

LAMP 2.0: A Robust Multi-Robot SLAM System for Operation in Challenging Large-Scale Underground Environments

Yun Chang^{1,*}, Kamak Ebadi^{2,*}, Christopher E. Denniston³, Muhammad Fadhil Ginting², Antoni Rosinol¹, Andrzej Reinke⁴, Matteo Palieri⁵, Jingnan Shi¹, Arghya Chatterjee⁶, Benjamin Morrell², Ali-akbar Agha-mohammadi², Luca Carbone¹

Abstract—Search and rescue with a team of heterogeneous mobile robots in unknown and large-scale underground environments requires high-precision localization and mapping. This crucial requirement is faced with many challenges in complex and perceptually-degraded subterranean environments, as the onboard perception system is required to operate in off-nominal conditions (poor visibility due to darkness and dust, rugged and muddy terrain, and the presence of self-similar and ambiguous scenes). In a disaster response scenario and in the absence of prior information about the environment, robots must rely on noisy sensor data and perform Simultaneous Localization and Mapping (SLAM) to build a 3D map of the environment and localize themselves and potential survivors. To that end, this paper reports on a multi-robot SLAM system developed by team CoSTAR in the context of the DARPA Subterranean Challenge. We extend our previous work, LAMP, by incorporating a single-robot front-end interface that is adaptable to different odometry sources and lidar configurations, a scalable multi-robot front-end to support inter- and intra-robot loop closure detection for large scale environments and multi-robot teams, and a robust back-end equipped with an outlier-resilient pose graph optimization based on Graduated Non-Convexity. We provide a detailed ablation study on the multi-robot front-end and back-end, and assess the overall system performance in challenging real-world datasets collected across mines, power plants, and caves in the United States. We also release our multi-robot back-end datasets (and the corresponding ground truth), which can serve as challenging benchmarks for large-scale underground SLAM.

Index Terms—Multi-Robot SLAM, SLAM, Multi-Robot Systems, Field Robots

I. INTRODUCTION

This work was supported by the Jet Propulsion Laboratory – California Institute of Technology, under a contract with the National Aeronautics and Space Administration. This work was partially funded by the Defense Advanced Research Projects Agency (DARPA). ©2022 All rights reserved.

*Equal contribution. Corresponding Authors: yunchang@mit.edu, kamak.ebadi@jpl.nasa.gov

¹Y. Chang, A. Rosinol, J. Shi, L. Carbone are with the Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, United States.

²K. Ebadi, M. Ginting, B. Morrell, A. Agha-mohammadi are with the NASA Jet Propulsion Laboratory – California Institute of Technology, United States.

³C. E. Denniston is with Robotic Embedded Systems Laboratory, University of Southern California, United States.

⁴A. Reinke is with the University of Bonn, Germany.

⁵M. Palieri is with the Dept. of Electrical And Information Engineering, Polytechnic University of Bari, Italy.

⁶A. Chatterjee is with Dept. of Mechanical Engineering, Bangladesh University of Engineering and Technology, Bangladesh.

Digital Object Identifier (DOI): see top of this page.



Fig. 1: (a) A heterogeneous team of robots undergoing initial calibration; (b) Kentucky Underground dataset: 4 robots traversing over 6 km (combined) in the Kentucky Underground Storage, Wilmore, Kentucky; (c) Tunnel dataset: 2 robots covering 2.5 km in the NIOSH safety research mine, Pittsburgh, Pennsylvania; (d) Urban dataset: 3 robots covering a multi-floor environment in the Satsop nuclear power plant, Elma, Washington; (e) Finals dataset: 4 robots mapping a heterogeneous course during the DARPA Subterranean Challenge finals in the Louisville Mega Cavern, Kentucky.

SIMULTANEOUS Localization And Mapping (SLAM) is a mature field of research, and there is a substantial body of literature dedicated to advancing SLAM algorithms and systems [1]. While there has been great progress in SLAM in urban and structured environments, localization and mapping in extreme underground scenarios has recently received increasing attention. In particular, the recent DARPA Subterranean Challenge [2] had the goal of developing robotic systems capable of exploring and mapping complex underground environments, to support high-precision localization of elements of interest (e.g., survivors). This paper presents the design and implementation of a multi-robot lidar-centric SLAM system, as an enabling factor for autonomous exploration of challenging underground environments.

Related work. We review related work on SLAM systems, and particularly lidar-based SLAM systems for subterranean and perceptually degraded environments, and refer the reader to [1] for a broader survey on SLAM. Early efforts on SLAM in subterranean environments can be traced back to the works of Thrun et al. [3] and Nuchter et al. [4], 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. [5], [6] presented a SLAM system for

exploration of underground tunnels using a team of robots. Zlot et al. [7] presented a 3D SLAM system using a spinning 2D lidar and an industrial-grade MEMS IMU to map over 17 kilometers of an underground mine. The three-year-long DARPA Subterranean (SubT) challenge, which began in 2018, aimed at developing innovative approaches to rapidly map, navigate, and search underground environments with a team of autonomous robots. This worldwide competition has led to developing breakthrough technologies and capabilities for underground robotic operation [8]–[30]. Rousek et al. [11] present a multi-robot heterogeneous system for robotic exploration of hazardous subterranean environments; the aerial platforms rely on Hector SLAM [31], and the ground robots use an ICP-based SLAM system [32]. Similarly, Petrik et al. [12] present a system intended for aerial Search and Rescue (SAR) operations in underground settings. Kramer et al. [15] present a sparse pose graph-based visual-inertial SLAM system that relies on onboard lighting for exploration of the subterranean environments; the system uses multiple cameras, with significant overlap between their fields of view. The system’s front-end detects visual features and matches them to previously seen landmarks using BRISK features [33], and it uses IMU measurements to propagate the camera pose from the last optimized pose to the current frame time and predicts where matched landmarks should be observed in the new pose. The back-end uses batch optimization over a sliding window of recent camera frames and IMU measurements to estimate the system’s state.

Azpuru et al. [19] present a pipeline for semi-autonomous exploration of confined environments (e.g., pipes, underground tunnel systems) that includes both offline photogrammetry for photorealistic map construction, and two parallel SLAM methods, a vision-based system based on RTAB-Map [34], and a lidar-based system based on LeGO-LOAM [35] for localization and mapping. Dang et al. [20], [22] present a loosely-coupled multi-modal SLAM system based on fusion of LiDAR, visual, thermal, and inertial data to provide robust and resilient localization in underground environments. In a similar study, Wisth et al. [21] present a graph-based odometry system for mobile robotic platforms based on integration of lidar features with standard visual features and IMU data. The system is able to handle under-constrained geometry that affects lidar, or textureless areas that affects vision by using the best information available from each sensor modality without any hard switches. Lajoie et al. present DOOR-SLAM [36], a fully distributed 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 front-end that detects inter-robot loop closures without exchanging raw sensor data. Chang et al., present Kimera-Multi [37], a fully distributed multi-robot system for dense metric-semantic SLAM. Each robot builds a local trajectory estimate and a local mesh. When two robots are within communication range, they initiate a distributed place recognition and robust pose graph optimization protocol.

In our previous work [8], we presented *Large-scale Autonomous Mapping and Positioning* (LAMP), a pose-graph-

based SLAM system developed in the context of the SubT Challenge. LAMP consisted of a front-end that performs lidar scan matching to obtain odometric estimates, and a back-end that performs pose graph optimization to obtain the best estimate of the robots’ trajectories given odometry and loop closure measurements. However, LAMP struggled with larger-scale long-duration operation in unknown subterranean environments subject to perceptual aliasing.

Contribution. This paper presents LAMP 2.0, a field-tested SLAM system for cooperative localization and mapping in unknown subterranean environments with a heterogeneous multi-robot team. The key contributions are:

- 1) A computationally efficient and outlier-resilient centralized multi-robot SLAM system that is adaptable to different input odometry sources, developed in the context of the DARPA Subterranean Challenge for operation in large-scale underground environments. The system includes the following improved modules:
 - a) A robust and scalable loop closure detection module that is able to handle and prioritize a rapidly growing number of loop closure candidates and includes modern 3D registration techniques to improve the accuracy and robustness of the detected loop closures.
 - b) An outlier-robust back-end based on Graduated Non-Convexity [38] for pose graph optimization.
- 2) The open-source release of LAMP 2.0¹ and a multi-robot dataset² of subterranean environments, including the pose graph and point clouds of caves, mines, and abandoned power plants (Fig. 1), along with ground truth trajectories and maps based on professionally surveyed data that can be used by the SLAM community for evaluation of novel multi-robot localization and mapping solutions.

II. LAMP 2.0

This section introduces LAMP 2.0, the SLAM system we developed in the context of the SubT Challenge for robust and scalable multi-robot SLAM in subterranean environments. Fig. 2 provides an overview of our system architecture.

The system consists of (i) a *single-robot front-end interface* on each robot that takes in an odometry and point cloud stream and send to the centralized base station pose graph segments, which consists of odometry edges and nodes, and keyed scans, which are point clouds associated to odometry nodes, (ii) a *multi-robot front-end*, running on a centralized 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 front-end to perform a joint pose graph optimization; the multi-robot back-end is also executed at the central station and simultaneously optimizes all the robot trajectories. In the rest of this section, we highlight the core components of LAMP 2.0 that enable large-scale and robust, multi-robot localization and mapping in challenging underground environments.

¹<https://github.com/NeBula-Autonomy/LAMP>

²<https://github.com/NeBula-Autonomy/nebula-multirobot-dataset>

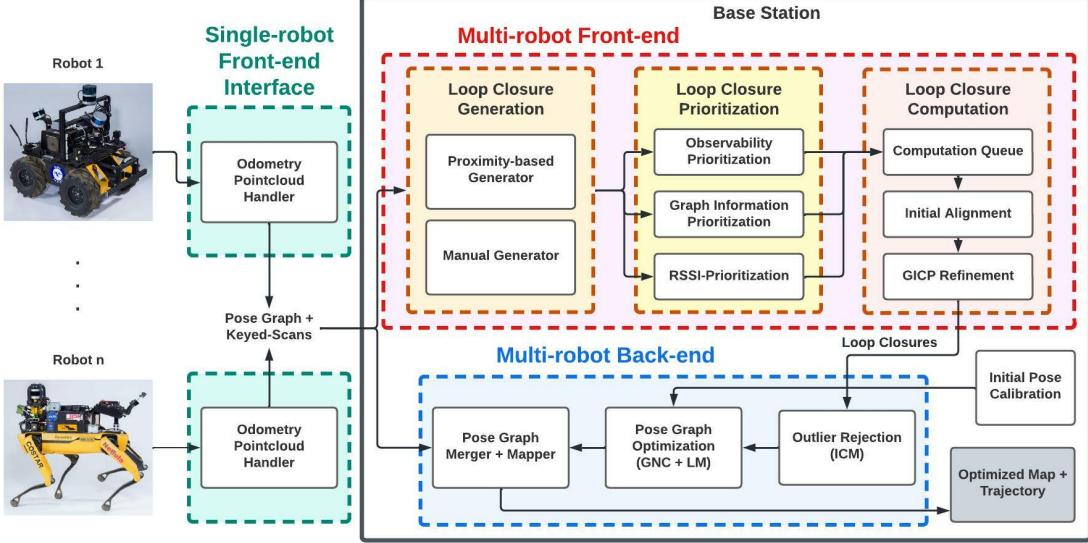


Fig. 2: Architecture of LAMP 2.0, our multi-robot localization and mapping system, which consists of a single-robot front-end interface to produce pose graph and keyed scans input (adaptable to different odometry sources and lidar configurations) on each robot, a multi-robot front-end for loop closure detection, and a multi-robot back-end for outlier-resilient pose graph optimization and map generation.

A. Initial Pose Calibration

The sensors onboard each robot are assumed to be calibrated before operation (both intrinsic and extrinsic calibration). Then, the initial pose calibration has the goal of establishing a common reference frame for all robots in the team. We achieve this by placing three reflective plates (fiducial markers) with known 3D coordinates on a gate at the entrance of the subterranean environment, and placing the robots in front of the gate. The robots then begin scanning the environment using the onboard lidar scanners. The lidar points corresponding to the three reflective plates are then segmented into three clusters using Euclidean clustering. The lidar pose can then be computed by finding the centroids of the point clusters, and solving for the lidar pose that minimizes the distance between all triplets of cluster centroids, and the triplet of known fiducial markers’ positions.

This gate calibration method is employed for compatibility with the SubT Challenge, but alternatives —such as the method in [39]— can be also implemented to establish a common reference frame without initial pose calibration.

B. Adaptable Single-robot Front-end Interface

The single-robot front-end interface allows LAMP 2.0 to interface with single-robot front-ends with different odometry sources and different lidar configurations. As we will demonstrate in the experiments section, LAMP 2.0 is able to support in the same experiment, robots with different number and configuration of lidar sensors and possibly using different odometry front-ends, including LOCUS [40] and Hovermap [41]. In the following sections we detail how the interface processes the point-cloud and odometry inputs.

Point cloud pre-processing. The front-end interface starts with a pre-processing step. First, any distortions in lidar scans caused by the robot motion are corrected using our Heterogeneous Robust Odometry (HeRO) [42] system; HeRO computes a local state estimate, and the relative motion during

the acquisition of a lidar scan is then corrected using the local state estimate. Next, given the extrinsic calibration between lidar scanners mounted on the robot, the undistorted point clouds are merged together into a single point cloud. To remove redundant and noisy points in the unified point cloud and to ensure that we retain a constant number of points regardless of environment geometry, point cloud density, and number of onboard lidar scanners, we apply an adaptive voxelization filter [40]. The adaptive voxelization filter ensures that point clouds received from different lidar configurations are comparable in terms of size and density. This improves the performance of the loop closure detection module by reducing the computational load associated with loop closure transform estimation, and also reduces the computation and memory usage of the mapper on the centralized base station.

Pose graph and keyed scans generation. To represent the estimated robot trajectory, LAMP uses a pose graph formulation [1] where each node in the graph corresponds to an estimated pose (i.e., robot pose, or position of objects of interest in the scene), while each edge connecting two nodes in the graph encodes the relative motion or position measurement between the pair of nodes. It is crucial to prevent the pose graph from growing too large due to limited computation and memory available onboard each robot. Thus, we use a sparse pose graph with new nodes only added after the robot exceeds a motion threshold (e.g., 2 m in translation and 30 degrees of rotation). We refer to these robot pose nodes as the *key nodes*. Each key node is associated with a *keyed scan*, which is the pre-processed point cloud obtained at the corresponding time. The pose graph and the keyed scans are sent to the multi-robot front-end for loop closure detection, and to the back-end for pose graph optimization and 3D map generation. When a robot moves out of communication range, the pose graph and keyed scans are queued up and are then sent as a batch to the base-station when the communication is re-established. We refer the reader to [43] for additional details on the networking setup.

C. Scalable Multi-robot Front-end

In our SLAM architecture, the multi-robot front-end is in charge of intra- and inter-robot loop closure detection. The ability to assert that a robot has returned to a previously visited location (i.e., a loop closure) is a crucial capability to reduce the accumulated error in the estimated robot trajectory. Moreover, loop closures are essential in detecting overlapping parts between local maps created by robots in the team, in order to fuse them into a geometrically consistent global map. Our loop closure detection module consists of three steps: Loop closure generation, prioritization, and computation. In the rest of this section, we describe the main components of LAMP 2.0’s multi-robot front-end and highlight prioritization techniques that allow our system to scale to large-scale environments and large teams of robots.

Loop closure generation. In the generation step, loop closure candidates are generated from nodes that lie within a certain Euclidean distance d_{max} from the current node in our pose-graph representation; d_{max} is adaptive and is defined as $d_{max} = \alpha|n_{curr} - n_{cand}|$ based on the relative traversal between two nodes for the single-robot case, and $d_{max} = \alpha n_{curr}$ based on the absolute traversal for the multi-robot case, where n_{curr} and n_{cand} denote the sequentially assigned index of the current node and the index of the candidate node respectively, and α is a fixed constant that decides the magnitude of d_{max} with respect to the traversed distance (0.2 m in our experiments). The design of the loop closure module is modular, such that other methods and environment representations (i.e., Bag-of-visual-words [44], junctions extracted from 2D occupancy grid maps [10]) can also be used for candidate generation.

Loop closure prioritization. The prioritization step inspects and adds the candidates to the queue for the computation step in an order based on how likely they are to improve the trajectory estimate.

Subterranean environments contain many self-similar and ambiguous scenes. While loop closure is 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 [10], as it could lead to spurious loop closure detections, or inaccurate estimation of rigid-transformation between two corresponding nodes due to a lack of observability. Furthermore, loop closure detection in large-scale environments, and with multiple robots, becomes increasingly more challenging as the density of nodes in the pose graph, and subsequently the number of loop closure candidates, increases. To overcome these challenges, our loop closure prioritization step includes three independent modules:

The *Observability Prioritization* module prioritizes feature-rich areas, as they are more likely to produce more accurate loop closures. To identify feature-rich scenes, similar to the works presented in [10], [45], [46], we rely on eigenvalue analysis of the information matrix of the pose estimate produced by point-to-plane ICP.

The *Graph Information Prioritization* module prioritizes loop closures that can lead to a more substantial reduction of the trajectory error, given the structure of the graph. We use a

Graph Neural Network (GNN) [47] to predict the outcome of a pose graph optimization triggered by a new loop closure.

The *Receiver Signal Strength Indication (RSSI) Prioritization* module prioritizes loop closures based on known locations indicated by RSSI beacons. We refer the reader to [48] for a more in-depth discussion and analysis of the prioritization module.

Loop closure computation. The computation step calculates the relative transform for each candidate loop closure in the queue; for each loop closure candidate, we implemented a two-stage pipeline to first find an initial alignment for the relative transform estimate using TEASER++ [49] or SAmple Consensus Initial Alignment (SAC-IA) [50]), then use the initial alignment to start the Generalized Iterative Closest Point (GICP) algorithm [51] to obtain the refined relative transform and filter out matches that correspond to poor alignments (i.e., have a larger accumulated error than the threshold after the initial alignment or after GICP). As presented in the experimental section, this two-stage pipeline significantly improves the accuracy and quality of loop closure detections, particularly because the GICP algorithm is prone to local minima.

D. Robust Multi-robot Back-end

In our SLAM architecture, the multi-robot back-end performs pose graph optimization to generate a globally consistent and drift-free 3D map of the environment. We adopt 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. The optimized map is then generated by transforming the keyed scans to the global frame using the optimized trajectory.

Outlier-robust pose graph optimization. The robust multi-robot back-end that receives odometry measurements from all robots in the form of odometry edges created by the single-robot front-end interface modules and inter- and intra-robot loop closures from the multi-robot front-end, and performs pose graph optimization to compute a globally consistent trajectory estimate for all robots in the team.

To safeguard against erroneous loop closures, our multi-robot back-end includes two outlier rejection options: Incremental Consistency Maximization (ICM) [8], 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) [38], which is used in conjunction with Levenberg-Marquardt to perform an outlier-robust pose graph optimization to 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 the Georgia Tech Smoothing and Mapping library (GTSAM) [52]. Table I, provides a summary of new capabilities and features implemented in LAMP 2.0, which extends our previous work [8].

TABLE I: LAMP vs. LAMP 2.0 capabilities by front-end (FE), back-end (BE) and overall system (Sys)

	Capability	LAMP	LAMP 2.0
FE	Proximity-based loop closure generation	X	X
	Adaptive Proximity-based loop gen		X
	Loop closure prioritization		X
	GICP loop closure pose estimation	X	X
BE	Two stage loop closure pose estimation		X
	LM / GN pose graph optimization	X	X
	ICM outlier rejection	X	X
Sys	GNC in conjunction with LM / GN		X
	Adaptable to different lidar configs	X	
Sys	Adaptable to different odometry input		X

III. EXPERIMENTS

This section showcases the performance of LAMP 2.0 and provides experimental results to highlight the new features and improvements from the previous version [8]. Section III-A describes our hardware setup. Section III-B describes the dataset we collected, open-sourced, and used for evaluation. Section III-C evaluates and discusses the performances of selected components in the LAMP 2.0 multi-robot front-end and back-end. Section III-D reports on the performance of the overall LAMP 2.0 system, including its performance in the three events of the SubT challenge.

A. Hardware Setup

Robots. In the experiments we present in this paper, the datasets are collected with two types of robots: Husky and Spot robots. The Husky is a wheeled platform and is equipped with three Velodyne lidars and a Hovermap [41]. The odometry input to LAMP 2.0 is either provided by LOCUS [40] or the Hovermap [41]. The Spot robot is a quadruped platform equipped with either a single lidar or a Hovermap. Similarly to the Husky, the odometry input to LAMP is either provided by the Hovermap or LOCUS [40]. Note that not only do the odometry inputs to the single-robot front-end interface vary (between Hovermap and LOCUS), but also the point cloud inputs vary in size and density between the Spot robots with a single lidar, Spot robots with a Hovermap, Husky robots with three lidars, and Husky robots with a Hovermap.

Base station. During the SubT Challenge, the base was a portable workstation with an AMD Ryzen Threadripper 3990X processor with 64 cores. The experiments in this paper ran on a laptop with Intel i7-8750H processor with 12 cores.

B. Datasets

We evaluate LAMP 2.0 on four datasets collected by Team CoSTAR. The first dataset is the **Tunnel Dataset**, which was collected in the NIOSH Safety Research Coal Mine in Pittsburgh, PA, and includes two Huskies traversing up to 2.5 km combined, in a coal mine that consists of mostly featureless narrow tunnels. The second dataset is the **Urban Dataset**, which was collected in the Satsop abandoned nuclear power plant in Elma, WA, and includes two Huskies and a Spot traversing up to 1.5 km combined in an abandoned nuclear power plant consisting of a two-floor environment with open areas, small rooms, narrow passageways, and stairs. The third dataset is the **Finals Dataset**, which was collected during the second preliminary run of the SubT finals and includes three

Spots and a Husky traversing up to 1.2 km combined in the DARPA-built course including tunnel, cave, and urban-like environments. The last dataset is the **Kentucky Underground Dataset (KU)**, which was collected by Team CoSTAR in the Kentucky Underground Storage in Wilmore, KY, and includes four Huskies traversing up to 6 km combined in a limestone mine, which consists of large 10-20 m wide tunnels. The maps and sample scenes from the four datasets are shown in Fig. 1.

For each dataset, we used a surveyed global map to generate an odometry ground-truth through scan-to-map localization [40]. To test the performance of multi-robot loop closure detection, we generate sets of ground-truth loop closure pairs from the ground-truth trajectory, including the correct transform, and also a set of false loop closures to stress the robustness of the system.

We have prepared these datasets, in the format of pose graphs and keyed scans, and released them open-source to promote further research on multi-robot loop closure detection and robust pose graph optimization.

C. Component Evaluation

This section focuses on the evaluation of the multi-robot front-end and the multi-robot back-end, and highlights the new features in these two modules compared to our previous system in [8], in order to assess their impact on our new LAMP 2.0 system.

We will first demonstrate the improvements to loop closure detection accuracy with the two-stage loop closure computation module. Then, we will present an ablation study with different outlier rejection configurations and highlight the performance of GNC. Lastly, we will evaluate the multi-robot front-end and back-end in conjunction, and show that with the new features in the front-end and the back-end, we are able to include more accurate loop closures in our pose graph and obtain a better trajectory and map estimate. We refer the reader to [48] for detailed experiments and analysis of the loop closure prioritization module.

Two-stage loop closure computation. We evaluate the results from the loop closure computation step for different types of initializations for ICP. Previously, LAMP used an odometric initialization approach [8]; on the other hand, LAMP 2.0 uses a two-stage approach and uses TEASER++ [49] or SAC-IA [50] for initial alignment (stage 1) and ICP for final refinement (stage 2). Table II shows the recall and false-positive rate, along with the mean translation and rotation error of the different initialization methods for ICP. In our experiments, we set the SAC cumulative error threshold to 32 m, the max number of iterations of SAC to 500, the ICP cumulative error threshold to 0.9 m, and the max number of ICP iterations to 200. The recall is the percentage of correct loop closures that passed SAC and ICP (i.e. had a lower SAC error than the threshold after initial alignment and had a lower ICP error than the threshold after ICP registration) and the false positive rate is the percentage of false loop closures that passed SAC and ICP. The mean and translation errors are computed using the correct loop closures that passed SAC and ICP against the ground truth trajectory. The odometry initialization typically

TABLE II: Comparison of different initialization methods for ICP-based loop closure relative pose estimation

		Initialization Methods			
		GT	OdomRot [8]	TEASER++	SAC-IA
\uparrow Recall (%)	Tunnel	90.8	93.9	76.6	81.9
	Urban	90.5	78.2	78.2	79.6
	Final	89.2	83.4	68.4	57.0
	KU	29.0	11.3	18.5	17.7
\downarrow F/P (%)	Tunnel	2.0	2.4	1.2	1.4
	Urban	0.8	1.6	0.6	1.0
	Final	1.2	7.4	0.6	1.0
	KU	0.4	0.0	0.2	0.0
\downarrow Mn Err. (m/deg)	Tunnel	0.09/0.86	0.86/8.02	0.71/10.82	0.67/9.63
	Urban	0.44/1.47	1.89/1.98	0.38/1.36	0.52/1.6
	Final	0.06/1.11	1.65/6.43	0.32/2.39	0.6/2.99
	KU	0.26/0.96	0.82/1.34	0.29/1.45	0.29/1.48

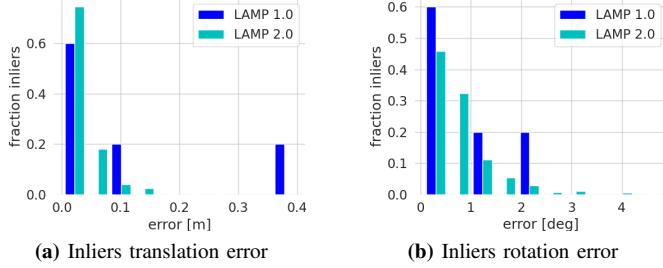


Fig. 3: Pose error of the loop closure inliers from LAMP 1.0 (Old Loop Closure module + ICM) compared to LAMP 2.0 (New Loop Closure module + GNC) for the Finals dataset.

has a better recall (i.e., more loop closures are found) in narrower environments, while TEASER++ and SAC-IA have better recall in wider environments (e.g., Urban and KU). More importantly, the newly available TEASER++ and SAC-IA initializations lead to a significant decrease in false-positive rate and mean translation and rotation error. The lower false-positive rate when using TEASER++ or SAC-IA is largely due to the additional filtering of poor alignments that do not have a sufficient number of inliers. LAMP 2.0 shows a decrease in the number of incorrect loop closures and increased pose accuracy for the correct loop closures.

Outlier-robust pose graph optimization. We demonstrate the capabilities of our outlier-robust pose graph optimization module with an ablation study, especially focusing on the performance of the newly added GNC option as compared to ICM in [8]. In Fig. 5 we compare the average final trajectory error with GNC or ICM as an outlier rejection method; we also report trajectory errors for the case when no outlier rejection is performed (“No Rej”) and when no loop closure is detected (“Odom”). While both GNC and ICM largely improve the trajectory estimate relative to the no loop closure or no outlier rejection case, GNC in general is more robust to outliers and gives better trajectory estimates, as especially evident in the longer Tunnel and Kentucky Underground datasets where ICM failed to reject some of the outliers in the first and was too conservative in the other.

Table III shows the number of loop closure candidates generated (“Generated”), the number of loop closures that passed ICP (“Verified”), and the number of ICM or GNC inliers (“Inliers”). Recall that LAMP 1.0 [8] has only a fixed

TABLE III: Amount of loop closures at the different stages in the multi-robot frontend and backend.

		# Generated	# Verified	# Inliers
Tunnel	LAMP 1.0	22206	3032	52
	LAMP 2.0	9755	5656	1645
Urban	LAMP 1.0	17742	497	237
	LAMP 2.0	5356	1505	284
Final	LAMP 1.0	12559	681	6
	LAMP 2.0	3856	1484	393
KU	LAMP 1.0	22616	14	11
	LAMP 2.0	22344	885	197

radius for candidate generation, no prioritization step, and uses an odometric ICP initialization for loop closure pose estimation and ICM as the outlier rejection method; on the other hand, the proposed LAMP 2.0 system includes candidate generated with an adaptive radius, the prioritization step, and uses SAC-IA/TEASER++ for ICP initialization and GNC as an outlier rejection method. With the new loop closure module and prioritization, even though we generate fewer candidates, we are able to find a larger number of verified and inlier loop closures, showing the validity of the adaptive radius, and that prioritization does prioritize loop closures of higher “quality”. Notice also that GNC is able to perform well with more than 80% outlier loop closures detected. Fig. 3 compares the translation and rotation errors (with respect to ground truth) of loop closures considered inliers by LAMP 1.0 [8] and LAMP 2.0. The figure shows how LAMP 2.0 not only produces a significantly larger number of inlier loop closures but also that the detected inliers are better biased towards lower errors compared to LAMP 1.0, confirming that the prioritization and initial alignment (Table II), together with the outlier rejection (Fig. 5) lead to an increased number of more accurate loop closures.

D. System Evaluation

In this section, we provide end-to-end results of the full LAMP 2.0 system. We play back the data in real-time to the base-station; the data is processed in the same order and rate as it was generated in the field and the results are collected at the end of data-stream (~ 1 hr). The odometry input is also consistent across the tests for fair comparison. In Table IV, we show the improvement in average trajectory error of LAMP 2.0 compared to LAMP 1.0 and LAMP 2.0 without inter-robot loop closures for each robot on the four datasets. LAMP 2.0 achieves minimal errors (below 2 m) with trajectory lengths of up to 2.2 km. Fig. 6 illustrates the impact of inter- and intra-robot loop closures. Finally, we present the full multi-robot mapping results by showing the LAMP 2.0 point cloud map colored by the cloud-to-cloud error against the ground truth map for the four datasets in Fig. 4. We are able to achieve map errors below 4 m in these different large-scale and challenging environments.

IV. CONCLUSION

In this paper we presented LAMP 2.0, a centralized multi-robot SLAM system, developed in the context of the DARPA Subterranean Challenge, which provides a robust estimate of

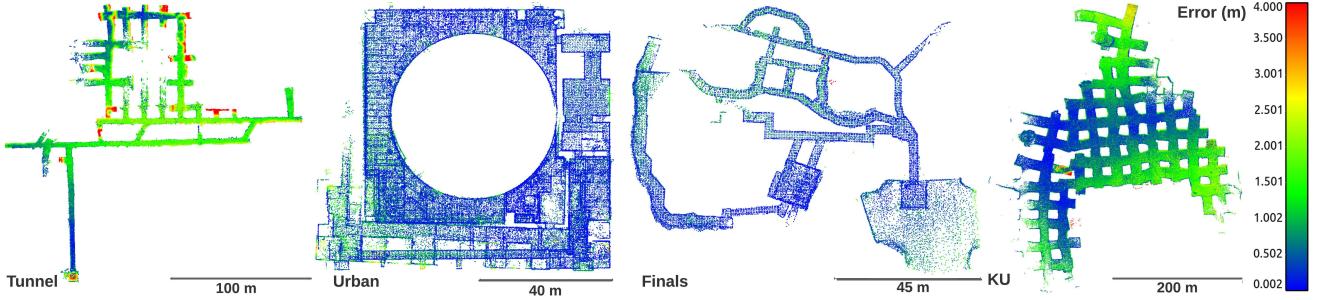


Fig. 4: Final map error in meters of the DARPA Subterranean Challenge finals course for the Tunnel, Urban, Finals, and Kentucky Underground (KU) datasets.

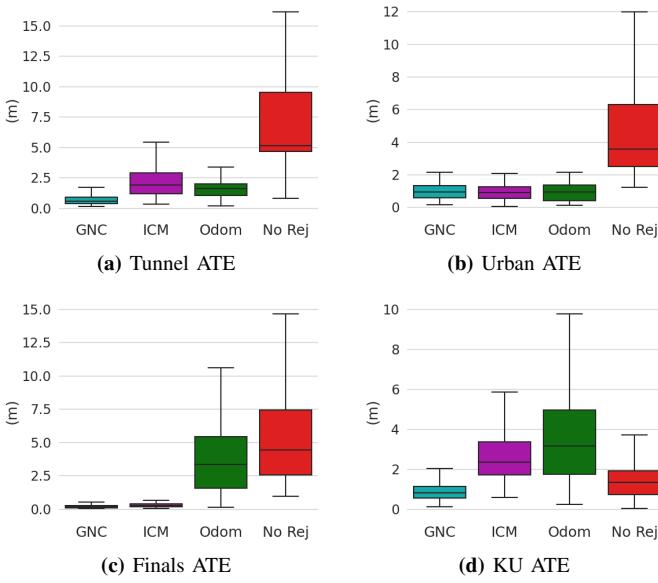


Fig. 5: Comparison of trajectory ATE (across the multi-robot trajectory) for ICM and GNC compared to the cases with no loop closure or no outlier rejection.

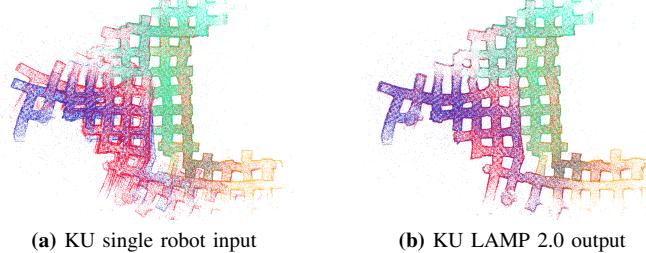


Fig. 6: Single-robot map (no loop closures) compared with LAMP 2.0 map (with inter- and intra- robot loop closures).

the trajectories of multiple robots and constructs a point cloud map using 3D lidar data. LAMP 2.0 includes a single-robot front-end interface that is adaptable to different odometry inputs and robots with different lidar configurations. The multi-robot back-end is based on a modular ICP initialization framework to improve the convergence of the algorithm in ambiguous settings, and uses a loop closure prioritization module to deal with the growing number of loop closure candidates in large-scale environments. Then the multi-robot back-end runs an outlier-robust pose graph optimization to estimate the trajectories of all robots in the team. While LAMP 2.0 performed well in our experiments with up to four robots, its centralized architecture may not scale to large robot teams.

TABLE IV: Comparison of LAMP 2.0 against LAMP 1.0

	Robot	Traversed [m]	LAMP 2.0 ATE [m]	Single Robot ATE [m]	LAMP 1.0 ATE [m]
Tunnel	husky3	1194	0.65	0.83	1.07
	husky4	1362	0.72	0.63	1.44
Urban	husky1	612	0.79	0.87	0.95
	husky4	416	0.76	0.79	0.78
	spot1	502	1.31	1.46	0.99
KU	husky1	2204	1.01	0.9	5.34
	husky2	1526	0.71	0.75	2.85
	husky3	1678	1.31	1.33	2.11
	husky4	896	0.69	0.86	5.49
Final	husky3	72	0.16	0.16	0.56
	spot1	430	0.2	0.37	0.37
	spot3	484	0.21	0.38	0.55
	spot4	238	0.15	0.23	0.62

Extending LAMP to a fully distributed system in underground domains with intermittent inter-robot communications is an interesting direction for future work.

REFERENCES

- [1] 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.
- [2] “Darpa subterranean (subt) challenge.” [Online]. Available: <https://www.subchallenge.com/>
- [3] 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, 2003, pp. 4270–4275 vol.3.
- [4] 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, 2004, pp. 1998–2003 Vol.2.
- [5] D. Tardioli, L. Sicignano, J. Riazuelo, L. Villarroel, and L. Montano, “Robot teams for exploration in underground environments,” *Workshop ROBOTI1: Robótica Experimental*, pp. 205–212, 2012.
- [6] D. Tardioli, L. Riazuelo, T. Seco, J. Espelosín, J. Lalana, J. Villarroel, and L. Montano, “A robotized dumper for debris removal in tunnels under construction,” 11 2018, pp. 126–139.
- [7] R. Zlot and M. Bosse, “Efficient large-scale 3d mobile mapping and surface reconstruction of an underground mine,” vol. 92, 07 2012.
- [8] K. Ebadi, Y. Chang, M. Palieri *et al.*, “Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments,” *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 80–86, 2020.
- [9] 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. PP, pp. 1–1, 12 2020.

- [10] K. Ebadi, M. Palieri, S. Wood, C. Padgett, and A.-a. Aghamohammadi, "Dare-slam: Degeneracy-aware and resilient loop closing in perceptually-degraded environments," *Journal of Intelligent & Robotic Systems*, vol. 102, 05 2021.
- [11] T. Rouček, M. Pecka, P. Čížek, T. Petricek, J. Bayer, V. Šalanský, D. Hert, M. Petrlík, T. Báča, V. Spurný, F. Pomerleau, V. Kubelka, J. Faigl, K. Zimmermann, M. Saska, T. Svoboda, and T. Krajiník, *DARPA Subterranean Challenge: Multi-robotic Exploration of Underground Environments*, 03 2020, pp. 274–290.
- [12] 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.
- [13] A. Bouman, M. F. Ginting, N. Alatur *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)*, 2020, pp. 2518–2525.
- [14] 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)*, 2020, pp. 263–270.
- [15] A. Kramer, M. Kasper, and C. Heckman, *VI-SLAM for Subterranean Environments*, 01 2021, pp. 159–172.
- [16] 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, 05 2021.
- [17] I. Miller, A. Cohen, A. Kulkarni, J. Laney, C. Taylor, V. Kumar, F. Cladera, A. Cowley, S. Skandan, E. Lee, L. Lipschitz, A. Bhat, N. Rodrigues, and A. Zhou, "Mine tunnel exploration using multiple quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 02 2020.
- [18] M. F. Ginting, K. Otsu, J. A. Edlund, J. Gao, and A.-A. Aghamohammadi, "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.
- [19] H. Azpúrrua, M. F. M. 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)*, 2021, pp. 2443–2449.
- [20] 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*, 2020, pp. 1–8.
- [21] 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. PP, pp. 1–1, 02 2021.
- [22] 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)*, 2020, pp. 1024–1029.
- [23] J. P. Queralta, J. Taipalmaa, B. Can Pullinen, V. K. Sarker, T. Nguyen Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191617–191643, 2020.
- [24] J. Gross, M. De Petrillo, J. Beard, H. Nichols, T. Swiger, R. Watson, C. Kirk, C. Kilic, J. Hikes, E. Upton, D. Ross, M. Russell, Y. Gu, and C. Griffin, "Field-testing of a uav-ugv team for gnss-denied navigation in subterranean environments," 10 2019, pp. 2112–2124.
- [25] 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, C. Heckman, and J. S. Humbert, "Multi-agent autonomy: Advancements and challenges in subterranean exploration," *ArXiv*, vol. abs/2110.04390, 2021.
- [26] J. Bayer and J. Faigl, "Speeded up elevation map for exploration of large-scale subterranean environments," in *Modelling and Simulation for Autonomous Systems*, J. Mazal, A. Fagiolini, and P. Vasik, Eds. Cham: Springer International Publishing, 2020, pp. 190–202.
- [27] B. Tidd, A. Cosgun, J. Leitner, and N. Hudson, "Passing through narrow gaps with deep reinforcement learning," 03 2021.
- [28] B. Lindqvist, C. Kanellakis, S. Mansouri, A.-a. Aghamohammadi, and G. Nikolakopoulos, "Compra: A compact reactive autonomy framework for subterranean mav based search-and-rescue operations," 08 2021.
- [29] 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)*, 2020, pp. 955–961.
- [30] J. Rogers, A. Schang, C. Nieto-Granda, J. Ware, J. Carter, J. Fink, and E. Stump, *The DARPA SubT Urban Circuit Mapping Dataset and Evaluation Metric*, 03 2021, pp. 391–401.
- [31] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 11 2011.
- [32] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing icp variants on real-world data sets," *Autonomous Robots*, 04 2013.
- [33] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International Conference on Computer Vision*, 2011, pp. 2548–2555.
- [34] M. Labb  , "Rtab-map as an open-source lidar and visual slam library for large-scale and long-term online operation," 2018.
- [35] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," 10 2018, pp. 4758–4765.
- [36] 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, pp. 1656–1663, 2020.
- [37] Y. Chang, Y. Tian, J. P. How, and L. Carlone, "Kimera-multi: a system for distributed multi-robot metric-semantic simultaneous localization and mapping," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11 210–11 218, 2021.
- [38] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 01 2020.
- [39] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE Transactions on Robotics*, 2022.
- [40] A. Reinke, M. Palieri, B. Morrell, Y. Chang, K. Ebadi, L. Carlone, and A. akbar Agha-mohammadi, "Locus 2.0: Robust and computationally efficient lidar odometry for real-time underground 3d mapping," 2022.
- [41] 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, 01 2020.
- [42] A. Santamar  -Navarro, R. Thakker, D. D. Fan, B. Morrell, and A. akbar Agha-mohammadi, "Towards resilient autonomous navigation of drones," in *ISRR*, 2019.
- [43] A. Agha, K. Otsu, B. Morrell *et al.*, "Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge," *ArXiv*, vol. abs/2103.11470, 2021.
- [44] D. Galvez-L  pez 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.
- [45] A. Tagliabue, J. Tordesillas, X. Cai, A. Santamar  -Navarro, J. P. How, L. Carlone, and A. akbar Agha-mohammadi, "Lion: Lidar-inertial observability-aware navigator for vision-denied environments," in *ISER*, 2020.
- [46] 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)*, 2016, pp. 809–816.
- [47] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," *ArXiv*, vol. abs/1812.08434, 2020.
- [48] 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," 2022. [Online]. Available: <https://arxiv.org/abs/2205.12402>
- [49] 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, 2021.
- [50] 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*, 2009, pp. 3212–3217.
- [51] A. Segal, D. H  nkel, and S. Thrun, "Generalized-icp," 06 2009.
- [52] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," 2012.