

# Present and Future of SLAM in Extreme Underground Environments

Kamak Ebadi, Lukas Bernreiter, Harel Biggie, Gavin Catt, Yun Chang, Arghya Chatterjee, Christopher E. Denniston, Simon-Pierre Deschênes, Kyle Harlow, Shehryar Khattak, Lucas Nogueira, Matteo Palieri, Pavel Petráček, Matěj Petrlík, Andrzej Reinke, Vít Krátký, Shibo Zhao, Ali-akbar Agha-mohammadi, Kostas Alexis, Christoffer Heckman, Kasra Khosoussi, Navinda Kottege, Benjamin Morrell, Marco Hutter, Fred Pauling, François Pomerleau, Martin Saska, Sebastian Scherer, Roland Siegwart, Jason L. Williams, Luca Carlone

**Abstract**—This paper surveys recent progress and discusses future opportunities for Simultaneous Localization And Mapping (SLAM) in extreme underground environments. SLAM in subterranean environments, from tunnels, caves, and man-made underground structures on Earth, to lava tubes on Mars, is a key enabler for a range of applications, such as planetary exploration, search and rescue, disaster response, and automated mining, among others. SLAM in underground environments has recently received substantial attention, thanks to the *DARPA Subterranean (SubT) Challenge*, a global robotics competition aimed at assessing and pushing the state of the art in autonomous robotic exploration and mapping in complex underground environments. This paper reports on the state of the art in underground SLAM by discussing different SLAM strategies and results across six teams that participated in the three-year-long SubT competition. In particular, the paper has four main goals. First, we review the algorithms, architectures, and systems adopted by the teams; particular emphasis is put on LIDAR-centric SLAM solutions (the go-to approach for virtually all teams in the competition), heterogeneous multi-robot operation (including both aerial and ground robots), and real-world underground operation (from the presence of obscurants to the need to handle tight computational constraints). We do not shy away from discussing the “dirty details” behind the different SubT SLAM systems, which are often omitted from technical papers. Second, we discuss the maturity of the field by highlighting what is possible with the current SLAM systems and what we believe is within reach with some good systems engineering. Third, we outline what we believe are fundamental open problems, that are likely to require further research to break through. Finally, we provide a list of open-source SLAM implementations and datasets that have been produced during the SubT challenge and related efforts, and constitute a useful resource for researchers and practitioners.

## I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) remains at the center stage of robotics research, after more than 30 years since its inception. SLAM is without a doubt a mature field of research, and the advances over the last three decades keep steadily transitioning into industrial applications, from domestic robotics [1]–[3], to self-driving cars [4] and virtual and augmented reality goggles [5], [6]. At the same time, its pervasive nature and its blurry boundaries as a robotics

This work was supported by the Jet Propulsion Laboratory – California Institute of Technology, under a contract with the National Aeronautics and Space Administration. © 2022. All rights reserved. This work was partially supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111820045. The presented content and ideas are solely those of the authors.

subfield still leave space for exciting research progress. While previous survey efforts have targeted SLAM in general [7], SLAM is also actively investigated in specific subdomains, from deployment on nano-drones [8] to city-scale mapping [9], to deployment in perceptually challenging conditions. This paper surveys algorithms and systems for LIDAR-centric SLAM in extreme underground environments.

### Present and Future of SLAM in Underground Worlds.

The past two decades have seen a growing demand for autonomous exploration and mapping of diverse subterranean environments, from tunnels and urban underground environments, to complex cave networks. This has led to an increasing attention towards underground SLAM, which is a key enabler for navigation in GPS-denied environments where a-priori maps are unavailable. Mature SLAM systems for subterranean mapping have the potential to enable a range of terrestrial and planetary applications, from surveying, search and rescue, disaster response, and automated mining, to exploration of planetary caverns that could hold clues about the evolution and habitability of the early Solar System.

Progress in underground SLAM has been particularly catalyzed by the recent *DARPA Subterranean (SubT) Challenge* [10], a three-year-long global competition ended in 2021, and having the goal of demonstrating and advancing the state of the art in mapping and exploration of complex underground environments. The competition had a *systems* track and a *virtual* track, and included three main events: the Tunnel circuit event, the Urban circuit event, and the Finals. The teams competing in the systems track had the goal of deploying a team of robots to map a sequence of large-scale, unknown underground environments (including caves, tunnels, and subways), detect *artifacts* (*i.e.*, objects of interest, including survivors, mobile phones, fire extinguishers, etc.), and report their locations with stringent performance requirements (*i.e.*, within 5 m errors, in underground networks branching for hundreds of meters to kilometers). While the team of robots was supervised by a single human operator, communication constraints as well as the fast-pace of the competition (the robots had to complete the exploration in under 1 hour) pushed the teams to develop robust and highly autonomous solutions that required minimal human intervention.

**Technical Challenges for Underground SLAM.** Robots exploring underground environments typically do not have access to sources of absolute positioning (*e.g.*, GNSS) and

rarely have access to prior maps of the environment. While in many cases (e.g., search and rescue operations) building a map is not the goal of the deployment, mapping remains a crucial prerequisite for successful underground operation. Mapping these environments is particularly challenging; poor lighting conditions make it challenging to deploy visual and visual-inertial SLAM solutions; while the lack of illumination can be partially compensated by onboard light sources, the resulting illumination is either tenuous or creates specular reflections that interfere with visual feature tracking. Beyond cameras, other sensors are also challenged by the strenuous conditions found in the undergrounds. The potential presence of dense obscurants, such as fog, whirling dust clouds, and smoke, challenges the use of LIDARs. The use of fast-moving platforms on rough terrains induces noise in inertial sensors, due to the aggressive 6-DoF motion and high-frequency vibrations.

Even when the sensors themselves perform to specifications, these environments create further challenges for SLAM algorithms. For instance, the lack of perceptual features (e.g., long corridors, large open spaces, and chambers) induces failures in LIDAR-odometry approaches based on feature or scan matching. Similarly, the presence of self-similar and symmetric areas, and the lack of distinctive visual texture increase the number of false positives in the place recognition methods that fuel loop closure detection in SLAM, and map fusion and merging in multi-robot systems. The complex and ambiguous terrain topography is further exacerbated by sudden changes in the scale of the environment (e.g., a small tunnel leading to a large cave), which clashes with potential scenario-dependent parameter tuning in SLAM systems.

The challenges of underground SLAM extends to system engineering. SLAM algorithms must operate on-board under computational constraints, which are particularly stringent on aerial platforms, and also require careful parameter tuning and code optimizations on wheeled and legged robots. Moreover, these SLAM systems are required to withstand intermittent and faulty sensor measurements, as well as unexpected motions and shocks due to potential robot falls and collisions.

**Related Surveys.** Progress in SLAM research has been reviewed by Durrant-Whyte and Bailey [11], [12] and more recently by Cadena *et al.* [7]. Other relevant surveys have recently focused on multi-robot SLAM and related applications. Kegeleirs *et al.* [13] and Dorigo *et al.* [14] provide an overview of challenges in SLAM with robotic swarms and their application for gathering, sharing, and retrieving information. Halsted *et al.* [15] survey distributed optimization algorithms for multi-robot applications. Parker *et al.* [16] examine multi-robot SLAM architectures with focus on communication issues and their impact on multi-robot teams. Lajoie *et al.* [17] provide a literature review of collaborative SLAM with focus on robustness, communication, and resource management. Zhou *et al.* [18] review algorithmic developments in making multi-robot systems robust to environmental uncertainties, failures, and adversarial attacks. Prorok *et al.* [19] discuss resilience in multi-robot systems. None of these surveys focus on SLAM in underground environments.

**Contribution.** This paper reports on the state of the art and state of practice in underground SLAM by discussing different

SLAM strategies and results across six teams that participated in the three-year-long SubT challenge. In particular, the paper has four main goals. First, we provide a broad review of related work (Section II) and then delve into the single- and multi-robot SLAM architectures adopted by six teams that participated in the systems track of the DARPA SubT challenge (Section III); particular emphasis is put on multi-modal LIDAR-centric SLAM solutions, heterogeneous multi-robot operation, and real-world underground operation. We also discuss the “dirty details” behind the different SubT SLAM systems, which are often omitted from technical papers. Second, we discuss the maturity of the field by highlighting what is possible with the current SLAM systems and what we believe is within reach with some good system engineering (Section IV). Third, we outline what we believe are fundamental open problems, that are likely to require further research to break through (Section V). Finally, we provide a list of open-source SLAM implementations and datasets that have been produced during the SubT challenge and related efforts, and constitute a useful resource for researchers and practitioners. These are summarized in Table I.

## II. OVERVIEW OF RELATED WORK

This section provides a brief overview of related work on SLAM systems for subterranean environments and multi-robot teams, before delving into the details of modern systems in Section III. Early efforts on SLAM in subterranean environments trace back to the work of Thrun *et al.* [20] and Nuchter *et al.* [21], which highlighted the importance of underground mapping and introduced early solutions involving a cart pushed by a human operator, or teleoperated robots equipped with laser range finders to acquire volumetric maps of underground mines. Tardioli *et al.* [22], [23] present a SLAM system for exploration of underground tunnels using a team of robots. The system comprised of a navigation control module, a feature-based robot localization module, a communication module, and a supervisor module for multi-robot collaborative exploration in a tunnel. Zlot *et al.* [24] propose a 3D SLAM system consisting of a 2D spinning lidar and an industrial-grade MEMS IMU to map over 17 km of an underground mine. Kohlbrecher *et al.* [25] present Hector SLAM, a flexible and scalable SLAM system with full 3D motion estimation developed specifically for urban search and rescue. The system consists of a navigation filter that uses an IMU for attitude estimation, and a 2D SLAM system for position and heading estimation within the ground plane.

Lajoie *et al.* [26] present DOOR-SLAM, a multi-robot SLAM system which consists of two key modules, a pose graph optimizer (combined with a distributed pairwise consistent measurement set maximization algorithm to reject spurious inter-robot loop closures), and a distributed SLAM frontend that detects inter-robot loop closures without exchanging raw sensor data. Chang, Tian, *et al.* [27], [28] present Kimaera-Multi, a distributed multi-robot system for dense metric-semantic SLAM. Each robot builds a local trajectory estimate and a 3D mesh. When robots are within communication range, they initiate a distributed place recognition and robust pose graph optimization protocol based on graduated non-convexity.

Autonomous exploration of extreme underground environments has received significant attention in the context of the DARPA SubT Challenge. The competition gave rise to and inspired breakthrough technologies and capabilities in the field of underground SLAM [29]–[54]. We review the details of key (multi-robot) SLAM systems developed in the context of the DARPA SubT challenge in the next section.

### III. STATE OF THE ART IN UNDERGROUND SLAM

This section examines the SLAM architectures adopted by six of the teams that participated in the systems track of the DARPA SubT Challenge, and highlights the important design choices, differences, and common themes that enabled autonomous exploration of unknown underground environments. Moreover, this section provides a table of open-source implementations and datasets that are made publicly available by each team. In particular, Section III-A reviews the standard architecture of a multi-robot SLAM system and provides basic terminology. Section III-B to Section III-G describe the specific SLAM architectures adopted by the six SubT teams and highlight key design choices and “dirty details”. Section III-H discusses common themes, and includes a table of open-source implementations and datasets (Table I).

#### A. Anatomy of Single- and Multi-Robot SLAM Systems

The architecture of a SLAM system typically includes two main components: the front-end and back-end [7];

The **SLAM front-end** is in charge of abstracting the raw sensor data into more compact intermediate representations (*e.g.*, odometry, loop closures, landmark observations). For instance, a LIDAR-based SLAM front-end may process LIDAR scans into odometry estimates either by registering salient features extracted from consecutive LIDAR scans —an approach adopted by teams CERBERUS (Section III-B) and Explorer (Section III-F)— or by dense registration of LIDAR point clouds (or surfels) using ICP or its variants —as adopted by teams CoSTAR (Section III-C), CSIRO (Section III-D), CTU-CRAS-Norlab (Section III-E) and MARBLE (Section III-G).

The **SLAM back-end** is in charge of building robot trajectory and map estimates by fusing the intermediate representations produced by the front-end. The back-end typically includes a nonlinear estimator, with the de-facto standard approach being maximum a-posteriori estimation via *factor graph optimization* [7]; this indeed has been adopted by virtually all teams below. A popular instance of factor graph optimization is *bluepose graph optimization*, where one optimizes the robot trajectory using relative pose measurements. The SLAM back-end can perform tightly-coupled and loosely-coupled sensor fusion, where the former fuses fine-grained measurements by different sensors (*e.g.*, 2D image features and inertial data), while the latter fuses intermediate estimates (*e.g.*, relative poses produced by a LIDAR and camera). Tightly-coupled approaches are generally more accurate, as they rely on more precise models of the sensor data and its noise. Loosely-coupled approaches are easier to implement (*i.e.*, they are more modular) and often more convenient (*e.g.*, they

give access to standard tools for outlier-rejection and health monitoring [55], [56]), but at the cost of decreased accuracy.

**Multi-robot SLAM** systems are characterized by the fact that sensor data is simultaneously collected by multiple robots, which are in charge of building a consistent map of the environment. Multi-robot SLAM architectures can be *centralized*, *decentralized*, or *distributed*. In centralized architectures, a base station collects data from all the robots (*e.g.*, raw sensor data or intermediate representations from the single robot front-ends) and then computes optimal trajectory and map estimates for the entire team. Each robot typically runs a local SLAM front-end (and possibly a local back-end) to pre-process the sensor data — this reduces the amount of data to be transmitted and the subsequent computation at the base station; then, the base station may implement a *multi-robot front-end*, which is in charge of detecting inter-robot loop closures, and a *multi-robot back-end*, that estimates the robots’ trajectories and map. In this paper, we call a multi-robot architecture *decentralized* if each robot is treated as a base station: it collects all the data from the other robots and performs joint estimation of the trajectory and global map of the entire team. Finally, we call an architecture *distributed* if each robot only exchanges partial information with its neighbors and only estimates its own map by relying on distributed inter-robot loop closure detection and distributed optimization protocols [28], [57]–[60]. The following sections describe the SLAM architectures for each SubT team.

#### B. Team CERBERUS

Team CERBERUS won the Final event of the DARPA SubT Challenge; their SLAM architecture is given in Figure 1. The architecture is powered by *CompSLAM* [46], a complementary multi-modal odometry and local mapping approach running at each (walking, flying, or roving) robot, and *M3RM*, a multi-modal, multi-robot mapping server running at the base station.

**Onboard Odometry and Mapping via CompSLAM.** CompSLAM [46] is a loosely-coupled approach that allows hierarchical fusion of a set of sensor-specific pose estimators as each estimate is refined by the next estimator. This enables operating in parallel into a single odometry estimate, while performing data- and process-level health checks [61]. In particular, CompSLAM performs a coarse-to-fine fusion of independent pose estimates including visual, thermal, depth, inertial, and possibly kinematic odometry sources. This loosely-coupled methodology provides redundancy and ensures robustness against perceptually degraded conditions, including self-similar geometries, low-light and low-texture scenes, and obscurants-filled environments (*e.g.*, fog, dust, smoke), assuming that each condition only affects a subset of sensors.

The visual- and thermal-inertial fusion (VTIO) components of CompSLAM build upon the work [62] and extends it to exploit 16-bit raw data from LongWave InfraRed cameras [63], [64] and depth from LIDAR. Furthermore, the depth data from the LIDAR is utilized to initialize or improve depth estimates of features tracked in visual and thermal imagery, providing robustness for scale estimation without the need for computationally expensive stereo-matching.

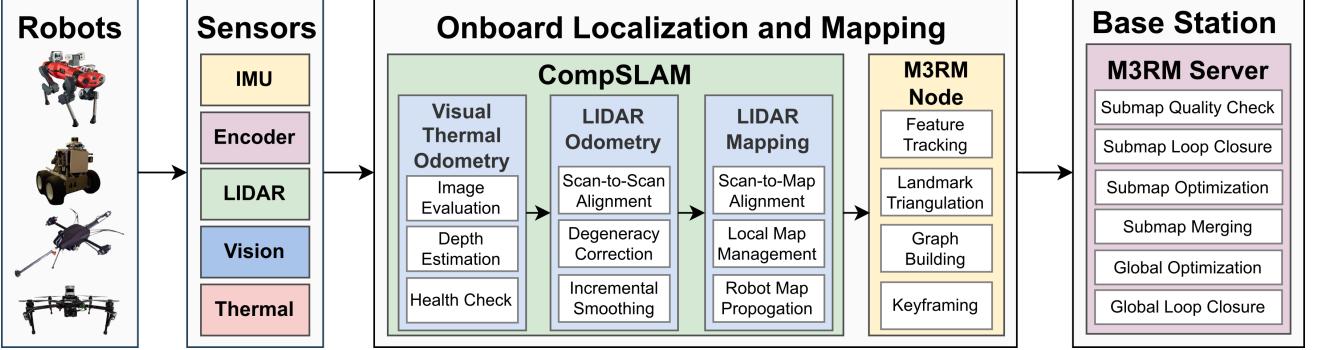


Figure 1: Overview of team CERBERUS’ SLAM architecture. Each robot estimates and operates on its own individual map. Periodically, these maps are sent to the mapping server on the base station for accumulation and for global multi-robot optimization.

The LIDAR Odometry And Mapping component of CompSLAM develops on top of LOAM [65]. This component, along with VTIO priors, utilizes LIDAR point clouds to perform a LIDAR Odometry (LO) scan-to-scan matching and scan-to-submap matching LIDAR Mapping (LM) step. Accordingly, the robot estimates its pose in the map and simultaneously constructs a local map of the environment. Following the hierarchical fusion approach, the estimates of the LO module are utilized by and refined upon by the LM module. To assess the quality at each iterative optimization step, the system utilizes a threshold on the eigenvalues of the underlying approximate Hessian [31], [66], to identify the degrees of freedom that are possibly ill-conditioned due to geometric self-similarity. In case certain directions are determined to be ill-conditioned, the pose estimates from the previous estimator in the hierarchy (e.g., visual-inertial odometry) are propagated forward, skipping the ill-conditioned module.

Finally, to produce smooth and consistent pose estimates, CompSLAM uses a factor-graph-based fixed-lag smoother, implemented as part of the LO module, with a smoothing horizon of 3 seconds. The factor graph is implemented using GTSAM [67] and integrates relative LO estimates with IMU pre-integration factors [68]. To reduce pose drift and improve IMU bias estimation, zero-velocity factors are added when more than one sensing modality reports no motion for 0.5 (consecutive) seconds. Moreover, during periods of no motion, roll and pitch estimates —calculated directly from bias-compensated IMU measurements— are added as prior factors.

**Multi-Robot Mapping and Optimization (M3RM).** The core component of the CERBERUS multi-modal and multi-robot mapping (M3RM) approach is a centralized mapping server that utilizes multiple modalities such as LIDAR, vision, IMU, wheel encoders, etc., in a single factor graph optimization. The deployed M3RM approach is based on the existing framework *maplab* [69] and can generally be subdivided into two components, namely the *M3RM node* and *server*.

The M3RM node runs onboard each robot and is in charge of creating a local factor graph capturing multi-sensor data collected and pre-processed by the robot, *e.g.*, odometry factors from CompSLAM. The node also tracks BRISK [70] features and triangulates the features to a visual map using the CompSLAM pose estimates. Additionally, the LIDAR scans (as

well as the corresponding timestamps and extrinsic calibration) are attached to the factor graph. The factor graph is broken into submaps. To reduce bandwidth, each LIDAR scan is compressed using DRACO [71] before transmission, reaching a total size of approximately 2 megabytes per submap. When robots establish a connection to the base station, the M3RM node transmits the completed submaps to the M3RM server. A synchronization logic ensures that only a completed submap transmission will be integrated into the multi-robot map.

The M3RM server runs at the base station and is in-charge of keeping track of all individual submaps for each robot and integrating them into a globally consistent multi-robot map. During the mission, the M3RM server allows a human operator to visualize the individual maps as well as the globally optimized multi-robot map which enables mission planning. Moreover, the server has certain management functions such as removal of maps, performance profiles, and allows switching between CompSLAM and M3RM map per robot. The CompSLAM maps are not attached to the global multi-robot map but can be visualized using an overlay. To integrate the individual robot submaps into a single multi-robot map, the M3RM server first processes each incoming submap using a set of operations, namely (i) visual landmark quality check, (ii) visual loop closure detection, (iii) LIDAR registrations, and (iv) submap optimization. Since each submap’s processing is independent of the processing of other submaps, the mapping server can process up to four submaps in parallel. For visual loop closure detection, the method presented in [72] is performed using the tracked BRISK features and an inverted multi-index. Correctly identified visual loop closures within a submap are implemented by merging the corresponding landmarks and are then integrated during the submap optimization. Moreover, additional LIDAR constraints are added to the factor graph by aligning consecutive scans within a submap using ICP. Since the onboard odometry and mapping pipeline already provides an estimate of the poses, a prior transformation is readily available for each registration. However, if the resulting transformation differs significantly from the prior, it is rejected for robustness reasons as we expect that the drift between consecutive nodes is relatively small.

After individual submaps processing, they are merged into the global multi-robot map, which is continuously optimized

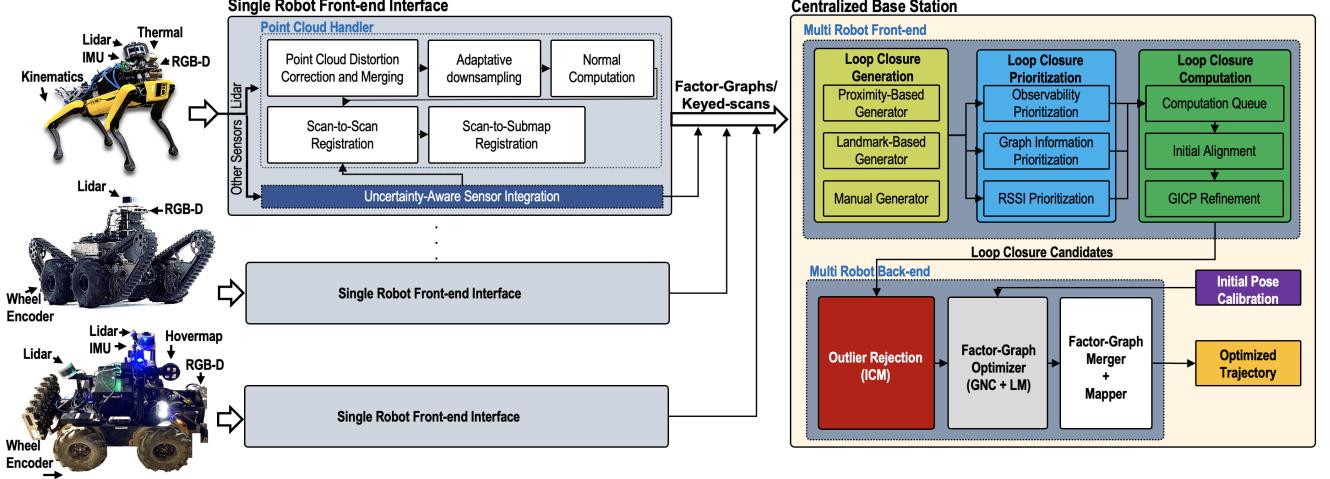


Figure 2: Overview of team CoSTAR’s SLAM architecture (LAMP). Each robot runs a local front-end and communicates to the base station, which runs a multi-robot front-end (for loop closure detection) and back-end (for pose graph optimization).

by the M3RM server. A predefined set of operations are executed in an endless loop on the global multi-robot map, i.e., (i) multi-robot visual loop closure detection, (ii) multi-robot LIDAR registrations, and (iii) factor graph optimization. In this case, these operations are performed on the entire multi-robot map, and have the goal of detecting intra- and inter-robot loop closures and performing a joint optimization.

**Dirty Details.** *Parameter tuning:* As for many other systems reviewed in this paper, the top performance of team CERBERUS’ SLAM solution requires a careful fine-tuning of all available parameters. For example, the degeneracy detection relies on a hand-tuned set of parameters and is robot-dependent. The tuning is performed by using a grid search over several clusters of parameters and measuring their performance across relevant environments. To complicate things further, the configurable parameters for the M3RM server have to be consistently applied to all robots in the global multi-robot map, making fine-tuning for specific robot types (e.g., flying and legged systems) as well as sensors (e.g., various camera and LIDAR systems) difficult at the stage of global mapping.

**Covariances:** While it is desirable to dynamically adjust the covariances in the factor graphs depending on the quality of the sensor data, it proved challenging to balance the uncertainty of the visual and LIDAR factors; therefore, the system relied on static (manually tuned) covariances for the latter.

**Loop closures:** None of the deployed robots performed onboard loop closure detection. Thus, in scenarios where robots stay out of communication range from the base station for a considerable time, the CompSLAM errors may accumulate, making it harder for the M3RM server to correct the estimates. Moreover, an incorrect robot map can “break” the whole global multi-robot map, which is why a human operator is needed to monitor and possibly remove specific robots from the multi-robot map.

### C. Team CoSTAR

Team CoSTAR won the Urban event of the DARPA SubT Challenge. An overview of team CoSTAR’s SLAM system,

namely, *Large-scale Autonomous Mapping and Positioning* (LAMP), is provided in Figure 2. LAMP is a key component of NeBula [73], team CoSTAR’s overall autonomy solution. LAMP relies on data from different odometry sources (i.e., LIDAR, visual-inertial, wheel-inertial, and IMU) to estimate the robot trajectories, as well as a point cloud map of the environment. The system consists of (i) a *single-robot front-end interface* that runs locally onboard each robot to produce an estimated robot trajectory and a point cloud map of the environment explored by each robot, (ii) a *multi-robot front-end*, running on the base station, which receives the robots’ local odometry and maps and performs multi-robot loop closure detection, and (iii) a *multi-robot back-end*, that uses odometry (from all robots) and intra- and inter-robot loop closures from the multi-robot front-end to perform a joint pose graph optimization; the multi-robot back-end runs on the base station and simultaneously optimizes all the robot trajectories.

**Single-Robot Front-End Interface.** LAMP relies on a multi-sensor front-end interface that enables the use of robots with different sensor configurations and odometry sources, including LOCUS [74] and Hovermap [75]. The front-end produces an odometric estimate of each robot’s trajectory, and stores the corresponding information in a factor graph, where each node corresponds to an estimated pose, while an edge connecting two nodes encodes the relative motion between the corresponding timestamps. Each odometry node is associated with a keyed-scan, a pre-processed point cloud obtained at the corresponding timestamp. The keyed-scan are used for loop closure detection and to form a 3D map of the environment.

Within the single-robot front-end, LOCUS [74] is CoSTAR’s LIDAR-centric odometry estimator. LOCUS starts with a pre-processing step, where —after removing motion-induced distortions in point clouds— scans from multiple onboard LIDARs are merged into a unified point cloud given the extrinsic calibration between LIDARs. An adaptive voxelization filter is then applied to ensure a constant number of points are retained independent of the environment geometry, point cloud density, and number of onboard LIDARs. This helps reduce the computation, memory usage, and communication

bandwidth associated with the subsequent processing. Odometric estimates are obtained using a two-stage scan-to-scan and scan-to-submap registration process; the registration relies on a fast implementation of point-to-plane ICP, initialized using IMU measurements or other odometry sources.

**Scalable Multi-robot Front-end.** The multi-robot front-end is in charge of intra- and inter-robot loop closure detection by leveraging a three-step process: loop closure generation, prioritization, and computation as outlined below.

The *Loop Closure Generation* module relies on a modular design, where loop closure candidates can be identified using different methods and environment representations (i.e., Bag-of-visual-words [76], junctions extracted from 2D occupancy grid maps [31]). The go-to loop closure generation approach within SubT has been based on LIDAR point clouds. In particular, loop closure candidates are simply identified from nodes in the factor graph that lie within a certain Euclidean distance from the current node; the distance is dynamically adjusted to account for the odometry drift between nodes.

The *Loop Closure Prioritization* module [77] selects the most promising loop closures for processing. While loop closures are crucial for map merging and drift reduction in the estimated robot trajectory, it is equally crucial to avoid closing loops in ambiguous areas with high degree of geometric degeneracy [31], as it could lead to spurious loop closure detections. Furthermore, loop closure detection in large-scale environments, and with large number of robots, becomes increasingly more computationally expensive as the density of nodes in the pose graph, and subsequently the number of loop closure candidates, increases. The purpose of this module is to prioritize loop closure candidates inserted in the computation queue by evaluating their likelihood of improving the trajectory estimate. This is achieved through a three-step process of (i) observability prioritization, where similar to the works presented in [31], [78], [79], eigenvalue analysis is performed to detect degenerate scan geometries, in order to prioritize loop closures in feature-rich areas, (ii) graph information prioritization, where a Graph Neural Network (GNN) [80] based on a Gaussian Mixture model layer is used to predict the impact of a loop closure on pose graph optimization, and (iii) Receiver Signal Strength Indication (RSSI) prioritization to prioritize loop closures based on known locations indicated by RSSI beacons —whenever a robot is within range of an RSSI beacon. The prioritized loop closure candidates are inserted into a queue for the computation step in a round-robin fashion.

The *Loop Closure Computation* module estimates the relative pose between a pair of loop closure candidate nodes in the queue using a two-stage process. First, an initial estimate of the relative pose is computed using TEASER++ [81] or SAmple Consensus Initial Alignment (SAC-IA) [82]. Then the Generalized Iterative Closest Point (GICP) algorithm [83] is initialized with the obtained solution to refine the relative pose and evaluate the quality of the LIDAR scan alignment.

**Robust Multi-robot Back-end.** LAMP uses a centralized multi-robot architecture, where a central base station receives the odometry measurements and keyed scans from each robot, along with loop closures from the multi-robot front-end, and performs pose graph optimization to obtain the optimized

trajectory for the entire team. The optimized map is then generated by transforming the keyed scans to the global frame using the optimized trajectory. To safeguard against erroneous loop closures, the multi-robot back-end includes two outlier rejection options: Incremental Consistency Maximization (ICM) [29], which checks detected loop closures for consistency with each other and the odometry before they are added to the pose graph, and Graduated Non-Convexity (GNC) [55], which is used in conjunction with the Levenberg-Marquardt solver to perform an outlier-robust pose graph optimization and obtain both the trajectory estimates and inlier/outlier decisions on the loop closure not discarded by ICM. Pose Graph Optimization and GNC are implemented using GTSAM [67].

**Dirty Details.** *Parameter tuning:* While LAMP provides a robust localization and mapping framework, it is difficult to find a set of parameters for the front-end and back-end modules that leads to nominal performance consistently across environments with different topography and geometry. In order to have a more systematic approach to parameter tuning, CoSTAR curated 12 SLAM datasets across multiple challenging underground environments for evaluation and benchmarking, with the goal of obtaining a set of parameters that gave the best performance across all domains. The parameter tuning was mostly manual, and was restricted to a small subset of parameters which had higher impact on the system’s performance. One area where parameter tuning was successful was LIDAR-based loop closure detection. Here, the dataset consisted of pairs of point clouds from a variety of environments, with 80% of the pairs being true loop closures, with known relative poses, and the rest being outliers.

#### D. Team CSIRO

Team CSIRO Data61 tied for the top score and won the second place at the Final event of the DARPA SubT challenge after the tiebreaker rules were invoked. The team also won the single most accurate artifact report award in the Urban and Final events. An overview of *Wildcat* [84], [85], CSIRO’s LIDAR-inertial decentralized multi-robot SLAM system, is given in Figure 3. We first review CSIRO’s distinctive sensing strategy, and then introduce the key modules in the Wildcat architecture: surfel generation, LIDAR-inertial odometry, frame generation and sharing, and pose graph optimization.

**Sensing Pack.** The ground robots carried a CatPack sensing payload designed by CSIRO. The CatPack uses an IMU and a Velodyne VLP-16 LIDAR that is mounted at 45° off horizontal and spins about the vertical axis of the CatPack. The CatPack also has four RGB cameras, which were used for artifact detection, but not for SLAM. The Emesent Hovermap [86] payload used on the aerial robots is a similar sensing pack with a spinning Velodyne VLP-16. Both the CatPack and Hovermap run the Wildcat SLAM system onboard, on their NVIDIA Jetson AGX Xavier and Intel NUC computers, respectively.

The spinning LIDAR configuration of CatPack provides dense depth measurements with an effective 120° vertical field of view. This played a major role in making CSIRO’s SLAM system robust in subterranean environments, *e.g.*, by providing improved visibility of the floor and roof of narrow tunnels. It

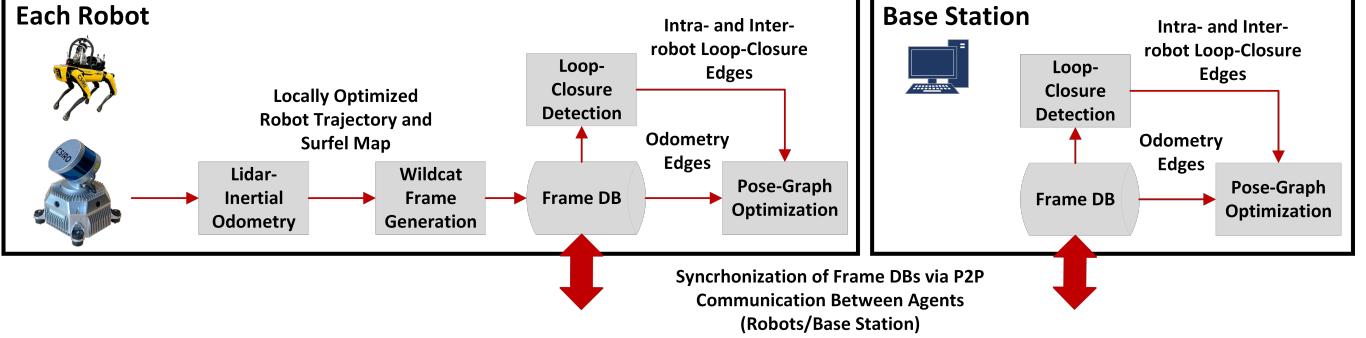


Figure 3: Overview of CSIRO’s decentralized multi-robot SLAM system in SubT. Each robot runs its own Wildcat’s LIDAR-inertial odometry module independently. The resulting locally-optimized odometry estimate and surfel submaps are used to generate Wildcat *frames*. These frames are stored in a database and are shared with other robots and the base station. Robots and the base station then use their collection of frames to independently build and optimize a pose graph.

also enabled use of surfel features, which exploit the dense depth measurements to provide a stable, robust feature set that is effective in a wide range of environments.

**Surfel Generation.** Wildcat uses planar surface elements (surfels) as dense features for estimating robot trajectory. Surfels are created every 0.5 s by spatial and temporal clustering of new LIDAR points. Specifically, the space is voxelized at multiple resolutions and points are clustered depending on their timestamp and the voxel they fall in. Clusters smaller than a predefined threshold (in terms of number of points) are discarded. An ellipsoid is then fit to each remaining cluster by computing the first two moments of its 3D points. The centroid (mean) of an ellipsoid specifies the position of the corresponding surfel, while its covariance matrix determines its shape. A planarity score [87, Eq. 4] is computed based on the spectrum of the covariance, and only sufficiently planar surfels are kept.

**LIDAR-Inertial Odometry.** Wildcat’s LIDAR-inertial odometry module processes surfels and IMU data in a sliding window. Within a time window, the processing alternates between (i) matching active surfel pairs and (ii) optimizing robot trajectory, for a predetermined number of times or until satisfying a convergence criterion. Surfel correspondences are established through  $k$ -nearest (reciprocal) neighbor search in the descriptor space comprising estimated surfel’s position, normal vector, and voxel size. The estimate of the segment of the robot trajectory within the current time window is then updated by minimizing a cost function mainly composed of residual error functions associated to matched surfel pairs and IMU measurements in the current time window. The cost function is made robust to outliers (*e.g.*, incorrect surfel correspondences) by using the Cauchy M-estimator.

**Frame Generation and Sharing.** A Wildcat *frame* comprises a six-second portion of surfel map and odometry produced by each robot’s LIDAR-inertial odometry. Each robot generates frames periodically and stores them in a database. A frame is discarded if its surfel submap has very high overlap with that of the previous frame. As shown in Figure 3, Wildcat leverages CSIRO’s peer-to-peer ROS-based data sharing system, *Mule* [85, Section 4.3], to synchronize robots’ frame databases every time two agents (robot-robot or robot-base

station) are within communication range.

**Pose Graph Optimization.** Each robot uses its collection of Wildcat frames —including those generated and shared by other robots— to independently build and optimize the team’s collective pose graph. Frames represent nodes of the pose graph. Each robot’s odometry estimate is used to create odometry edges (*i.e.*, relative pose measurements) between the robot’s consecutive frames. Additionally, intra- and inter-robot loop-closure edges are created by aligning frames’ surfel maps. This is done using ICP for nearby frames (for which a good initial guess is available from odometry) and global registration methods for distant ones. Pose graph nodes with significant overlap in their local maps are merged together. As a result, the computational complexity of the solver grows with the size of the explored environment rather than mission duration. The solver is made robust to outliers using Cauchy M-estimator. The collective pose graph built and optimized by each robot is used to render a surfel map of the environment.

**Dirty Details.** *Parameter tuning:* CSIRO’s solution uses a single set of parameters tuned to perform across a wide range of environments. However, ground robots and drones use different parameters due to the independent tuning processes.

*Calibration:* CatPacks undergo extensive calibration on production, comprising both LIDAR-IMU and LIDAR-camera calibration. The incorporation of the cameras in the CatPack successfully avoided the need for subsequent calibration, even when packs are switched between platforms.

*Loop closures:* Complex LIDAR-based place recognition techniques were rarely found to be necessary at the scale of SubT environments, therefore team CSIRO found loop closures candidates by searching for past poses within a Mahalanobis distance from the current robot pose. In SubT, the first inter-robot loop closures were created upon startup based on joint observation of the same starting region. This process at startup was imperfect, but difficulties could be addressed procedurally, *e.g.*, by restarting the affected agent. Since each agent also solves independently for its own multi-robot solution, it was necessary to ensure that these inter-robot loop closures are successfully detected not only on the base, but also on each robot. After difficulties in the Urban Event of the competition, user interface elements were introduced

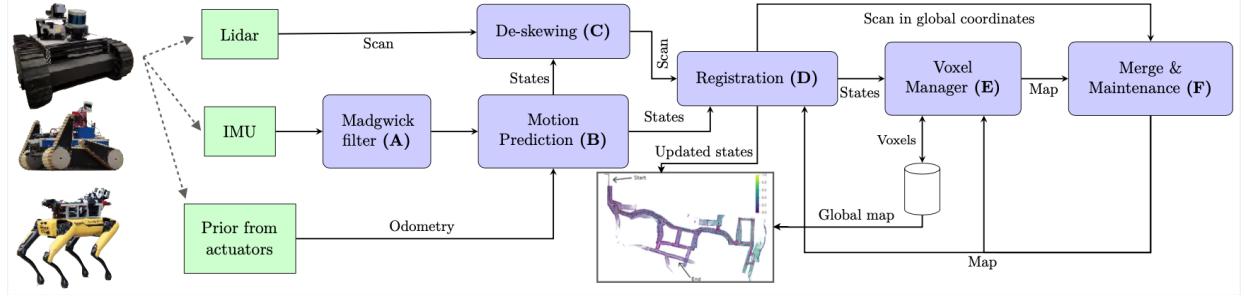


Figure 4: CTU-CRAS-Norlab UGV SLAM architecture (Norlab ICP Mapper). Green boxes correspond to the inputs and output, and the purple ones represent submodules of the SLAM architecture.

to prominently report the connectivity status of the collective pose graph to detect anomalies. By the Final event, these failures were rare.

**Hardware robustness:** While hardware robustness is seldom discussed in the SLAM literature, it is a significant feature of the CSIRO system’s maturity. On the rare occasions when Wildcat diverged in testing, almost all occurrences were found to coincide with sensor dropouts caused by significant kinetic impacts of the platform, or hardware failures, which typically start with intermittent errors.

#### E. Team CTU-CRAS-Norlab

The CTU-CRAS-Norlab team employed two separate SLAM systems for their Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs). The corresponding architectures are given in Figure 4 and Figure 5, respectively.

**UGV SLAM.** The UGV SLAM architecture relies exclusively on a LIDAR odometry system, Norlab ICP Mapper, that focuses on reducing drift at the front-end level. The mapper operates as follows: (A) first, the robot orientations during a LIDAR scan are estimated by passing the IMU measurements through a Madgwick filter [88]. (B) then, this orientation information is fused with translation estimates from wheel odometry to estimate the robot motion during the scan. (C) the motion estimate allows to de-skew the current LIDAR scan (*i.e.*, motion correction). (D) once de-skewed, the LIDAR scan is registered in the local map using ICP, taking the robot pose as prior. A modified version of point-to-plane ICP [89] is used, where only 4 degrees of freedom (3D position and yaw angle of the scan) are optimized, while roll and pitch angles are directly obtained from the IMU. (E) the robot pose found using registration, is used by the voxel manager to load and unload voxels of the local map to ensure it stays centered on the robot. (F) lastly, the registered cloud is merged into the local map and maintenance operations are performed. These maintenance operations include identifying and removing points belonging to dynamic objects using the technique described in [90]. The resulting map is then set as the new local map. These steps are performed in different threads to allow the system to localize at a higher rate than the rate at which the map is updated.

**UAV SLAM.** The UAV SLAM architecture relies on a LIDAR sensor that is complemented by an IMU for precise roll-and-pitch orientation estimation. While not necessary for localization, which utilizes only LIDAR and IMU measurements, data from upward- and downward-facing cameras

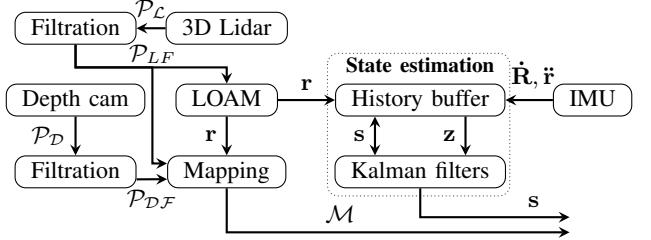


Figure 5: CTU-CRAS-Norlab UAV SLAM architecture.  $\mathcal{P}_D$  and  $\mathcal{P}_{LC}$  are the depth camera and 3D LIDAR point clouds.  $\mathcal{P}_{DF}$  and  $\mathcal{P}_{LCF}$  are the respective point clouds after filtration. The outputs are the map  $\mathcal{M}$  and the state  $s$ , which consists of position  $r$ , orientation  $R$ , and their first derivatives  $\dot{r}$  and  $\dot{R}$ . The Kalman filter corrections  $z$  consist of  $r$ ,  $\dot{r}$ , and  $\ddot{r}$ .

are integrated into a dense metric map to cover the blind spots of the LIDAR field of view. The output of the system is a state estimate (*i.e.*, robot poses in a gravity-aligned reference frame, and their derivatives), and a volumetric occupancy map.

The UAV SLAM pipeline (Figure 5) starts with pre-processing of LIDAR scans. First, a range-clip filter is applied to the raw scans to filter out the robot frame and distant measurements. Second, a local intensity-threshold filter is applied to the data, which proved to be a highly robust method for filtration of dust even in the harshest conditions. Due to computational constraints, this pre-processing does not involve LIDAR scan de-skewing; while this negatively impacts the SLAM performance, it reduces the delay incurred by the pose estimate. The processed data is passed to LOAM [65], which optimizes the alignment of geometric features extracted from the data in a two-step odometry process — fast scan-to-scan and slow scan-to-map matching in the feature space. The team has adapted the advanced implementation of LOAM (A-LOAM<sup>1</sup>) to be suitable for UAVs by extending the method with platform-optimized parallelization. The state estimation module (based on [35]) takes the LOAM pose estimate and fuses it with the IMU measurements using a linear Kalman filter to obtain a high-rate delay-compensated state estimate that is suitable for the control system feedback loop [91]. The non-constant delay introduced by LOAM negatively impacts the controller performance and, most importantly, the control error. The idea of the delay-compensation method [92] is to recompute the current state if a measurement with a past

<sup>1</sup><https://github.com/HKUST-Aerial-Robotics/A-LOAM>

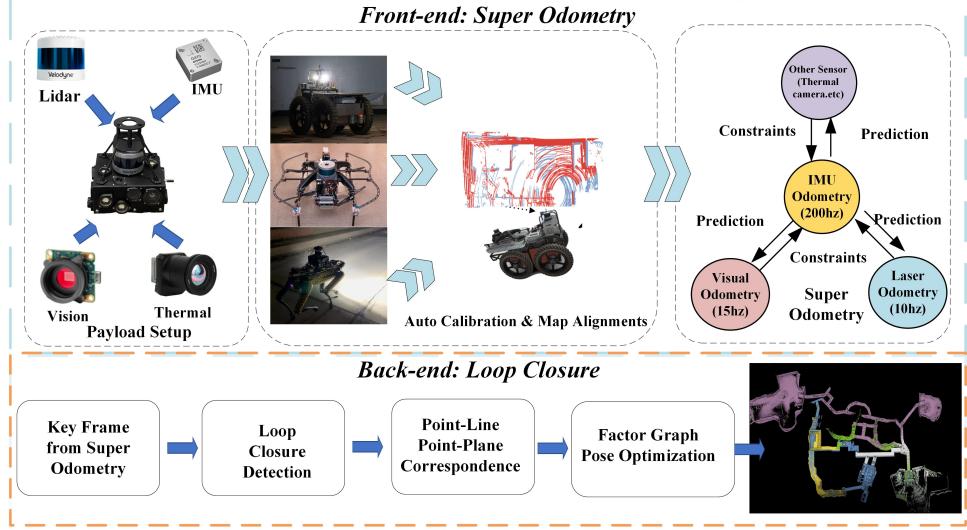


Figure 6: Overview of team Explorer’s SLAM architecture. Each robot estimates and operates on its own individual map.

timestamp arrives. When a delayed measurement arrives, it is applied to the state in a circular buffer with the nearest timestamp. The corrected state is then propagated to the current time using the system model and other relevant updates.

The approach is fully decentralized, with each UAV running its own SLAM pipeline. To allow multi-robot cooperation, the reference frames of UAVs are initially aligned by one of the following procedures. The reference frames are either given in advance (*e.g.*, from a total station) or their alignment is estimated with respect to a leader-robot by scan matching using LIDAR data shared among all the UAVs before takeoff.

**Dirty Details. Loop closures:** The only tunable parameter of the UAV SLAM method is the resolution of the feature map. For both SLAM systems, it was found empirically that one set of parameters worked well in a majority of scenarios; any changes to the parameters led to degraded state estimation quality or slower-than-real-time performance. Not adapting the parameters dynamically also ensured static assignment of computational resources, which helped to predict and optimize the system behavior under resource constraints.

**Loop closures:** Neither the UGV nor the UAV SLAM systems detect loop closures; therefore, no pose graph optimization is used to refine the odometric trajectories.

**Computation prioritization:** When the CPU is fully loaded, critical components such as control and SLAM might have to wait for CPU resources shared with non-flight-critical software such as object detectors, which results in triggering failsafe recovery behaviors. Prioritizing the critical modules at the process level by reducing CPU affinity and using negative nice [93] values for non-critical processes resulted in lower computation times, lower jitter, and smoother flights. Additional performance was gained by running the algorithms that process large amounts of data as nodelets under a common ROS nodelet manager. This avoids the overhead of copying large data structures by simply passing pointers instead.

#### F. Team Explorer

Team Explorer won the Tunnel event of the DARPA SubT Challenge. Team Explorer’s SLAM architecture is given in Figure 6. The architecture relies on Super Odometry [94] to fuse outputs of multiple odometry sources including visual or thermal fusion [95] using a probabilistic factor graph optimization, and a loop-closing back-end.

**Lidar-Inertial Localization Module for Odometry Estimation.** The lidar-inertial localization module relies on Super Odometry (SO) [94]. In SO, a factor graph optimization performs estimation over a sliding window of recent states by combining IMU pre-integration factors with point-to-point, point-to-line, and point-to-plane LIDAR factors. SO strikes a balance between loosely- and tightly-coupled estimation methods. The IMU-centric sensor fusion architecture does not combine all sensor data into a full-blown factor graph. Instead, it breaks it down into several “sub-factor-graphs”, with each one receiving the prediction from an IMU pre-integration factor. The motion from each odometry factor is recovered in a coarse-to-fine manner and in parallel, which significantly improves real-time performance. SO enables achieving high accuracy and operates with a low failure rate, since the IMU sensor is environment-independent, and the architecture is highly redundant. As long as other sensors can provide relative pose information to constrain the IMU pre-integration results, these sensors will be fused into the system successfully.

**Loop-Closing Back-End.** While SO is a low-drift odometry algorithm, it is still important for the SLAM system to be able to correct long-term drift. This is specially true when the traversed distance is high. Considering that Explorer’s ground robots moved at  $0.5 \text{ m s}^{-1}$  and had an aggressive exploration style, it was not uncommon to observe traversed distances larger than 1 km in a single test. Team Explorer’s solution reduces the drift by detecting loop closures and performing pose graph optimization. In particular, the back-end filters the poses and point clouds generated by the front-end. It applies a heuristic method to accumulate these results into a *keyframe*, which is composed by a *key pose* and a *key cloud*, which

is the point cloud generated by accumulating all the point clouds generated since the last keyframe, and downsampling to maintain a fixed size. The heuristic used is distance-based: a new keyframe is created after the robot moves by 0.2 m. Search for loop closures is performed using a radius-search among the nearest poses, or by querying a database of sensor data to find matches with previously visited places.

**Autocalibration.** To achieve a common task, multiple robots need to be able to establish and operate in a common frame of reference. Towards this goal, team Explorer used *Autocalibration*, a process in which a Total Station is used to obtain the pose of one robot with respect to the fiducial markers with known positions in the world frame set by DARPA. This robot shares its pose with respect to its own map frame and the latest three keyframes it created. All the following robots will then be placed near the calibration location of the first robot; they receive the reference information from the base station, and use GICP [83] to align their current keyframes to establish their initial pose in the world frame.

**Dirty Details.** *Loop closures:* The most important parameters tuned during testing were those related to the downsampling of the point clouds before the scan-to-map registration. This downsampling affected the number and the quality of features available to SO. In particular, a key parameter is the voxel size used in the PCL library voxel grid filter. When the robot traverses a narrow urban corridor, it is desirable to use a smaller voxel size, to avoid decimating important details in the point cloud. In contrast, in a large cave, a larger voxel size is required, otherwise the processing becomes too slow due to the large number of features. To solve this problem, we created an heuristic method to switch voxel sizes in real-time. The method consists in calculating, for each 3D axis separately, the average distance to the points in the current point cloud. Then, we multiplied the 3 values together to obtain an “average volume”. This volume was thresholded to create 3 different modes, each associated to a predefined voxel size.

*Dust Filters:* Team Explorer had a strong focus on UAVs. These platforms bring their own unique challenges to the SLAM problem. One that was particularly important for SubT was being able to handle the dust that arises due to the robot’s propellers. The simple solution that was implemented was to test if there was a minimum number of features farther than 3 m from the robot. If true, all the other points inside this radius were ignored when performing pose estimation. This solution was based on the assumption that dust would usually accumulate circularly around the robot, but usually there are still other distant features in the environment that allow the robot to solve the optimization correctly. If the robot were capable of performing estimation and continuing operation, it would usually escape the dusty area in the environment. Otherwise, dust would eventually cover the robot from all sides and a catastrophic failure would occur.

*Robot-specific computational budget:* Analysis of empirical results showed that after a certain number, having more LIDAR features does not necessarily translate to substantial accuracy gains. Therefore, a threshold on the number of surface features is used. If the current scan frame contains more than the threshold, the list of features is sampled uniformly

such that the number of features does not exceed the threshold.

#### G. Team MARBLE

Team MARBLE’s SLAM architecture is given in Figure 7. The core of MARBLE’s LIDAR-centric solution<sup>2</sup> is the open-source LIO-SAM [98] package, which performs tightly-coupled fusion of IMU data and LOAM-based LIDAR features [65]. The localization results are then passed to MARBLE Mapping, that creates a voxel map.

**LIDAR Localization via LIO-SAM.** Each robot in the MARBLE system was responsible for its own localization, from input (*i.e.*, LIDAR scans at 20 Hz and IMU data at 500 Hz) to optimization. The localization process includes multiple subcomponents (Figure 7). First of all, in order to be processed by LIO-SAM, each point in the LIDAR point cloud required two extra data fields in addition to the standard  $x, y, z$  position: a timestamp, and a ring number to provide their relative position in the vertical scan. This additional data is used to de-skew the point clouds. While current Velodyne LIDAR and Ouster-OS1 LIDAR drivers provide this information by default, the Ouster-OS1 LIDAR driver required some slight modification to enable this information. In particular, timestamps were added to each vertical angle of arrival, and rings were designated by their elevation angle.

Localization via LIO-SAM is based on factor graph optimization and involves three types of factor. The first type consists in IMU pre-integration factors [68]. The second type includes LIDAR odometry factors; in particular, once the LIDAR has been de-skewed, LIO-SAM extracts key features along lines and edges (as in LOAM [65]). These features are then compared and scan-matched along a subset of local key frames in a sliding window filter. Lastly, loop closure factors are determined by a naive Euclidean distance metric. Each time a new factor is added to the graph, the iSAM2 solver [99] is applied to optimize the graph using GTSAM [67]. After generating an odometry estimate, LIO-SAM then estimates the IMU bias with the updated odometry.

**Multi-Robot Mapping via Octomap.** LIO-SAM outputs robot pose estimates. A voxel based map can also be queried via a ROS service call, however, MARBLE relied on a separate custom package, MARBLE Mapping, a fork of Octomap [100], which allowed creating voxel grid map differences with low data transfer requirements. In particular, MARBLE Mapping uses the latest LIO-SAM pose estimate and the corresponding LIDAR scan to update the log-odds probability (occupancy) value inside an octomap with voxel size of 0.15 m. When enough voxels have been added or have changed state, or if enough time or distance has been traversed, a new map difference is created by the robot with the changed voxels.

Map differences are then shared between robots in a peer-to-peer fusion. Each robot tracks differences in a sequence tied to

<sup>2</sup>Initially, MARBLE explored the use of onboard cameras and a visual-inertial odometry system [38], namely, Compass [96]. While this solution was tenable in some cases, changing lighting conditions and specular highlights caused by on board illumination in dark scenes often lead to instability. This was especially true in longer deployments, such as the hour-long runs necessitated by SubT. A dataset for benchmarking visual-inertial SLAM systems with onboard illumination was released as a part of these tests [97], however team MARBLE switched to a LIDAR based architecture soon after.

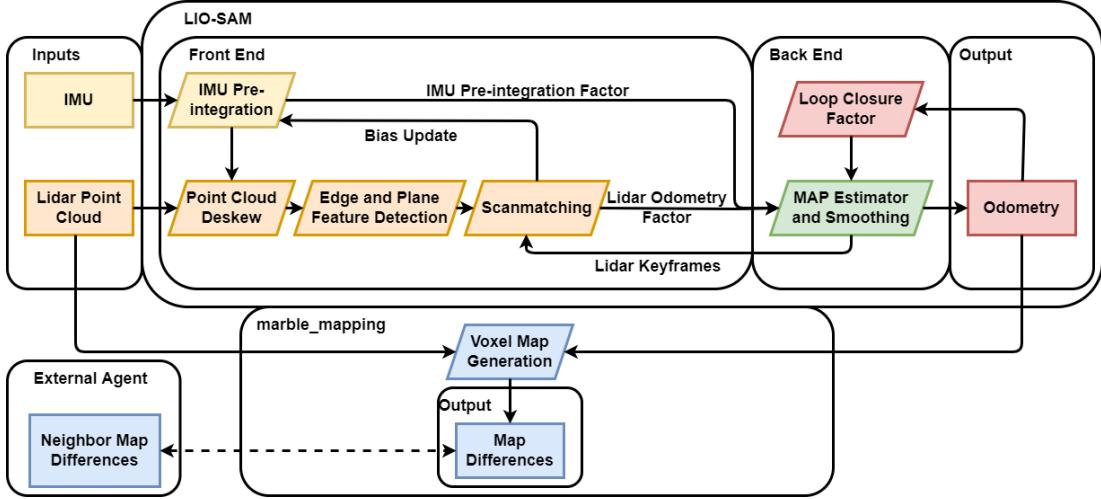


Figure 7: Overview of team MARBLE’s SLAM architecture.

its own identifier and that of their neighbors. Then when two robots connect to each other or the base station via a deployed mesh network, they request any differences not contained with their own maps, and pass on any differences they had generated to the neighboring agent. To minimize overhead, maps were transmitted in their binary state, after thresholding the occupancy probability to an occupied/unoccupied state.

As each robot only optimized its own trajectory and map, any significant drift or misalignment between robots could cause potential downstream issues with multi-agent planning algorithms. To mitigate this issue, each robot prioritizes its own map, specifically during the merging process, where free voxels in the parent robot were kept free and the occupied voxels were merged together. The base station operator also had the ability to remove or stop merging differences from specific agents if significant tracking errors occurred.

**Dirty Details. Parameter tuning and IMU:** The IMU covariance was found to have a substantial impact on the roll, pitch, and yaw estimation. In constrained passage ways, rotation accuracy significantly decreased as a result of a significant number of LIDAR points falling below a minimum range threshold. Relying more heavily on the IMU during these maneuvers improved rotation accuracy substantially (although it did not fully eliminate the problem). In this regard, using a good IMU is paramount: the LORD Microstrain 3DM-GX5-15, provides exceptionally high accuracy pitch and roll estimates of  $0.4^\circ$ , along with a  $0.3^\circ/\sqrt{\text{hr}}$  gyro estimate [101], which allowed the MARBLE system to rely on IMU-only measurements for extended periods of time.

A second key parameter in the system was the key-frame search radius for loop closures. Given the localization maintained qualitatively good accuracy, the Euclidean search distance was continually reduced, resulting in a final distance of 2 m for loop closure constraints. As loop closure optimization were computationally expensive, this saved on CPU cycles and additionally helped avoid spurious loop closure between different elevations of tunnels or floors in a building.

**Hardware design:** Team MARBLE relied on precision machining to obtain (and preserve) an accurate extrinsic calibration between LIDAR and IMU. Further calibration may

have benefited the final solution —specifically, improved IMU noise and bias estimation. It was found that certain IMUs did not perform as well as others in qualitative analysis of two robots traversing roughly the same trajectories. The team opted to swap hardware over further exploration of the cause of these errors. The chosen hardware likely had the closest noise parameters to those provided by the IMU manufacturer.

**LIO-SAM enhancements:** Team MARBLE also made two minor adjustments to LIO-SAM. During initialization, the team chose to ignore measurements from the IMU until a point cloud had been received, since the IMU was not a full AHRS unit and did not have a heading compass. The second adjustment was to the IMU timestamps, prior to integration. As a result of the (ACM-based) USB driver used by the IMU, the measurements did not have guaranteed priority on the kernel. This meant the timestamps generated by the system were not always consistent, which often caused negative timestamp values in the IMU pre-integration, leading to instabilities. To avoid this issue, the MARBLE implementation replaced the timestamps (using the nominal IMU frequency) if they were outside an acceptable range. While this method was less precise than a full hardware clock sync, it was fairly easy to implement given the available onboard connections (a full hardware sync would have required an extra RS232 port). In practice, we noticed that the back-end optimizer was able to mitigate the impact of minor timestamp mismatches.

#### H. Common Themes on the Path to Robustness

Despite the unique features that distinguish the architectures adopted by the SubT teams, the previous sections reveal a substantial convergence of technical approaches across teams. This convergence is a testament of the maturity of multi-robot LIDAR-centric SLAM, at least for small robot teams, (*e.g.*, 5–10 robots). We discuss commonalities across systems below.

**Sensing.** Most teams relied on LIDAR and IMU as the dominant sensing modalities; IMUs are not sensitive to perceptual aliasing (*i.e.*, the case where different places have the same appearance/sensor footprint) and environmental disturbances; LIDARs afford accurate and long-range depth measurements

even in the absence of external illumination. At the same time, visual, thermal, and wheel/kinematic odometry remain an important addition to LIDAR, especially in the presence of obscurants and to increase redundancy. Many teams (*e.g.*, CSIRO, Explorer, MARBLE, CoSTAR) adopted a common sensor payload to be mounted on the different robots. This modular design allows standardizing calibration and testing procedures, and partially decouples the development of the SLAM system from other hardware choices.

**SLAM Front-end and Back-end.** All teams relied on local (single-robot) front-ends to pre-process the LIDAR data. Such pre-processing reduces the data volume communicated to the base station or to the other robots. Moreover, it allows splitting computation across the robots, improving scalability. Most solutions perform extensive point cloud pre-processing, including de-skewing and voxel grid filtering. The front-ends then process the LIDAR scans using feature-based (akin to LOAM [65]) or dense (*e.g.*, ICP-based) matching. Regarding the SLAM back-end, virtually all teams relied on factor graph or pose graph optimization (except for the Kalman-filter-based odometry from CTU-CRAS-Norlab). Several teams decided not to detect loop closures (*e.g.*, CTU-CRAS-Norlab, and partially CERBERUS), based on considerations about the scale of the environment and the computational constraints at the robots. Finally, most teams built on top of open-source libraries for the LIDAR front-end and back-end, including GTSAM [67], maplab [69], LOAM [65], LIO-SAM [98], Octomap [100], and libpointmatcher [102].

**Loosely-coupled vs. Tightly-coupled Architectures.** Most teams resorted to loosely-coupled sensor fusion techniques, where estimates from multiple sensors are first fused into pose estimates and then combined together. Loosely-coupled approaches enable a more modular software design and make the implementation of health checks for each data source and intermediate pose estimate easier. This has been shown to largely increase robustness to hardware and software failures, *e.g.*, [30], [46], [94]. In addition, tightly-coupled fusion leads to larger optimization problems, which prevents scaling the multi-robot back-ends to large teams.

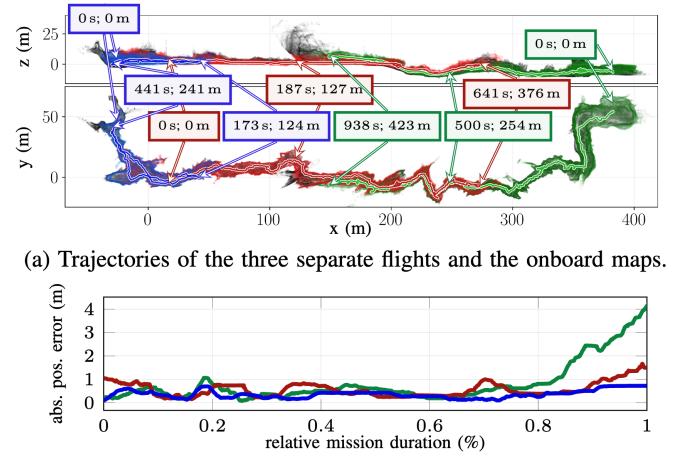
**Centralized and Decentralized Architectures.** CERBERUS and CoSTAR adopted centralized architectures, where the base station performs a joint optimization over the entire robot team. All the other teams adopted a decentralized

approach, where each robot mostly operated on its own, with the occasional exchange of the mapping results (see CTU-CRAS-Norlab, Explorer, and MARBLE) or with a multi-robot pose graph optimization executed at each robot (CSIRO). No team adopted a distributed architecture, which are still the subject of active research [28] and were less amenable to the rules of the SubT competition, which required collecting data at a base station for visualization and scoring purposes.

#### IV. STATE OF PRACTICE AND MATURITY OF UNDERGROUND SLAM

The previous section discussed state-of-the-art approaches for SLAM in underground environments across six SubT teams. This section reports on the practical performance that can be achieved by these approaches, which provides useful data points to assess the maturity of LIDAR-centric SLAM in underground worlds. We focus on three dimensions — odometry, loop closures, and multi-robot mapping— and for each we discuss performance and key aspects impacting it.

**Odometry Estimation Accuracy.** This section shows that modern LIDAR-centric odometry estimators can achieve a



(a) Trajectories of the three separate flights and the onboard maps.  
(b) Localization accuracy during the three exploration deployments colored with respect to (a).

Figure 8: CTU-CRAS-Norlab’s odometry accuracy for three single-UAV deployments in the Bull Rock Cave system [103]

Table I: Open-source Datasets and Code Released by the SubT teams

Team	Code	Dataset
CERBERUS	<a href="https://www.subt-cerberus.org/code--data.html">https://www.subt-cerberus.org/code--data.html</a>	
CoSTAR	<a href="https://github.com/NeBula-Autonomy">https://github.com/NeBula-Autonomy</a>	
CTU-CRAS-Norlab	<a href="https://github.com/ctu-mrs/aloam">https://github.com/ctu-mrs/aloam</a> <a href="https://github.com/ctu-mrs/octomap_mapping_planning">https://github.com/ctu-mrs/octomap_mapping_planning</a> <a href="https://github.com/ctu-mrs/mrs_uav_system">https://github.com/ctu-mrs/mrs_uav_system</a> <a href="https://github.com/norlab-ulaval/norlab_icp_mapper">https://github.com/norlab-ulaval/norlab_icp_mapper</a> <a href="https://github.com/norlab-ulaval/norlab_icp_mapper_ros">https://github.com/norlab-ulaval/norlab_icp_mapper_ros</a> <a href="https://github.com/ethz-asl/libpointmatcher">https://github.com/ethz-asl/libpointmatcher</a> <a href="https://github.com/norlab-ulaval/libpointmatcher_ros">https://github.com/norlab-ulaval/libpointmatcher_ros</a> <a href="https://github.com/ethz-asl/libnabo">https://github.com/ethz-asl/libnabo</a>	<a href="https://github.com/ctu-mrs/slam_datasets">https://github.com/ctu-mrs/slam_datasets</a>
Explorer	<a href="https://www.superodometry.com/">https://www.superodometry.com/</a> <a href="https://theairlab.org/research/2022/05/02/subt_code/">https://theairlab.org/research/2022/05/02/subt_code/</a>	<a href="https://theairlab.org/dataset/interestingness">https://theairlab.org/dataset/interestingness</a> <a href="https://www.superodometry.com/datasets">https://www.superodometry.com/datasets</a>
MARBLE	<a href="https://github.com/arpq/marble_mapping">https://github.com/arpq/marble_mapping</a> <a href="https://github.com/arpq/LIO-SAM">https://github.com/arpq/LIO-SAM</a>	<a href="https://arpq.github.io/coloradar">https://arpq.github.io/coloradar</a> <a href="https://arpq.github.io/oivio">https://arpq.github.io/oivio</a>

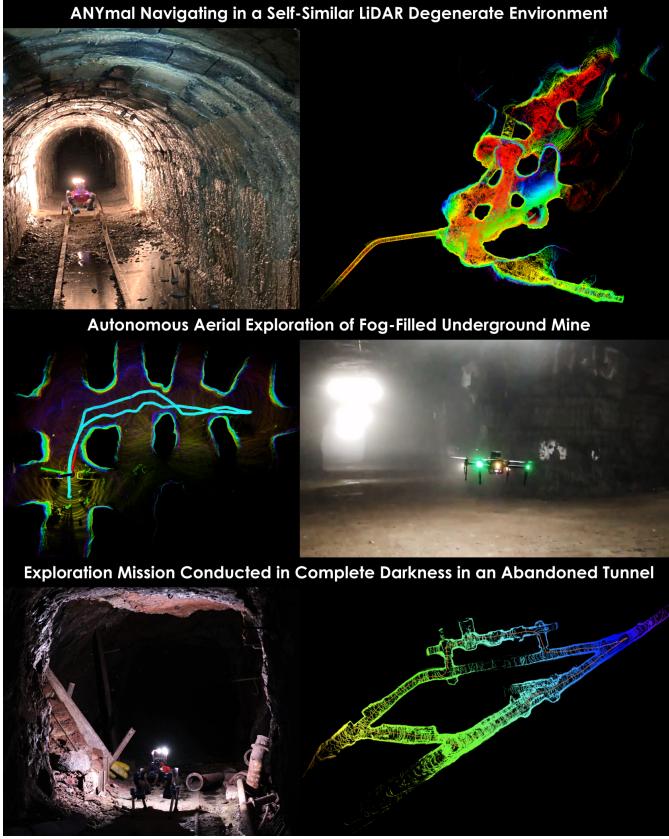


Figure 9: (Top) Autonomous exploration of a self-similar environment making LIDAR only localization unreliable. (Middle) Aerial exploration in a fog-filled environment using CompSLAM and exploiting thermal vision. (Bottom) Underground tunnel environment exploration in conditions of darkness and subject to reflections due to mud/water puddles.

very low-drift (0.1-0.5 % of the trajectory traveled) in challenging underground environments. This enables impressive localization performance over long distances. For instance, Figure 8 shows results from team CTU-CRAS-Norlab’s unmanned aerial vehicles, achieving localization error under 1 m in the Bull Rock cave system with flights reaching trajectory lengths of 600 m and maximum velocities up to  $2 \text{ m s}^{-1}$ .

**Multi-modality:** Multi-modal sensing enhances robustness in challenging environmental conditions (*e.g.*, darkness, fog, smoke, dust, or feature-less scenes), as well as in the presence of hardware and software failures. Figure 9 shows the map obtained by the multi-modal, onboard CompSLAM approach by team CERBERUS; CompSLAM achieves a low-drift trajectory estimate in extreme conditions with significant dust and obscurants. Although primarily driven by LIDAR, CompSLAM also uses other modalities (*e.g.*, kinematic odometry or thermal) that are less sensitive to dense obscurants. This is still achieved on a modest computational budget: CompSLAM has been deployed on both ANYmal C robots that are equipped with powerful processors (i7-class systems), and on the RMF-Owl aerial robot [104], that relies on a single-board computer.

**LIDAR pre-processing:** LIDAR data pre-processing is a key ingredient for accurate odometry estimation. Figure 10 shows an ablation study conducted by team CTU-CRAS-NORLAB

on an unmanned ground vehicle, which highlights the impact of de-skewing the LIDAR scans, as well as the impact of constraining the roll and pitch of the platform using IMU data during ICP (see Section III-E). The path consists of a robot traveling through an unknown environment up to 150 m (the “exploration” phase), to which point it turned around to come back to the base station (the “exploitation” phase). Although CTU-CRAS-Norlab’s SLAM solution does not use loop closures, it assumes low odometry drift and can reuse its global map for scan-to-map matching when revisiting known environments. All curves in Figure 10(a) exhibit increasing errors (drift) during exploration but the de-skewing and roll-and-pitch-constrained optimization lead to reduced errors. The result is confirmed by the localization error box plots in Figure 10(b). LIDAR pre-processing (*e.g.*, point down sampling via voxel grid filtering) is also crucial to reduce the computational burden, see the analysis in [74].

**Importance of Loop Closures.** While LIDAR-centric solutions compute low-drift odometric trajectories, such trajectory estimates keep accumulating error over time. With a 0.5% odometry drift, a robot would have 5 m error after 1 km traverse. This stresses the importance of detecting and enforcing loop closures to keep the localization error *bounded*. Figure 11 provides an example of accurate localization and mapping results by team Explorer, achieved by successful detection of loop closures. The figure shows mapping results in Brady’s Bend cave near Pittsburgh, PA, on a wheeled ground robot. According to DARPA, team Explorer’s SLAM system achieved a deviation of 6% in the grand finale of the DARPA SubT challenge, a performance that is the second best among all competing teams, behind CSIRO’s Wildcat.<sup>3</sup>

**Robustness to outliers:** LIDAR-based loop closure detection is quite challenging in underground scenarios due to perceptual aliasing. At larger scales, and with more robots, the chances of false positive loop closures increase, especially in environments with self-similar locations. False loop closures, if not rejected, can have a negative impact on localization

<sup>3</sup>Team Explorer’s performance was also recognized with the “Most Sectors Explored Award” by DARPA.

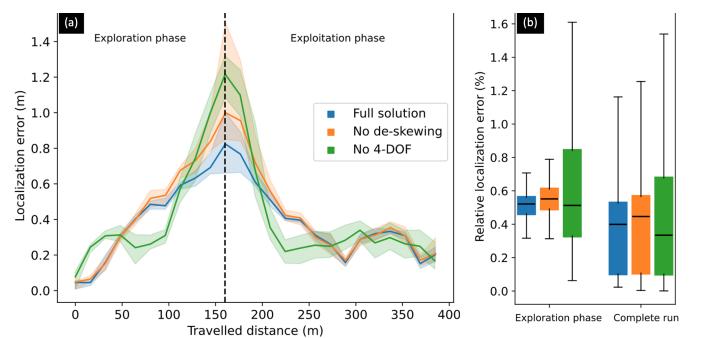


Figure 10: Localization error as a function of distance traveled. The solid lines are the median error, and the colored areas represent the first and third error quartiles. The dashed line delimits the exploration phase, during which the robot explored new areas, before returning to previously visited areas. Statistics are computed over ten experiments.

performance and lead to dramatic distortions in the map. Figure 12 shows CoSTAR’s SLAM results (a) without and (b) with outlier rejection. CoSTAR’s GNC-enabled [55] approach has been shown to produce accurate maps and reject up to 90% outlier loop closures during the Final event of the DARPA SubT challenge [105]. As depicted in Figure 13, CoSTAR’s outlier-robust loop closure detection enables creating high-precision 3D maps from multi-level urban environments with a combination of large rooms and small spaces, to complex weaving lava tubes, to mines that are massive in scale, and finally the narrow passages found in the SubT Final event.

*Heterogeneous environments:* Other examples of high precision localization and mapping in a large-scale and long-duration exploration are shown in Figure 14 and Figure 15 for the LIO-SAM system adopted by team MARBLE. In these experiments, a robot is teleoped from within the University of Colorado-Boulder Engineering Center through all three levels of a parking garage before returning to its approximate original location in an hour-long operation. The test spans

heterogeneous environment types, from tight urban indoor environments (with sharp turns, feature-less and narrow corridors, and staircases) to wide-open outdoor environments. The SLAM system accurately maintains elevation estimation through multiple levels of the parking garage with high level of geometric self-similarity while relying on only the OS1 LIDAR and IMU. The 2.2 km long trajectory shows a position difference of 0.31 m from the start to the final position, which is equivalent to an error (after loop closures) of just 0.014%.

**Importance of Multi-robot Operation.** Multi-robot SLAM allows mapping larger areas while simultaneously reducing the localization and mapping errors thanks to inter-robot loop closures. Figure 16 shows the maps produced by CSIRO’s Wildcat decentralized multi-robot SLAM system in two SubT events (Urban and Final) and in a cave in Australia. The map in Figure 16(a) is built by three ground robots, while the maps in Figure 16(b-c) are created by four robots (including a UAV, in the cave case). According to DARPA, in the Final SubT event Wildcat produced the top map with less than 1%

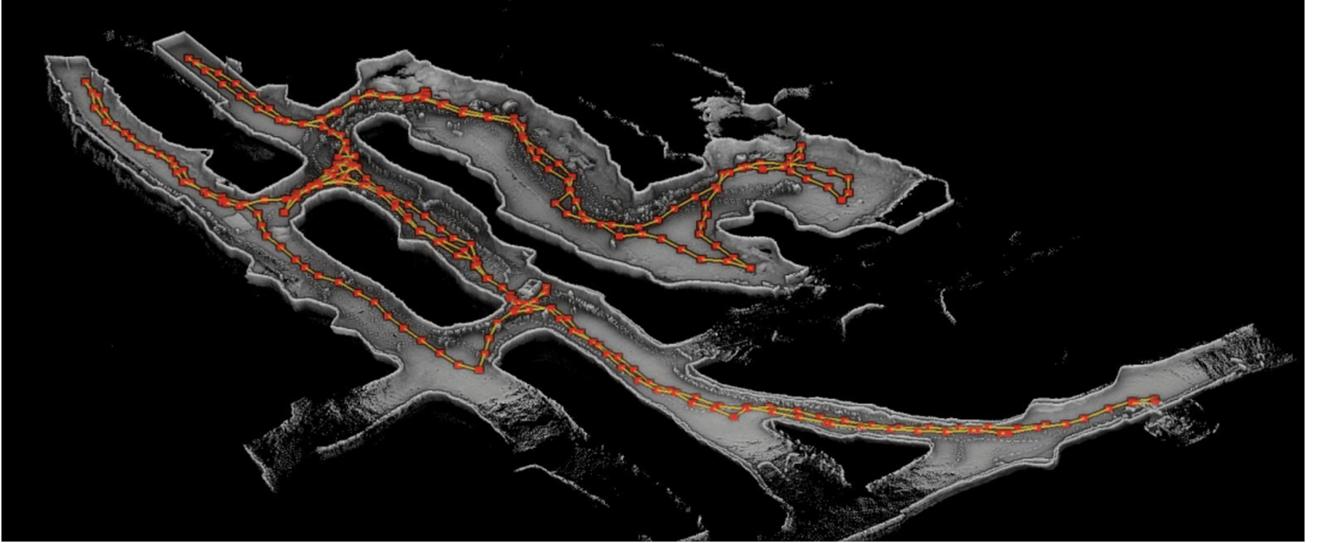


Figure 11: Team Explorer mapping results in Brady’s Bend cave near Pittsburgh, PA, on a wheeled ground robot. The red dots represent the key poses and yellow edges show potential loop closure edges.

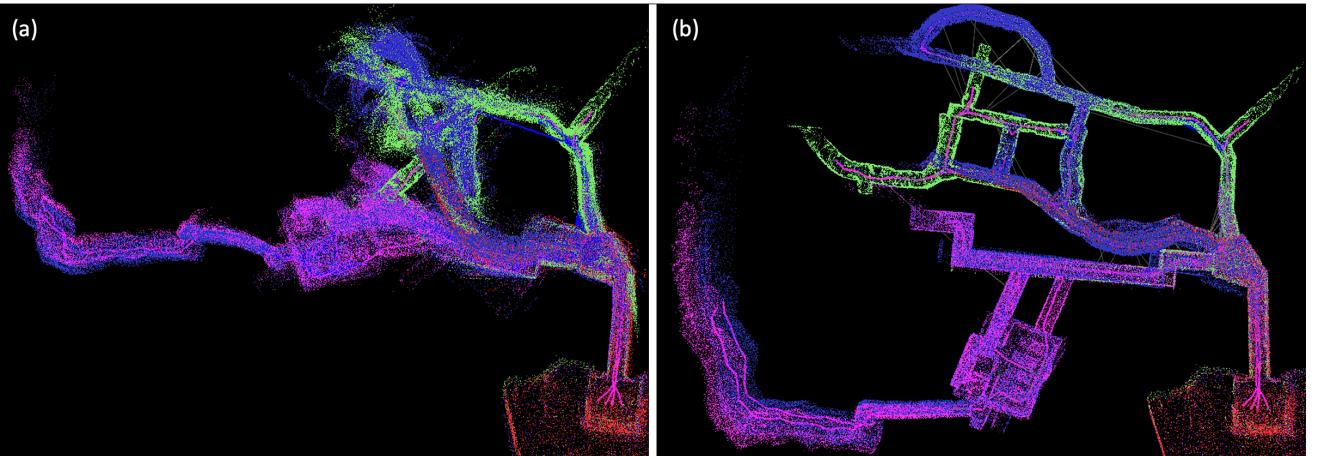


Figure 12: CoSTAR’s SLAM results (a) without and (b) with the outlier rejection module during the preliminary run of the SubT Finals. Each color represents the map of a different robot (four in total), blue lines represent accepted loop closures and gray lines represent rejected loop closures. GNC is able to successfully reject numerous false loop closures (gray lines in (b)).

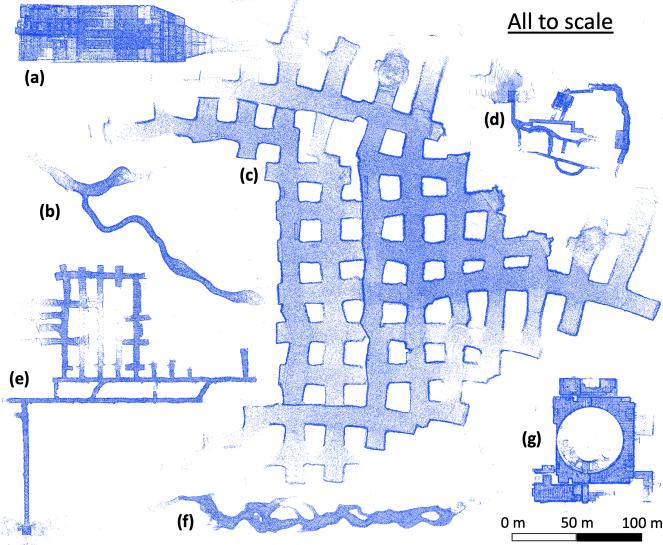


Figure 13: A to-scale representation of the robot-produced maps across a variety of environments team CoSTAR tested in. (a) a 3-level abandoned subway in Los Angeles. (b) A lava tube in Lava Beds National Monument. (c) part of Kentucky Underground. (d) DARPA-created SubT Finals course. (e) Bruceton Research Mine (SubT Tunnel Competition). (f) Valentine cave (a lava tube) in Lava Beds National Monument. (g) Satsop power plant (SubT Urban Competition). All maps are the best runs from a single-robot.

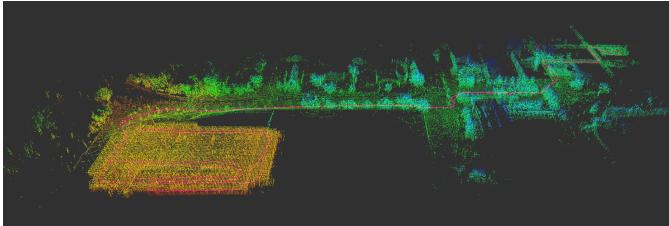


Figure 14: LIO-SAM map generated by MARBLE’s Spot robot traversing from the University of Colorado-Boulder Engineering Center to the bottom of a nearby parking garage and back. The trajectory is marked in pink, and the point cloud is colored by elevation from red low to blue high.

deviation from the ground truth where they defined deviation as the percentage of points in the submitted point cloud that are farther than one meter from the points in the surveyed point cloud map. Wildcat also produced the single most accurate reports in the Urban and Final events with 22 cm and 4.8 cm error, respectively. We refer the reader to [84] for a more extensive experimental evaluation. Additional qualitative results produced by Wildcat in perceptually challenging environments are also available on the websites of two commercial partners of CSIRO, Emesent [86] and Automap [106].

*Inter-robot loop closures:* Figure 17 shows the dramatic reduction of the Absolute Pose Error (APE) in team CoSTAR’s SLAM architecture due to inter-robot loop closures and multi-robot pose graph optimization. As in the single-robot case, capitalizing on inter-robot loop closures requires a good strategy for outlier rejection, since many inter-robot loop closure detections will be incorrect due to perceptual aliasing.

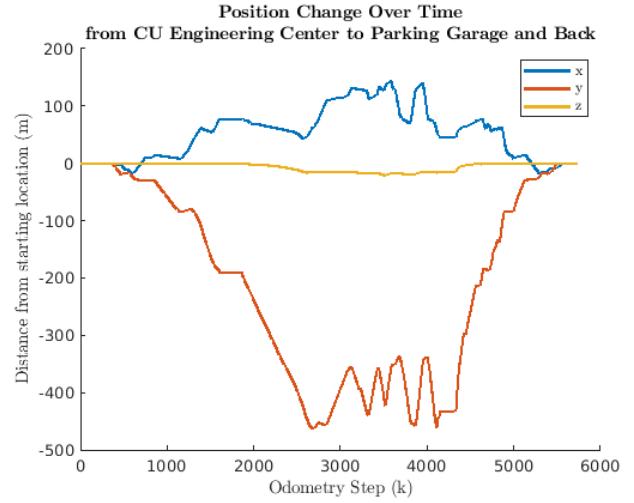


Figure 15: Position  $x$ ,  $y$ ,  $z$  trajectory data of path in Figure 14. The final position offset from the initial starting location was, 0.31 m with a total trajectory length of 2.2 km.

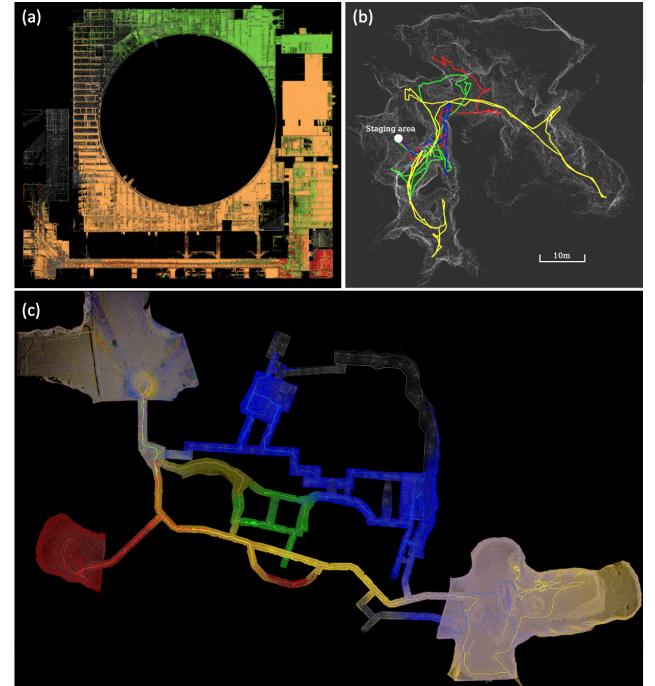


Figure 16: CSIRO’s Wildcat results: (a) Point cloud map produced by three ground robots in the Beta Course of the Urban Event; estimated map is color-coded by agent, while DARPA reference point cloud is shown in gray. (b) Map produced during cave testing at Capentaria Caves, Chillagoe, Queensland; the merged agent-collected map is shown in gray, with agent trajectories in colors (drone colored as yellow). (c) Merged point cloud map from the Final Event, color-coded by agent, with DARPA reference cloud in gray.

In the DARPA SubT finals event, team CERBERUS deployed four ANYmal quadrupedal robots to autonomously navigate a total distance of 1.75 km. The maps generated by the onboard solution, CompSLAM, along with scoring artifacts, are qualitatively compared against the DARPA-provided ground truth map in Figure 18. The individual robot results are

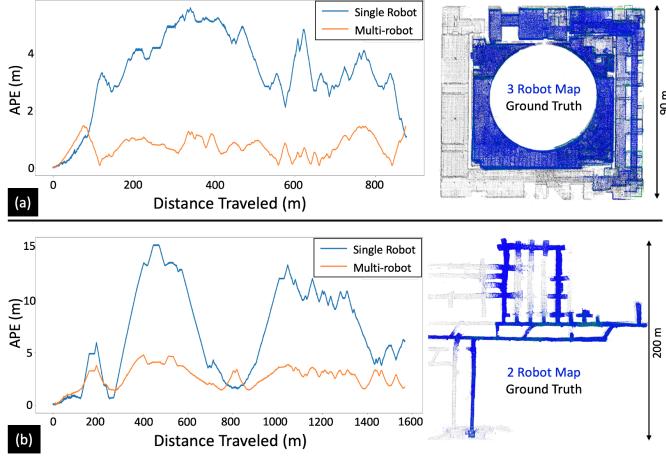


Figure 17: CoSTAR results: improvement in Absolute Pose Error (APE) due to inter-robot loop closures and multi-robot pose graph optimization, with the resulting multi-robot map on the right. (a) Test with three robots in the Satsop power plant. (b) Test with two robots in the Bruceton Research Mine.

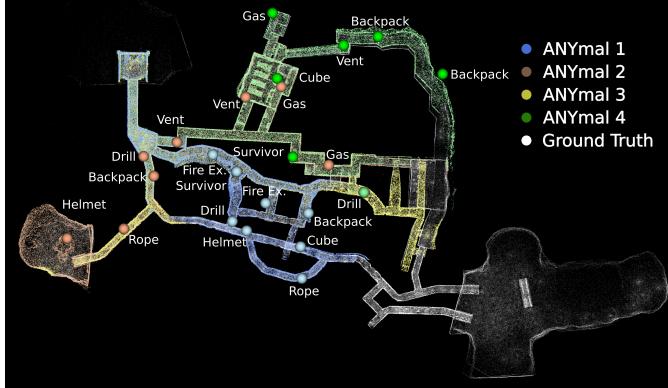


Figure 18: CERBERUS results: Onboard maps for all deployed robots generated using CompSLAM are overlaid on top of the DARPA provided ground truth for the final event of the DARPA SubT Challenge. The scored artifacts are shown and are colored to correspond with the reporting robot.

made globally consistent by M3RM by exploiting inter-robot loop closures; a quantitative comparison between the onboard and global mapping approaches is presented in Table II.

Robot	CompSLAM (Onboard)		M3RM (Server)	
	Rotation [°]	Translation [m]	Rotation [°]	Translation [m]
ANYmal 1	2.45 (0.67)	0.72 (0.41)	<b>1.59</b> (0.46)	<b>0.25</b> (0.13)
ANYmal 2	3.97 (0.40)	1.29 (0.90)	<b>0.96</b> (0.30)	<b>0.36</b> (0.28)
ANYmal 3	<b>0.89</b> (0.50)	0.23 (0.43)	2.30 (1.02)	<b>0.20</b> (0.34)
ANYmal 4	2.22 (0.79)	1.00 (0.71)	<b>2.16</b> (0.55)	<b>0.24</b> (0.17)

Table II: Comparison of the mean and standard deviation of the Absolute Pose Error (APE) for CERBERUS’ CompSLAM (each robot) and M3RM (all robots considered together) approaches for the DARPA SubT challenge Final Event.

*Heterogeneous teams:* We already commented on the benefit of having heterogeneous sensing capabilities. Here, we discuss the advantage of using heterogeneous platforms for exploration. Indeed, most of the SubT teams used a combination

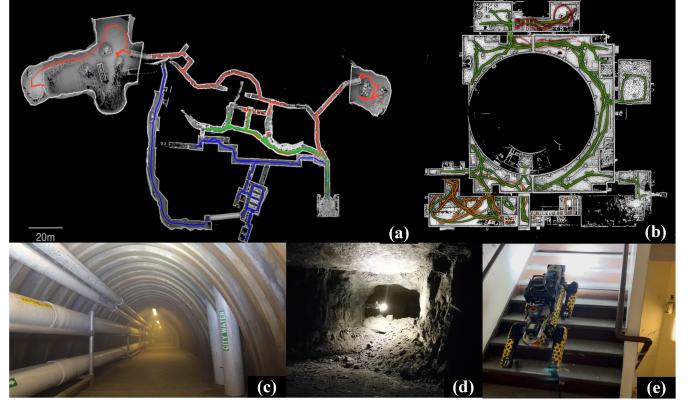


Figure 19: Testing sites and results of Explorer’s SLAM system. (a) UAV, UGV, and legged robot exploration in the Final Circuit of SubT, including urban, cave, and tunnel environments. (b) UAV and UGV exploration in the Urban Circuit of SubT. (c) Tunnel environments, test site with smoke. (d) Cave environment UAV test site. (e) Urban environment with legged robot.

of wheeled and legged ground robots and UAVs. Figure 19(c) shows Explorer’s mapping result in the Urban Challenge Alpha Course reconstructed by multiple robots (UGV1, UAV1, and UAV2) operating in a dark and foggy environment with a vertical shaft. Green, orange, and red lines are the estimated trajectories of UGV1, UAV1, and UAV2 respectively. Figure 19(a) shows the mapping result in the SubT Finals by a heterogeneous fleet. The blue, green and red lines are the estimated trajectories of UGV1, UGV2, and UGV3 respectively. The travel distance of UGV1, UGV2, and UGV3 are 445.2 m, 499.8 m, and 596.6 m, respectively. Explorer’s SLAM solution achieved accurate localization and mapping despite the challenging environmental conditions, including low light, long corridors, heavy dust/fog, and even dynamic scenes. Heterogeneity enables mapping a broader variety of environments (*e.g.*, UAVs enable exploring vertical shafts) and allows richer exploration strategies (*e.g.*, using UAVs for fast exploration, and UGVs for more accurate mapping).

## V. FUTURE RESEARCH DIRECTIONS AND OPEN PROBLEMS

In the light of the results in Section IV and the outcome of the DARPA SubT competition, this section provides a summary of which problems in underground SLAM can be considered solved or can be solved with some good engineering and what are still open problems that will likely require more fundamental research.

LIDAR-centric SLAM solutions have become increasingly robust to challenging environments. Feature detection or scan pre-processing enable real-time point cloud alignment. Tight coupling with inertial data enables more robust motion estimation, by allowing de-skewing the LIDAR scans, bootstrapping ICP-based scan matching, and potentially eliminating roll and pitch drift. Keyframe-based or sub-map-based approaches, combined with a factor graph framework, allow sparsifying the trajectory into a reduced set of poses and enable online

operation in large-scale, long-term, multi-robot explorations with reduced computational complexity. The addition of other sensing modalities further increases robustness.

Looking across the six solutions examined in Section III-IV, there is reason to believe that the underground SLAM problem, with high-quality multi-modal sensing suites, is a solved problem. Yet only solved with sufficient qualifications of the environment, the scale, the sensors, the parameter tuning, and the computation power. We believe that in the context of extreme subterranean environments, the majority of open problems defined in [7] still applies. In the rest of this section, we highlight current challenges and open problems in underground localization and mapping.

**Robust and Resilient Perception.** One of the common failure modes observed across most of the presented architectures is localization failure due to falls, drops, or collisions [107] when traversing rough terrains in unstructured underground environments. These high-frequency motions are not entirely captured by the onboard perception system, *e.g.*, due to the lower sampling frequency of structured-light sensors [108]. This could lead to poor motion estimates and eventually localization failure. With robotic systems that can withstand a fall and continue to operate (*e.g.*, Boston Dynamics Spot, Flyability drones, BIA5 Titan, ANYmal, RMF-Owl), a relatively under-explored area is reliable state estimation under unexpected collisions and temporary interruptions of the sensor streams. Although early work on localization subject to collision shows promising results [109], better exploring the limitations of different systems and algorithms in “crash tests” scenarios would help improve all-round real-world robustness. Furthermore, engineering work in incorporating velocity-based sensors (*e.g.*, event-based cameras [110]) which might maintain ego-motion tracking without saturation during adverse events could greatly benefit SLAM systems.

At a more fundamental level, underground operation requires redundancy and resourcefulness, but this needs to be achieved beyond just “adding more sensors”. The SLAM literature is lacking fundamental research in *resilient* algorithms and systems. While robust systems are designed to withstand (often small) disturbances (*e.g.*, degraded sensing or environmental changes), resilient methods dynamically reconfigure to regain performance in the face of changing environmental stressors [19]. For instance, a resilient system would dynamically change its parameters (or even its algorithmic components) depending on the scenario, contrarily to the current SLAM systems, which are “rigid” and heavily rely on manual parameter tuning; see the comments about parameter tuning in the dirty details subsections in Section III, as well as the discussion about the “curse of parameter tuning” in [7].

**Beyond Traditional SLAM Sensors.** Achieving robustness under perceptual aliasing, dense obscurants, and severe environment degradation remains a challenge and can benefit from incorporating non-traditional sensing modalities and designing methods for failure detection and recovery. Thermal vision allows penetrating conditions of visual degradation, where cameras and LIDARs fail due to the presence of obscurants. Similarly, radar is able to maintain localization despite the presence of fog, as the wavelengths in commercial automotive

millimeter-wave radars are large enough to bypass particulate such as fog and dust that causes spurious reflections that render LIDAR point clouds unusable for localization and mapping purposes. While research into millimeter-wave radar-based localization [111]–[114] and the creation of radar factors for SLAM applications is ongoing, including the release of public datasets such as [115], [116], the integration of these sensors is not as established as other sensing modalities, due to complexity of the corresponding sensor models and data association. Multi-modal SLAM systems could also be pushed further by developing failure detection and recovery methods. Autonomous exploration of subterranean settings requires dynamically adaptive algorithmic architectures to achieve solution resourcefulness. Still related to resilient operation, it would be desirable to design approaches that can detect failures of a sensing modality and reconfigure the system accordingly. The importance of degeneracy detection in multi-modal sensing is discussed in [31], while fault detection in perception system is investigated in [117].

**Scaling Up: Centralized vs. Distributed Systems.** Multi-robot LIDAR-centric SLAM is a mature research area. This paper showed that centralized approaches can achieve accurate and real-time performance for moderate team sizes (5-10 robots); moreover, decentralized approaches attain small errors even without relying on inter-robot loop closures in moderate-scale scenarios (*e.g.*, <1 km traversal). However, scaling up SLAM solutions to very large teams (*e.g.*, >100 robots) and very large-scale scenarios (*e.g.*, city-scale [9] and forest-scale [118]) is likely to require a more distributed approach. In centralized approaches, large team sizes would quickly reach a bottleneck in terms of communication as well as processing at the base station.<sup>4</sup> Therefore, distributed architectures are likely to be needed to scale up operation. For large fleets covering large-scale geographic areas, it will be necessary to consider (i) resource-aware collaborative inter-robot loop closure detection techniques [57]–[59] that intelligently utilize limited mission-critical resources available onboard (*e.g.*, compute, battery, and bandwidth) and (ii) distributed factor-graph and pose-graph optimization methods [27], [28], [60], [119], both of which are active research areas. We also believe that hierarchical map representations (*e.g.*, [120]) will be needed for large-scale environments where point-cloud or voxel-based representations would clash with memory constraints.

In terms of engineering, it would be desirable to develop and release open-source implementations of multi-robot SLAM systems. As we observed, SLAM progress in SubT was also enabled by the availability of high-quality open-source implementation for SLAM components (*e.g.*, the back-end provided by GTSAM) or entire systems (*e.g.*, LIO-SAM). Therefore, the development of distributed SLAM systems will benefit from a similar open-source infrastructure.

**Scaling Down: Miniaturization and Low-Cost Sensing.** All solutions examined in this paper leverage one or multiple

<sup>4</sup>The SubT teams carefully handled communication (*e.g.*, via compression and down-sampling) to meet the bandwidth constraints. Moreover, teams using centralized solutions relied on powerful base stations, *e.g.*, CoSTAR’s base station relied on an AMD Ryzen Threadripper 3990X with 64 cores/128 threads at 2.9GHz.

LIDARs, and powerful embedded computers. More work is required to enable the capabilities presented in this paper, but with low-cost components that might be suitable for smaller, cheaper, expendable systems. For instance, it would be desirable to deploy a large number of expendable robots for high-risk missions (*e.g.*, search & rescue, planetary exploration), or to design more affordable robots to increase adoption by first responders. These platforms would ideally have a small form factor to enable exploration of narrow passages (*e.g.*, pipes) while being easy to transport by human operators. Achieving this goal entails both engineering efforts (*e.g.*, development of novel sensors or specialized ASICs for on-chip SLAM [8]) and more research on vision-based SLAM in degraded perceptual conditions (*e.g.*, dust or fog).

## VI. CONCLUSION

While progress in SLAM research has been reviewed in prior works, none of the previous surveys focus on underground SLAM. Given the astonishing progress over the past several years, this paper provided a survey of the state-of-the-art, and the state-of-the-practice in SLAM in extreme subterranean environments and reports on what can be considered solved problems, what can be solved with some good systems engineering, and what is yet to be solved and likely requires further research. We reviewed algorithms, architectures, and systems adopted by six teams that participated in the DARPA Subterranean (SubT) Challenge, with particular emphasis on LIDAR-centric SLAM solutions, heterogeneous multi-robot operation (including both aerial and ground robots), and real-world underground operation (from the presence of obscurants to the need to handle tight computational constraints). Furthermore, we provided a table of open-source SLAM implementations and datasets that have been produced during the SubT challenge and related efforts, and constitute a useful resource for researchers and practitioners.

## REFERENCES

- [1] J. Forlizzi and C. DiSalvo, “Service robots in the domestic environment: a study of the roomba vacuum in the home.” pp. 258–265, 2006.
- [2] “Amazon. Introducing Amazon Astro – household robot for home monitoring, with Alexa,” 2021.
- [3] M. Keynes and G. Blackman, “Vision cleans up: Dyson robot vacuum navigates with imaging.” *Imaging and Machine Vision Europe*, vol. 81, pp. 28–29, 2017.
- [4] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous localization and mapping: A survey of current trends in autonomous driving,” *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.
- [5] S. Karthika, P. Praveena, and M. GokilaMani, “Hololens,” *International Journal of Computer Science and Mobile Computing*, vol. 6, no. 2, pp. 41–50, 2017.
- [6] P. R. Desai, P. N. Desai, K. D. Ajmera, and K. Mehta, “A review paper on oculus rift-a virtual reality headset.” *arXiv preprint arXiv:1408.1173*, pp. 41–50, 2014.
- [7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [8] A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, and V. Sze, “Navion: A 2-mw fully integrated real-time visual-inertial odometry accelerator for autonomous navigation of nano drones.” *IEEE Journal of Solid-State Circuits* 54, vol. 4, pp. 1106–1119, 2019.
- [9] Max Yin and Sebastian Scherer, “Automerge: A framework for map assembling and smoothing in city-scale environments,” [https://github.com/MetaSLAM/AutoMerge\\_Server](https://github.com/MetaSLAM/AutoMerge_Server), online; accessed 9 May 2022.
- [10] DARPA, “DARPA Subterranean Challenge,” <https://www.subchallenge.com>, 2018, online; accessed 8 March 2022.
- [11] H. F. Durrant-Whyte and T. Bailey, “Simultaneous localisation and mapping (SLAM): Part I,” *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, 2006.
- [12] T. Bailey and H. F. Durrant-Whyte, “Simultaneous localisation and mapping (SLAM): Part II,” *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, 2006.
- [13] M. Kegeleirs, G. Grisetti, and M. Birattari, “Swarm SLAM: Challenges and perspectives,” *Frontiers in Robotics and AI*, vol. 8, p. 23, 2021.
- [14] M. Dorigo, G. Theraulaz, and V. Trianni, “Swarm robotics: Past, present, and future [point of view],” *Proceedings of the IEEE*, vol. 109, pp. 1152–1165, 2021.
- [15] T. Halsted, O. Shorinwa, J. Yu, and M. Schwager, “A survey of distributed optimization methods for multi-robot systems,” *ArXiv Preprint*, pp. 1335–1379, 2021.
- [16] L. E. Parker, D. Rus, and G. S. Sukhatme, “Multiple mobile robot systems,” *Springer handbook of robotics*, pp. 1335–1379, 2016.
- [17] P. Y. Lajoie, B. Ramtoula, F. Wu, and G. Beltrame, “Towards collaborative simultaneous localization and mapping: a survey of the current research landscape.” *ArXiv Preprint: 2108.08325*, 2021.
- [18] L. Zhou and P. Tokekar, “Multi-robot coordination and planning in uncertain and adversarial environments,” *ArXiv Preprint: 2108.08325*, pp. 1–11, 2021.
- [19] A. Prorok, M. Malencia, L. Carlone, G. S. Sukhatme, B. M. Sadler, and V. Kumar, “Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems,” *CoRR*, vol. abs/2109.12343, 2021. [Online]. Available: <https://arxiv.org/abs/2109.12343>
- [20] S. Thrun, D. Hahnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker, “A system for volumetric robotic mapping of abandoned mines,” in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 3. IEEE, 2003, pp. 4270–4275.
- [21] A. Nuchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun, “6d slam with an application in autonomous mine mapping,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 2. IEEE, 2004, pp. 1998–2003.
- [22] D. Tardioli, D. Sicignano, L. Riazuelo, J. Villarroel, and L. Montano, “Robot teams for exploration in underground environments,” in *Workshop ROBOT11: Robtica Experimental*, 2012, pp. 205–212.
- [23] D. Tardioli, L. Riazuelo, T. Seco, J. Espelosín, J. Lalana, J. L. Villarroel, and L. Montano, “A robotized dumper for debris removal in tunnels under construction,” in *Iberian Robotics conference*. Springer, 2017, pp. 126–139.
- [24] R. Zlot and M. Bosse, “Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine,” in *Field and service robotics*. Springer, 2014, pp. 479–493.
- [25] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable slam system with full 3D motion estimation,” in *2011 IEEE international symposium on safety, security, and rescue robotics*. IEEE, 2011, pp. 155–160.
- [26] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, “Door-slam: Distributed, online, and outlier resilient slam for robotic teams.” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1656–1663, 2020.
- [27] Y. Chang, Y. Tian, J. P. How, and L. Carlone, “Kimera-multi: a system for distributed multi-robot metric-semantic simultaneous localization and mapping,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11210–11218.
- [28] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. How, and L. Carlone, “Kimera-Multi: Robust, distributed, dense metric-semantic slam for multi-robot systems,” *IEEE Transaction on Robotics*, 2021.
- [29] K. Ebadi, Y. Chang, M. Palieri, A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood *et al.*, “LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 80–86.
- [30] M. Palieri, B. Morrell, A. Thakur, K. Ebadi, J. Nash, A. Chatterjee, C. Kanellakis, L. Carlone, C. Guaragnella, and A.-a. Agha-Mohammadi, “Locus: A multi-sensor lidar-centric solution for high-precision odometry and 3D mapping in real-time,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 421–428, 2020.
- [31] K. Ebadi, M. Palieri, S. Wood, C. Padgett, and A. Agha-mohammadi, “DARE-SLAM: Degeneracy-aware and resilient loop closing in perceptually-degraded environments,” *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, pp. 1–25, 2021.

- [32] B. Morrell, M. Palieri, N. Funabiki, A. Thakur, J. G. Blank, and A. Aghamohammadi, "Robotic localization and multi-sensor, semantic 3D mapping for exploration of subsurface voids," in *AGU Fall Meeting Abstracts*, vol. 2020, 2020, pp. P057–03.
- [33] J. G. Blank, B. Morrell, A. Bouman, T. Touma, M. F. Ginting, C. Patterson, and A.-a. Agha-Mohammadi, "Autonomous Mapping and Characterization of Terrestrial Lava Caves Using Quadruped Robots: Preparing for a Mission to a Planetary Cave," in *Workshop on Terrestrial Analogs for Planetary Exploration*, ser. LPI Contributions, vol. 2595, Jun. 2021, p. 8122.
- [34] T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanský, D. Heřt, M. Petrlík, T. Báča, V. Spurný *et al.*, "DARPA subterranean challenge: Multi-robotic exploration of underground environments," in *International Conference on Modelling and Simulation for Autonomous Systems*. Springer, 2019, pp. 274–290.
- [35] M. Petrlík, T. Báča, D. Heřt, M. Vrba, T. Krajiník, and M. Saska, "A robust uav system for operations in a constrained environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2169–2176, 2020.
- [36] A. Bouman, M. F. Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick *et al.*, "Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2518–2525.
- [37] J. Williams, S. Jiang, M. O'Brien, G. Wagner, E. Hernandez, M. Cox, A. Pitt, R. Arkin, and N. Hudson, "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)*. IEEE, 2020, pp. 263–270.
- [38] A. Kramer, M. Kasper, and C. Heckman, "Vi-slam for subterranean environments," in *Field and Service Robotics*. Springer, 2021, pp. 159–172.
- [39] H. Azpúrrua, M. F. Campos, and D. G. Macharet, "Three-dimensional terrain aware autonomous exploration for subterranean and confined spaces," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2443–2449.
- [40] V. Krátký, P. Petráček, T. Báča, and M. Saska, "An autonomous unmanned aerial vehicle system for fast exploration of large complex indoor environments," *Journal of field robotics*, vol. 38, no. 8, pp. 1036–1058, 2021.
- [41] I. D. Miller, F. Cladera, A. Cowley, S. S. Shivakumar, E. S. Lee, L. Jarin-Lipschitz, A. Bhat, N. Rodrigues, A. Zhou, A. Cohen *et al.*, "Mine tunnel exploration using multiple quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2840–2847, 2020.
- [42] M. F. Ginting, K. Otsu, J. A. Edlund, J. Gao, and A.-A. Agha-Mohammadi, "Chord: Distributed data-sharing via hybrid ros 1 and 2 for multi-robot exploration of large-scale complex environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5064–5071, 2021.
- [43] H. Azpúrrua, A. Rezende, G. Potje, G. P. d. C. Júnior, R. Fernandes, V. Miranda, J. Domingues, F. Rocha, F. L. M. de Sousa, L. G. D. de Barros *et al.*, "Towards semi-autonomous robotic inspection and mapping in confined spaces with the espeleorobô," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 4, pp. 1–27, 2021.
- [44] T. Dang, F. Mascarich, S. Khattak, H. Nguyen, H. Nguyen, S. Hirsh, R. Reinhart, C. Papachristos, and K. Alexis, "Autonomous search for underground mine rescue using aerial robots," in *2020 IEEE Aerospace Conference*. IEEE, 2020, pp. 1–8.
- [45] D. Wisth, M. Camurri, S. Das, and M. Fallon, "Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1004–1011, 2021.
- [46] S. Khattak, H. Nguyen, F. Mascarich, T. Dang, and K. Alexis, "Complementary multi-modal sensor fusion for resilient robot pose estimation in subterranean environments," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1024–1029.
- [47] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *Ieee Access*, vol. 8, pp. 191 617–191 643, 2020.
- [48] J. Gross, M. De Petrillo, J. Beard, H. Nichols, T. Swiger, R. Watson, C. Kirk, C. Kilic, J. Hikes, E. Upton *et al.*, "Field-testing of a uav-ugv team for gnss-denied navigation in subterranean environments," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 2019, pp. 2112–2124.
- [49] M. T. Ohradzansky, E. R. Rush, D. G. Riley, A. B. Mills, S. Ahmad, S. McGuire, H. Biggie, K. Harlow, M. J. Miles, E. W. Frew *et al.*, "Multi-agent autonomy: Advancements and challenges in subterranean exploration," *arXiv preprint arXiv:2110.04390*, 2021.
- [50] J. Bayer and J. Faigl, "Speeded up elevation map for exploration of large-scale subterranean environments," in *International Conference on Modelling and Simulation for Autonomous Systems*. Springer, 2019, pp. 190–202.
- [51] B. Tidd, A. Cosgun, J. Leitner, and N. Hudson, "Passing through narrow gaps with deep reinforcement learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3492–3498.
- [52] B. Lindqvist, C. Kanellakis, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Compra: A compact reactive autonomy framework for subterranean may based search-and-rescue operations," *arXiv preprint arXiv:2108.13105*, 2021.
- [53] J. G. Rogers, J. M. Gregory, J. Fink, and E. Stump, "Test your slam! the subt-tunnel dataset and metric for mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 955–961.
- [54] J. Carter, J. Fink, and E. Stump, "The DARPA subt urban circuit mapping dataset and evaluation metric," in *Experimental Robotics: The 17th International Symposium*, vol. 19. Springer Nature, 2021, p. 391.
- [55] H. Yang, P. Antonante, V. Tzoumas, and L. Carbone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.
- [56] A. Santamaría-Navarro, R. Thakker, D. D. Fan, B. Morrell, and A.-a. Agha-mohammadi, "Towards resilient autonomous navigation of drones," in *The International Symposium of Robotics Research*. Springer, 2019, pp. 922–937.
- [57] Y. Tian, K. Khosoussi, and J. P. How, "A resource-aware approach to collaborative loop-closure detection with provable performance guarantees," *The International Journal of Robotics Research*, vol. 40, no. 10–11, pp. 1212–1233, 2021.
- [58] M. Giamou, K. Khosoussi, and J. P. How, "Talk resource-efficiently to me: Optimal communication planning for distributed loop closure detection," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3841–3848.
- [59] Y. Tian, K. Khosoussi, M. Giamou, J. How, and J. Kelly, "Near-optimal budgeted data exchange for distributed loop closure detection," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.
- [60] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed certifiably correct pose-graph optimization," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2137–2156, 2021.
- [61] S. Khattak, C. Papachristos, and K. Alexis, "Visual-thermal landmarks and inertial fusion for navigation in degraded visual environments," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–9.
- [62] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 298–304.
- [63] S. Khattak, F. Mascarich, T. Dang, C. Papachristos, and K. Alexis, "Robust thermal-inertial localization for aerial robots: A case for direct methods," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2019, pp. 1061–1068.
- [64] S. Khattak, C. Papachristos, and K. Alexis, "Keyframe-based thermal-inertial odometry," *Journal of Field Robotics*, vol. 37, no. 4, pp. 552–579, 2020, wiley Online Library.
- [65] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.
- [66] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," 2017.
- [67] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.
- [68] C. Forster, L. Carbone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.
- [69] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1418–1425, 2018.
- [70] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International conference on computer vision*. IEEE, 2011, pp. 2548–2555.
- [71] Google, "DRACO: 3D Data Compression," <https://google.github.io/draco/>, online; accessed 26 July 2022.

- [72] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization," in *Robotics: Science and Systems*, vol. 1, 2015, p. 1.
- [73] A. Agha, K. Otsu, B. Morrell, D. D. Fan, R. Thakker, A. Santamaría-Navarro, S.-K. Kim, A. Bouman, X. Lei, J. Edlund *et al.*, "Nebula: Quest for robotic autonomy in challenging environments; team costar at the DARPA subterranean challenge," *arXiv preprint arXiv:2103.11470*, 2021.
- [74] A. Reinke, M. Palieri, B. Morrell, Y. Chang, K. Ebadi, L. Carlone, and A.-a. Agha-Mohammadi, "LOCUS 2.0: Robust and computationally efficient lidar odometry for real-time underground 3D mapping," *Submitted to the IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2022.
- [75] E. Jones, J. Sofonia, C. Canales, S. Hrabar, and F. Kendoul, "Applications for the hovermap autonomous drone system in underground mining operations," *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 120, no. 1, pp. 49–56, 2020.
- [76] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.
- [77] C. E. Denniston, Y. Chang, A. Reinke, K. Ebadi, G. S. Sukhatme, L. Carlone, B. Morrell, and A.-a. Agha-Mohammadi, "Loop closure prioritization for efficient and scalable multi-robot slam," *Submitted to the IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2022.
- [78] A. Tagliabue, J. Tordesillas, X. Cai, A. Santamaría-Navarro, J. P. How, L. Carlone, and A.-a. Agha-mohammadi, "Lion: Lidar-inertial observability-aware navigator for vision-denied environments," in *International Symposium on Experimental Robotics*. Springer, 2020, pp. 380–390.
- [79] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 809–816.
- [80] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
- [81] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.
- [82] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3D registration," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 3212–3217.
- [83] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.
- [84] M. Ramezani, K. Khosoussi, G. Catt, P. Moghadam, J. Williams, P. Borges, F. Pauling, and N. Kottege, "Wildcat: Online Continuous-Time 3D Lidar-Inertial SLAM," *arXiv preprint arXiv:2205.12595*, 2022.
- [85] N. Hudson, F. Talbot, M. Cox, J. Williams, T. Hines, A. Pitt, B. Wood, D. Frousheger, K. L. Surdo, T. Molnar *et al.*, "Heterogeneous ground and air platforms, homogeneous sensing: Team csiro data61's approach to the DARPA subterranean challenge," *Journal of Field Robotics*, vol. 2, 2022.
- [86] EMESENT, "Emesent-autonomy technology for industrial drones," <https://emesent.io>, 2017, online; accessed 22 February 2022.
- [87] M. Bosse and R. Zlot, "Continuous 3D scan-matching with a spinning 2d laser," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 4312–4319.
- [88] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE international conference on rehabilitation robotics*. IEEE, 2011, pp. 1–7.
- [89] V. Kubelka, M. Vaidis, and F. Pomerleau, "Gravity-constrained point cloud registration," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [90] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale, and R. Siegwart, "Long-term 3D map maintenance in dynamic environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3712–3719.
- [91] T. Baca, M. Petrik, M. Vrba, V. Spurny, R. Penicka, D. Hert, and M. Saska, "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*, vol. 102, no. 1, pp. 1–28, 2021.
- [92] V. Pritzl, M. Vrba, C. Tortorici, R. Ashour, and M. Saska, "Adaptive estimation of uav altitude in complex indoor environments using degraded and time-delayed measurements with time-varying uncertainties," *submitted to Robotics and Autonomous Systems*, 2022.
- [93] Wikipedia, "nice (Unix)," [https://en.wikipedia.org/wiki/Nice\\_\(Unix\)](https://en.wikipedia.org/wiki/Nice_(Unix)), online; accessed 26 July 2022.
- [94] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, "Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8729–8736.
- [95] S. Zhao, P. Wang, H. Zhang, Z. Fang, and S. Scherer, "Tp-tio: A robust thermal-inertial odometry with deep thermalpoint," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4505–4512.
- [96] F. Nobre, C. R. Heckman, and G. T. Sibley, "Multi-sensor slam with online self-calibration and change detection," in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 764–774.
- [97] M. Kasper, S. McGuire, and C. Heckman, "A benchmark for visual-inertial odometry systems employing onboard illumination," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5256–5263.
- [98] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5135–5142.
- [99] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "Isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.
- [100] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [101] Lord, "Lord microstrain 3dm-gx5-vru datasheet," [https://www.microstrain.com/sites/default/files/3dm-gx5-15\\_datasheet\\_8400-0094\\_rev\\_o.pdf](https://www.microstrain.com/sites/default/files/3dm-gx5-15_datasheet_8400-0094_rev_o.pdf), 2020, online; accessed 15 February 2022.
- [102] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets: Open-source library and experimental protocol," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.
- [103] P. Petracek, V. Kratky, M. Petrlik, T. Baca, R. Kratochvil, and M. Saska, "Large-Scale Exploration of Cave Environments by Unmanned Aerial Vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7596–7603, 10 2021.
- [104] P. De Petris, H. Nguyen, M. Dharmadhikari, M. Kulkarni, N. Khedekar, F. Mascarich, and K. Alexis, "Rmf-owl: A collision-tolerant flying robot for autonomous subterranean exploration," *arXiv preprint arXiv:2202.11055*, 2022.
- [105] Y. Chang, K. Ebadi, C. E. Denniston, M. F. Ginting, A. Rosinol, A. Reinke, M. Palieri, J. Shi, A. Chatterjee, B. Morrell, L. Carlone, and A.-a. Agha-Mohammadi, "LAMP 2.0: A robust multi-robot slam system for operation in challenging large-scale underground environments," *IEEE Robotics and Automation Letters*, 2022.
- [106] AUTOMAP, "Automap: The future starts with an accurate map," <https://automap.io>, 2016, online; accessed 22 February 2022.
- [107] B. Nisar, P. Foehn, D. Falanga, and D. Scaramuzza, "Vimo: Simultaneous visual inertial model-based odometry and force estimation," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2785–2792, 2019.
- [108] M. Khader and S. Cherian, "An introduction to automotive lidar," *Texas Instruments Incorporated*, 2020.
- [109] S.-P. Deschênes, D. Baril, V. Kubelka, P. Giguère, and F. Pomerleau, "Lidar scan registration robust to extreme motions," in *2021 18th Conference on Robots and Vision (CRV)*, 2021.
- [110] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Machine Intell. (TPAMI)*, 2020.
- [111] A. Kramer, C. Stahoviak, A. Santamaría-Navarro, A.-A. Agha-Mohammadi, and C. Heckman, "Radar-inertial ego-velocity estimation for visually degraded environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5739–5746.
- [112] C. X. Lu, S. Rosa, P. Zhao, B. Wang, C. Chen, J. A. Stankovic, N. Trigoni, and A. Markham, "See through smoke: robust indoor mapping with low-cost mmwave radar," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020, pp. 14–27.
- [113] A. Kramer and C. Heckman, "Radar-inertial state estimation and obstacle detection for micro-aerial vehicles in dense fog," in *International Symposium on Experimental Robotics*. Springer, 2020, pp. 3–16.

- [114] C. X. Lu, M. R. U. Saputra, P. Zhao, Y. Almalioglu, P. P. de Gusmao, C. Chen, K. Sun, N. Trigoni, and A. Markham, “milliego: single-chip mmwave radar aided egomotion estimation via deep sensor fusion,” in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 109–122.
- [115] A. Kramer, K. Harlow, C. Williams, and C. Heckman, “Coloradar: The direct 3D millimeter wave radar dataset,” *arXiv preprint arXiv:2103.04510*, 2021.
- [116] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, “Mulran: Multimodal range dataset for urban place recognition,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6246–6253.
- [117] P. Antonante, H. Nilsen, and L. Carlone, “Monitoring of perception systems: Deterministic, probabilistic, and learning-based fault detection and identification,” *arXiv preprint arXiv: 2205.10906*, 2022.
- [118] D. Baril, S.-P. Deschênes, O. Gamache, M. Vaidis, D. LaRocque, J. Laconte, V. Kubelka, P. Giguère, and F. Pomerleau, “Kilometer-scale autonomous navigation in subarctic forests: challenges and lessons learned,” *Field Robotics*, 2022.
- [119] A. Davison and J. Ortiz, “FutureMapping 2: Gaussian belief propagation for spatial AI,” *arXiv preprint arXiv: 1910.14139*, 2019.
- [120] N. Hughes, Y. Chang, and L. Carlone, “Hydra: a real-time spatial perception engine for 3D scene graph construction and optimization,” in *Robotics: Science and Systems*, 2022.