees of autonomy and the means to control a large team of robots through intuitive high-level commands. The team believed that every robot should be capable of resilient individual autonomy without human teleoperation or without the need to communicate with other robots or the Base Station. We have demonstrated that an exploration planning policy reflecting the key geometric challenges of underground environments could uniformly guide vastly different robotic configurations to

explore and search efficiently. The "Supervised Autonomy" paradigm (as described in section 3.6) proved to be effective for such an application. However, some key aspects require significant further improvements. First, for our team only single-robot autonomy was deployed despite multi-robot coordination methods being developed [Kulkarni et al., 2022]. We believe that significantly more field testing experience and accordingly algorithmic refinements are necessary in this direction. Simultaneously, the type of autonomy we deployed did not explicitly exploit semantics (e.g., explicit detection of- and navigation through-staircases or doors) in the environment and did not present a comprehensive modeling of navigation risks (e.g., the interplay between perception conditions and path planning decisions), while a further and more tight integration of traversability constraints at the exploration planning level would have possibly proved beneficial.

Testing multi-robot mapping for resilience: Deployment tests of multi-robot mapping approaches, such as M3RM, is difficult as it depends heavily on the downstream tasks such as locomotion, communication, and the general state of the robot. Thus, isolated multi-robot testing for scalability and degeneracy issues would yield more information on its weaknesses without having the full system-of-systems running. Moreover, tuning mapping systems gets harder the more robots and environments are involved. A better generic applicability requires less tuning-dependent solutions with fewer parameters and/or self-tuning or adapting thresholds, eventually leading to an environment agnostic real world mapping approach. It is insufficient to handle failures locally in individual components of the whole pipeline. Failures need to be considered and treated holistically when, e.g., computing the map or trajectories. This obviously increases the amount of investigation but greatly increases the overall efficiency of the team.

Advances on legged robotics: In the course of the SubT challenge, we could profit from rapid progress in the field of legged robots. Our team relied on the ANYbotics "ANYmal B300" platform during the first two years as its main ground robot. Despite this hardware solution featuring a good level of robustness and reliability, we experienced a general lack of performance consistency across multiple missions, limited payload capabilities, and insufficient mobility to handle the diverse and challenging terrain of the SubT Challenge. Switching to the ANYmal C quadruped series provided our team with the advantages of a new, stronger, larger, and more dexterous robot compared to the previous B-type platform. The ANYmal C-series was a mature, robust, and reliable hardware solution that allowed us to focus on developing specialized software modules, such as the learning-based perceptive locomotion controller and the navigation planner, that fully leveraged the capabilities of the new robot type and boosted its mobility and robustness. For example, in two years we could increase our locomotion speed from $0.3\,\mathrm{m\,s^{-1}}$ (Tunnel Circuit) to $0.45\,\mathrm{m\,s^{-1}}$ (Urban Circuit), up to $0.7\,\mathrm{m\,s^{-1}}$ (Final Event, where the speed was kept at this level on purpose even though the robot could locomote faster than $1\,\mathrm{m\,s^{-1}}$). In the Final Event, the four ANYmal platforms could operate without a single fall, indicating this solution's maturity. Proof of the advancement in legged robotics is also visible from the overall teams' choices during the three years of the SubT Challenge. In fact, during the first year of the competition (2019), only two teams out of eleven chose legged robots as their main ground vehicles, and both did not perform very well. Only two years later, during the Final Event, the top-6 teams utilized quadruped systems ("ANYmal C" for our case and "Boston Dynamics Spot" for the other teams). We believe that this is a tipping point in the field of UGVs, with legged robots becoming a superior and broadly applicable choice of platform. However, despite these advances, there is still plenty of room to increase the capabilities of the systems further. For example, the ability to climb over rough terrain is still in its infancy compared to the natural counterparts and can certainly be improved. Second, the separation of navigation, motion planning/following, and locomotion control drastically limits the possible maneuvers. Third, the robots still rely on abstract geometric terrain representations for locomotion while lacking a semantic understanding for locomotion. And lastly, they have limited to no active interaction capabilities to manipulate the environment, e.g., for terrain probing or to remove obstacles.

Marsupial aerial-ground system-of-systems are important: Our team developed a host of flying robots that conducted successful exploration missions during the first two circuit events, including the vertical traversal of staircases. However, as the SubT Challenge entailed that all robots start from the same starting point it essentially implied that flying robots would be much more useful if deployed deeper inside the underground environment. Indeed, the finals course contained a massive "cave"-like chamber where if we had the marsupial integration of one of our small-sized flying robots onboard ANYmal we could have

gained the benefits from the versatile navigation capabilities of aerial systems without their limited endurance being an important constraint. Unfortunately, due to limited development time, our team could complete the marsupial integration of ANYmal C with RMF-Owl [De Petris et al., 2022] only after the competition was over.

Communications: During the first two years of the competition, the reliability and performance of the employed communication solutions corresponded to a critical challenge in our experience. Therefore, before the Final Event, we developed and deployed an improved and more reliable hardware solution for subterranean communication, based on an off-the-shelf baseline radio module, alongside a new iteration of the deployable WiFi communication-extender modules. However, achieving an always-reliable communication link is particularly challenging in underground environments. To that end, in our experience a high level of autonomy in the robot agents allowed to mitigate this problem by enabling exploration when not in communication range.

Use of simulation and synthetic data: The artifact detection module was trained using hand-labeled data collected in underground settings. However, this approach was slow and cumbersome, resulting in the collection of a limited amount of data. Photo-realistic simulators and rendering engines could be exploited to generate data captured using various simulated camera parameters, light conditions, obscurants, and degradation, making the artifact detection module more robust. Apart from this, simulation can help in stress testing the methods, debugging minor errors, and tuning parameters of the algorithms which is not feasible during a field test. On this front, we utilized the RotorS Gazebo Simulator [Furrer et al., 2016] for the simulation of aerial robots and a Gazebo-based simulator for the ANYmal C legged robots. Specifically, the path planning algorithms were tested in various simulation environments, both artificial and real models obtained or created from field data. However, the use of simulation was limited to testing individual software components for a single robot. High-fidelity simulators could be exploited to simulate more complex elements such as photo-realistic environments, obscurants, variable lighting conditions, dynamic obstacles, and behavior of communication networks for large-scale realistic simulations. Additionally, multi-robot simulation can be exploited for complete mock missions.

5.2 On the importance of the frequency and the character of field testing

We realized that many characteristics of the subterranean settings as well as the challenges presented by them cannot be fully understood without going into these environments. The experience from field deployments is crucial in making each component in the system robust.

Making the robots "field-ready"
Team CERBERUS conducted a large number of field deployments in diverse environments, across different countries, replicating the settings of the competition as much as possible including timed missions (detailed description of the nature of field deployments is provided in Section 4.1.1). We conducted at least one field test every month in the year before the Final Event. The selected deployment sites spanned across all three categories of environments in the competition - tunnel, urban, and cave, each presenting different challenges as described in section 4.1.2. We have demonstrated that through testing in diverse environments, the systems were able to tackle the majority of the challenges presented across all three types of environments. Furthermore, multiple tests in the same environment helped in benchmarking the systems and keeping track of the progress.

These field tests not only helped in making the systems field ready in terms of the technology, but also served as practice missions for the Human Supervisor and the Pit Crew. Treating the deployments in the field tests as if they were in the actual competition helped in fine tuning the deployment procedures and related technologies such as the user interface, software stack launching scripts, and safety checks before deployment. Thoroughly planning, preparing, and testing the hardware and software components in the days before the field deployment and during the weekly shakeouts was a crucial practice that significantly improved the outcome of each field test, both in terms of the number of tested features and overall system

integration in a mock-up mission scenario. The day before each field test, a functionality test of all the robots and features that were to be deployed was conducted. If issues were found in any feature, it was not tested in the field experiment. This helped in identifying issues in small yet critical parts of the systems with can easily get ignored while developing the core technology. Furthermore, this approach ensured that each implemented feature was thoroughly tested during the development process before deploying on the robot. Periodic field deployments forced us to have internal, hard deadlines that pushed the development of "field-ready" robotic systems.

As our team was distributed across the globe, several components of the system, including entire robots, were developed at different places. However, for the whole system-of-systems to work together good integration among all the hardware and software components across the robots is critical. Hence, we conducted several integration exercises involving members of all groups participating in the team. These exercises included integration of software and hardware components across all robots, functionality testing, and field deployment of heterogeneous robot teams. In our experience, the integration exercises were important for a) ensuring that all robots had the necessary hardware and software components correctly configured for working with each other and the Base Station, b) making the Human Supervisor familiar with the setup and user interface for all types robots, and c) conducting mock, timed missions using the entire fleet.

Importance of field testing for resilient perception and autonomy
The response of real sensors and systems in the real environments is hard to replicate synthetically, especially for different sensing modalities. Therefore such experience and data is crucial for the development and testing of the perception and autonomy algorithms. Many of the features and development directions of the CERBERUS autonomy solutions have evolved from our experiences in the field tests. Some of these features of GBPlanner2 include, but are not limited to, the bifurcated local/global planning architecture for handling large scale topology with complex geometry and terrain, optimistic ground existence check to tackle gaps in the volumetric map, introduction of the hanging vertices for handling negative obstacles. Similarly, we implemented the safety checks and recovery behaviors from unexpected events in the ANYmal behavior tree based on our experience in the field. Moreover, we introduced batched motion cost prediction for the ANYmal navigation planner to improve real-world path safety and prevent very rare planning time outliers, which only happened when executing on the robot. Field tests also motivated the introduction of a foothold safety threshold to avoid planning close to negative obstacles, which are often not adequately mapped during actual deployment. Additionally, field experiments provided a better understanding of the robot's response to the navigation commands and what can be considered as a safe action.

Not all sensing modalities can be captured in simulation accurately (e.g., thermal camera frames), and their behavior in subterranean environments can be understood only from field data. Furthermore, several hardware challenges affect the performance of perception solutions. These include handling sensor delays, time synchronization between sensors, and issues caused by other components of the robot (e.g., vibrations, occlusions, etc.). Through field evaluation, several cases of sensor degradation were identified (e.g., self-similarity, smoke, fog, darkness) which guided the development of the multi-modal localization and mapping framework. Hence, the data collected in the field is the only way to evaluate, benchmark, and tune the localization and mapping algorithms to tackle all the challenges present in the subterranean environments. Furthermore, the field experiments also showed the computing limitations of each robot so that the algorithms can be modified or tuned accordingly.

6 Conclusions

This paper presents a comprehensive report of the technological progress made by Team CERBERUS participating in the DARPA SubT Challenge and specifically towards the Final Event. We detailed the CERBERUS' vision and motivation for the Team's overall "system-of-systems" methodology, followed by an extensive presentation of the employed robotic systems. The contributed research allowed a high level of autonomy and robustness in the CERBERUS robots, operating inside visually-degraded, large-scale, nar-

row, and broadly complex underground settings. At the core of the CERBERUS robotic system is a team of legged and flying robots, including collision-tolerant designs, further enhanced with a roving platform. The implemented autonomy components, including multi-modal perception and path-planning, navigation and control of legged robots, and the automated artifact detection and scoring systems, are detailed alongside the operator interfaces for exerting high-level control. Finally, we presented the Team's performance in the winning Prize Round of the challenge, followed by the critical lessons learned from these experiences. We hope that the presented work and released open-source code can contribute towards the overall community goal of enabling resilient robotic autonomy in extreme underground environments.

Acknowledgments

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

The authors would also like to thank ANYmal Bear, ANYmal Badger, ANYmal Cerberus, ANYmal Chimera, ANYmal Camel, ANYmal Caiman, ANYmal Coyote, SMB Armadillo, Aerial Scouts Alpha, Bravo, Charlie, Gagarins, RMF-Owl, Kolibri, and Kraken that were not harmed during the DARPA SubT Challenge events, even though the same might not be said for the prior field deployments. We want to thank all members of Team CERBERUS who contributed to the team's success. We extend our gratitude to all the SubT Community and the DARPA team for the exciting challenge and collaborative community that was built.

A List of open source packages and data

With the goal to support the overall community around subterranean robotics, our team focused on open-sourcing significant components of our technological solution. Similarly, data from field deployments are released.

A.1 Open source code

A selected - but not exhaustive - list of relevant open-source packages of our team are outlined below.

GBPlanner2 - a graph-based exploration planner for subterranean environments: `https://github.com/ntnu-arl/gbplanner_ros`

Art Planner - a local navigation planner for legged robots: `https://github.com/leggedrobotics/art_planner`

Elevation Mapping Cupy - a GPU-based robot-centric elevation mapping framework for rough terrain navigation: `https://github.com/leggedrobotics/elevation_mapping_cupy`

Maplab - an open visual-inertial mapping framework, which includes the M3RM centralized multi-modal, multi-robot mapping extension: `https://github.com/ethz-asl/maplab`

Darknet ROS - Real-Time Object Detection for ROS: `https://github.com/leggedrobotics/darknet_ros`

Kalibr - a visual-inertial calibration toolbox: `https://github.com/ethz-asl/kalibr`

ROVIO - a Robust Visual Inertial Odometry framework: `https://github.com/ethz-asl/rovio`

Voxblox - a library for flexible voxel-based mapping: `https://github.com/ethz-asl/voxblox`

A.2 Simulation models

The SubT Challenge Systems and Virtual Competitions are interconnected with the Systems' teams delivering models of their robots for the Virtual Competition, while progress achieved in the Virtual Competition can benefit the autonomy functionalities in the Systems Competition. Models of most of the robots fielded by our team in the Final Event have already been released. The following robots related to the Final Event are available.

ANYmal C SubT Explorer: `https://github.com/osrf/subt/tree/master/submitted_models/cerberus_anymal_c_sensor_config_1`

RMF-Owl aerial scout: `https://github.com/osrf/subt/tree/master/submitted_models/cerberus_rmf_sensor_config_1`

DJI Matrice M100-based aerial scout: `https://github.com/osrf/subt/tree/master/submitted_models/cerberus_m100_sensor_config_1`

A.3 Open datasets

We have released an open dataset from the Prize Round of the Final Event of the DARPA SubT Challenge. This dataset is accessible at `https://www.subt-cerberus.org/code--data.html`. Through the same link, possible further releases of datasets or new software will become accessible.

References

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., Lee, H., Kanellakis, C., Mayo, J., Harper, S., Kaufmann, M., Dixit, A., Correa, G., Lee, C., Gao, J., Merewether, G., Maldonado-Contreras, J., Salhotra, G., Da Silva, M. S., Ramtoula, B., Kubo, Y., 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., Choi, C., Daftry, S., Toupet, O., Wee, I., Thakur, A., Feras, M., Beltrame, G., Nikolakopoulos, G., Shim, D., Carlone, L., and Burdick, J. (2021). NeBula: Quest for Robotic Autonomy in Challenging Environments; TEAM CoSTAR at the DARPA Subterranean Challenge. *arXiv preprint arXiv:2103.11470*.

Agrawal, V. (2019). Ground up design of a multi-modal object detection system. Master's thesis, The Robotics Institute, Carnegie Mellon University.

Atkeson, C., Babu, B., Banergee, N., Berenson, D., Bove, C., Cui, X., and DeDonato, M. (2016). What happened at the darpa robotics challenge and why? *https://www.cs.cmu.edu/~cga/drc/jfr-what.pdf*.

Baca, T., Petrlik, M., Vrba, M., Spurny, V., Penicka, R., Hert, D., and Saska, M. (2021). The mrs uav system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, 102:1573 – 0409. Springer Science+Business Media.

Bayer, J. and Faigl, J. (2019). On autonomous spatial exploration with small hexapod walking robot using tracking camera intel realsense t265. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE.

Bayer, J. and Faigl, J. (2020). Speeded up elevation map for exploration of large-scale subterranean environments. In *Modelling and Simulation for Autonomous Systems*, pages 190–202. Springer International Publishing.

Bernreiter, L., Khattak, S., Ott, L., Siegwart, R., Hutter, M., and Cadena, C. (2022). Collaborative robot mapping using spectral graph analysis. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

Bloesch, M., Burri, M., Omari, S., Hutter, M., and Siegwart, R. (2017a). Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 36(10):1053–1072. SAGE Publishing.

Bloesch, M., Burri, M., Sommer, H., Siegwart, R., and Hutter, M. (2017b). The two-state implicit filter recursive estimation for mobile robots. *IEEE Robotics and Automation Letters*, 3(1):573–580. IEEE.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Carrillo, H., Reid, I., and Castellanos, J. A. (2012). On the comparison of uncertainty criteria for active slam. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2080–2087. IEEE.

Chao, C., Hongbiao, Z., Howie, C., and Ji, Z. (2021). TARE: A Hierarchical Framework for Efficiently Exploring Complex 3D Environments. In *Robotics: Science and Systems XVII*, Virtual. Robotics: Science and Systems Foundation.

Colas, F., Mahesh, S., Pomerleau, F., Liu, M., and Siegwart, R. (2013). 3d path planning and execution for search and rescue ground robots. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 722–727. IEEE.

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.

De Petris, P., Khattak, S., Dharmadhikari, M., Waibel, G., Nguyen, H., Montenegro, M., Khedekar, N., Alexis, K., and Hutter, M. (2022). Marsupial walking-and-flying robotic deployment for collaborative exploration of unknown environments.

Dellaert, F., Kaess, M., et al. (2017). Factor graphs for robot perception. *Foundations and Trends in Robotics*, 6(1-2):1–139.

DeTone, D., Malisiewicz, T., and Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236.

Ebadi, K., Chang, Y., Palieri, M., Stephens, A., Hatteland, A., Heiden, E., Thakur, A., Funabiki, N., Morrell, B., Wood, S., Carlone, L., and Agha-mohammadi, A.-a. (2020). LAMP: Large-Scale Autonomous Mapping and Positioning for Exploration of Perceptually-Degraded Subterranean Environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 80–86. IEEE.

Fan, D., Agha, A., and Theodorou, E. (2020a). Deep learning tubes for tube mpc. In *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation.

Fan, D., Otsu, K., Kubo, Y., Dixit, A., Burdick, J., and Agha-mohammadi, A.-a. (2021). STEP: Stochastic Traversability Evaluation and Planning for Risk-Aware Off-road Navigation. In *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation.

Fan, D. D., Nguyen, J., Thakker, R., Alatur, N., Agha-mohammadi, A.-a., and Theodorou, E. A. (2020b). Bayesian learning-based adaptive control for safety critical systems. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4093–4099. IEEE.

Furgale, P., Rehder, J., and Siegwart, R. (2013). Unified temporal and spatial calibration for multi-sensor systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1280–1286. IEEE.

Furrer, F., Burri, M., Achtelik, M., and Siegwart, R. (2016). Rotors-a modular gazebo mav simulator framework. In *Robot Operating System (ROS)*, pages 595–625. Springer.

Ginting, M. F., Otsu, K., Edlund, J. A., Gao, J., and Agha-Mohammadi, A.-A. (2021). CHORD: Distributed Data-Sharing via Hybrid ROS 1 and 2 for Multi-Robot Exploration of Large-Scale Complex Environments. *IEEE Robotics and Automation Letters*, 6(3):5064–5071. IEEE.

Graeter, J., Wilczynski, A., and Lauer, M. (2018). Limo: Lidar-monocular visual odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7872–7879. IEEE.

Guzzi, J., Chavez-Garcia, R. O., Nava, M., Gambardella, L. M., and Giusti, A. (2020). Path planning with local motion estimations. *IEEE Robotics and Automation Letters*, 5(2):2586–2593. IEEE.

Hauser, K. (2015). Lazy collision checking in asymptotically-optimal motion planning. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2951–2957. IEEE.

Hines, T., Stepanas, K., Talbot, F., Sa, I., Lewis, J., Hernandez, E., Kottege, N., and Hudson, N. (2021). Virtual surfaces and attitude aware planning and behaviours for negative obstacle navigation. *IEEE Robotics and Automation Letters*. IEEE.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*. Springer.

Huang, J. and Grizzle, J. W. (2020). Improvements to target-based 3d lidar to camera calibration. *IEEE Access*, 8:134101–134110.

Hudson, N., Talbot, F., Cox, M., Williams, J. L., Hines, T., Pitt, A., Wood, B., Frousheger, D., Surdo, K. L., Molnar, T., Steindl, R., Wildie, M., Sa, I., Kottege, N., Stepanas, K., Hernández, 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. C. (2021). Heterogeneous ground and air platforms, homogeneous sensing: Team CSIRO data61's approach to the DARPA subterranean challenge. *arXiv preprint arXiv:2104.09053*.

Johnson, D. L., Ntlatlapa, N., and Aichele, C. (2008). Simple pragmatic approach to mesh routing using batman. In *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries*. Springer.

Khattak, S., Mascarich, F., Dang, T., Papachristos, C., and Alexis, K. (2019). Robust thermal-inertial localization for aerial robots: A case for direct methods. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1061–1068. IEEE.

Khattak, S., Nguyen, H., Mascarich, F., Dang, T., and Alexis, K. (2020). Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1024–1029. IEEE.

Khattak, S., Papachristos, C., and Alexis, K. (2019). Visual-thermal landmarks and inertial fusion for navigation in degraded visual environments. In *2019 IEEE Aerospace Conference*, pages 1–9. IEEE.

Khattak, S., Papachristos, C., and Alexis, K. (2020). Keyframe-based thermal–inertial odometry. *Journal of Field Robotics*, 37(4):552–579. Wiley Online Library.

Kim, S.-K., Bouman, A., Salhotra, G., Fan, D. D., Otsu, K., Burdick, J. W., and akbar Agha-mohammadi, A. (2021). PLGRIM: Hierarchical Value Learning for Large-scale Exploration in Unknown Environments. In *2021 International Conference on Automated Planning and Scheduling (ICAPS)*, volume 31, pages 161–173. AAAI Press.

Kristan, M. and Pernuÿs, F. (2004). Entropy based measure of camera focus.

Krátký, V., Petráček, P., Báča, T., and Saska, M. (2021). An autonomous unmanned aerial vehicle system for fast exploration of large complex indoor environments. *Journal of Field Robotics*, 38(8):1036–1058. Wiley Online Library.

Kulkarni, M., Dharmadhikari, M., Tranzatto, M., Zimmermann, S., Reijgwart, V., De Petris, P., Nguyen, H., Khedekar, N., Papachristos, C., Ott, L., Siegwart, R., Hutter, M., and Alexis, K. (2022). Autonomous teamed exploration of subterranean environments using legged and aerial robots. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Labbé, M. and Michaud, F. (2019). Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446. Wiley Online Library.

Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. Ieee.

Loupos, K., Doulamis, A. D., Stentoumis, C., Protopapadakis, E., Makantasis, K., Doulamis, N., Amditis, A., Chrobocinski, P., Victores, J., Montero, R., Menendez, E., Balaguer, C., Lopez-Tarazon, R., Cantero, M., Navarro, R., Roncaglia, A., Belsito, L., Camarinopoulos, S., Komodakis, N., and Praveer, P. (2018). Autonomous robotic system for tunnel structural inspection and assessment. *International Journal of Intelligent Robotics and Applications*, 2:43–66. Springer Singapore.

Mansouri, S. S., Castaño, M., Kanellakis, C., and Nikolakopoulos, G. (2019). Autonomous mav navigation in underground mines using darkness contours detection. In *2019 International Conference on Computer Vision Systems (ICVS)*, pages 164–174. Springer.

Miki, T., Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., and Hutter, M. (2022a). Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822.

Miki, T., Wellhausen, L., Grandia, R., Jenelten, F., Homberger, T., and Hutter, M. (2022b). Elevation mapping for locomotion and navigation using gpu. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (submitted)*.

Montgomery, D. (2021). D. Montgomery, "The Pentagon's $82 Million Super Bowl of Robots", *The Washington Post Magazine* (2021), `https://www.washingtonpost.com/magazine/2021/11/10/darpa-robot-competition/`.

Murphy, R. R., Kravitz, J., Stover, S. L., and Shoureshi, R. (2009). Mobile robots in mine rescue and recovery. *IEEE Robotics Automation Magazine*, 16(2):91–103.

Novák, P., Babjak, J., Kot, T., Bobovský, Z., Olivka, P., Moczulski, W., Timofiejczuk, A., Adamczyk, M., Guzmán, B. G., Armada, A. G., and Rodríguez, Á. (2017). Telerescuer - reconnaissance mobile robot for underground coal mines. In *2017 Coal Operators' Conference*.

Ohradzansky, M. T., Rush, E. R., Riley, D. G., Mills, A. B., Ahmad, S., McGuire, S., Biggie, H., Harlow, K., Miles, M. J., Frew, E. W., Heckman, C., and Humbert, J. S. (2021). Multi-agent autonomy: Advancements and challenges in subterranean exploration. *arXiv preprint arXiv:2110.04390*.

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

Olson, E. (2011). Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407.

Palieri, M., Morrell, B., Thakur, A., Ebadi, K., Nash, J., Chatterjee, A., Kanellakis, C., Carlone, L., Guaragnella, C., and Agha-mohammadi, A.-a. (2020). Locus: A multi-sensor lidar-centric solution for high-precision odometry and 3d mapping in real-time. *IEEE Robotics and Automation Letters*, 6(2):421–428. IEEE.

Petris, P. D., Nguyen, H., Dharmadhikari, M., Kulkarni, M., Khedekar, N., Mascarich, F., and Alexis, K. (2022). Rmf-owl: A collision-tolerant flying robot for autonomous subterranean exploration. *arXiv preprint arXiv:2202.11055*.

Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. (2013). Comparing icp variants on real-world data sets. *Autonomous Robots*. Springer.

Pomerleau, F., Krüsi, P., Colas, F., Furgale, P., and Siegwart, R. (2014). Long-term 3d map maintenance in dynamic environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3712–3719. IEEE.

Qin, H., Meng, Z., Meng, W., Chen, X., Sun, H., Lin, F., and Ang, M. H. (2019). Autonomous exploration and mapping system using heterogeneous uavs and ugvs in gps-denied environments. *IEEE Transactions on Vehicular Technology*, 68(2):1339–1350.

Ramezani, M., Khosoussi, K., Catt, G., Moghadam, P., Williams, J., Borges, P., Pauling, F., and Kottege, N. (2022). Wildcat: Online Continuous-Time 3D Lidar-Inertial SLAM. *arXiv preprint arXiv:2205.12595*.

Ramezani, M., Wang, Y., Camurri, M., Wisth, D., Mattamala, M., and Fallon, M. (2020). The newer college dataset: Handheld lidar, inertial and vision with ground truth. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4353–4360. IEEE.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Rouček, T., Pecka, M., Čížek, P., Petříček, T., Bayer, J., Šalanskỳ, V., Azayev, T., Heřt, D., Petrlík, M., Báča, T., et al. (2021). System for multi-robotic exploration of underground environments ctu-cras-norlab in the darpa subterranean challenge. *arXiv preprint arXiv:2110.05911*.

Rouček, T., Pecka, M., Čížek, P., Petříček, T., Bayer, J., Šalanský, V., Heřt, D., Petrlík, M., Báča, T., Spurný, V., Pomerleau, F., Kubelka, V., Faigl, J., Zimmermann, K., Saska, M., Svoboda, T., and Krajník, T. (2020). Darpa subterranean challenge: Multi-robotic exploration of underground environments. In *Modelling and Simulation for Autonomous Systems*, pages 274–290, Cham. Springer International Publishing.

Scherer, S., Agrawal, V., Best, G., Cao, 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., K. 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., Hsuan Lee, Y., Luong, A., Mangelson, J., 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., A. Hollinger, G., and Travers, M. (2021). Resilient and modular subterranean exploration with a team of roving and flying robots. *Submitted to the Journal of Field Robotics*.

Shan, T. and Englot, B. (2018). Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., and Rus, D. (2020). LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE.

Silver, D., Ferguson, D., Morris, A., and Thayer, S. (2006). Topological exploration of subterranean environments. *Journal of Field Robotics*, 23(6-7):395–415. Wiley Online Library.

Tabib, W., Goel, K., Yao, J., Boirum, C., and Michael, N. (2021). Autonomous cave surveying with an aerial robot. *IEEE Transactions on Robotics*, pages 1–17.

Tardioli, D., Riazuelo, L., Sicignano, D., Rizzo, C., Lera, F., Villarroel, J. L., and Montano, L. (2019). Ground robotics in tunnels: Keys and lessons learned after 10 years of research and experiments. *Journal of Field Robotics*, 36:1074 – 1101. Wiley Online Library.

Tatum, M. (2020). Communications coverage in unknown underground environments. Master's thesis, The Robotics Institute, Carnegie Mellon University.

Terry, E., Lei, X., Morrell, B., and Daftry, S. (2020). Object and Gas Source Detection with Robotic Platforms in Perceptually-Degraded Environments. In *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation.

Thakker, R., Alatur, N., Fan, D. D., Tordesillas, J., Paton, M., Otsu, K., Toupet, O., and Agha-mohammadi, A.-a. (2021). Autonomous off-road navigation over extreme terrains with perceptually-challenging conditions. In *Experimental Robotics*, pages 161–173, Cham. Springer International Publishing.

Tidd, B., Cosgun, A., Leitner, J., and Hudson, N. (2021). Passing through narrow gaps with deep reinforcement learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3492–3498. IEEE.

Tranzatto, M., Mascarich, F., Bernreiter, L., Godinho, C., Camurri, M., Khattak, S., Dang, T., Reijgwart, V., Loeje, J., Wisth, D., et al. (2022). Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge. *Field Robotics*, 2:274–324.

Tychsen-Smith, L. and Petersson, L. (2018). Improving object localization with fitness nms and bounded iou loss. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6877–6885. IEEE.

Wellhausen, L. and Hutter, M. (2021). Rough terrain navigation for legged robots using reachability planning and template learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6914–6921. IEEE.

Wellhausen, L. and Hutter, M. (2022). Artplanner: Robust legged robot navigation in the field. In *Submitted to: Field Robotics*.

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 *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 263–270. IEEE.

Yang, B., Wellhausen, L., Miki, T., Liu, M., and Hutter, M. (2021a). Real-time optimal navigation planning using learned motion costs. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

Yang, F., Cao, C., Zhu, H., Oh, J., and Zhang, J. (2021b). FAR Planner: Fast, Attemptable Route Planner using Dynamic Visibility Update. *arXiv preprint arXiv:2110.09460*.

Zhang, J., Kaess, M., and Singh, S. (2016). On degeneracy of optimization-based state estimation problems. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 809–816. IEEE.

Zhang, J. and Singh, S. (2014). LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems X*. Robotics: Science and Systems Foundation.

Zhao, S., Zhang, H., Wang, P., Nogueira, L., and Scherer, S. (2021). Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8729–8736. IEEE.

Özaslan, T., Loianno, G., Keller, J., Taylor, C. J., Kumar, V., Wozencraft, J. M., and Hood, T. (2017). Autonomous navigation and mapping for inspection of penstocks and tunnels with mavs. *IEEE Robotics and Automation Letters*, 2(3):1740–1747. IEEE.

Özaslan, T., Mohta, K., Keller, J., Mulgaonkar, Y., Taylor, C. J., Kumar, V., Wozencraft, J. M., and Hood, T. (2016). Towards fully autonomous visual inspection of dark featureless dam penstocks using mavs. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4998–5005. IEEE.