An Online Multi-Robot SLAM System for 3D LiDARs

Renaud Dubé Abel Gawel Hannes Sommer Juan Nieto Roland Siegwart Cesar Cadena*

***Abstract*— Using multiple cooperative robots is advantageous for time critical Search and Rescue (SaR) missions as they permit rapid exploration of the environment and provide higher redundancy, as opposed to using a single robot. A considerable number of applications such as autonomous driving and disaster response could benefit from merging mapping data from several sources. Online multi-robot localization and mapping has mainly been addressed for robots equipped with cameras or 2D LiDARs. However, in challenging real-life scenarios, a mapping system can potentially benefit from a rich 3D geometric solution. In this work, we present an online localization and mapping system for multiple robots equipped with 3D LiDARs. This system is based on incremental sparse pose-graph optimization using sequential and place recognition constraints, the latter being identified using a 3D segment matching approach. The result is a unified representation of the world and relative robot trajectories. The complete system is evaluated with two experiments in different environments, notably urban and disaster scenarios. The source code of the system is open sourced and easy to run demonstrations are publicly available.**

I. INTRODUCTION

Teams of autonomous mobile robots offer several advantages compared to their single robot counterpart: robustness to single robot failure, quicker exploration of environments in time critical SaR missions, execution of tasks of high complexity and reduction of human risks and costs associated to disaster response [1, 2]. Accurate online localization and mapping is a crucial competency for enabling collaboration between multiple mobile robots. This is however a difficult task as stated by Saeedi et al. [3] which identified 10 major challenges to achieve online multi-robot Simultaneous Localization and Mapping (SLAM). The current work addresses the challenges of *closing loops*, *complexity* and *communication*. Inter-robot global associations are found and used to solve the full 3D SLAM problem with minimum time, memory and communication bandwidth requirements.

There are already existing vision-based multi-robot SLAM approaches [3] which can however become unreliable when strong changes in illumination occur, and in the presence of strong viewpoint variations [4]. In this work, we therefore consider 3D LiDARs for their ability to accurately represent the inherent 3D nature of our environment while providing higher robustness to changes in external illumination and in view-point.

Current multi-robot SLAM systems for 3D LiDARs do not offer a complete online solution [5, 6] which is perhaps due to the absence of efficient algorithms to perform global data association with dense 3D point clouds. This is contrastingly a well-studied problem in visual SLAM. Global place recognition techniques for 3D point clouds based on global descriptors [7, 8] and keypoint descriptors [9, 10] are presented but rarely integrated in a full online SLAM system, let alone a multi-robot one. Contrastingly, most of the current 3D LiDAR single-robot approaches propose to recognize places based on local *submap* matching. These local searches cannot correct for drift which occurs when long distances are travelled and large estimation errors accumulate. In the multi-robot case, finding robot-associations by performing global search based on *submaps* does not scale well with increasing number of robots and would require the raw LiDAR data to be transmitted, which reflect the aforementioned multi-robot challenges.

This paper presents an online 3D LiDAR SLAM system capable of simultaneously and accurately estimating multiple trajectories, as illustrated in Figure 1. To the best of our knowledge, this is the first proposed solution to the online multi-robot SLAM problem for 3D LiDARs. To achieve this, a pose-graph formulation is adopted by incorporating sequential and place recognition constraints. We perform intra and inter-robot place recognition by leveraging our previously proposed and publicly available *SegMatch* algorithm [11] which was one key ingredient, along with

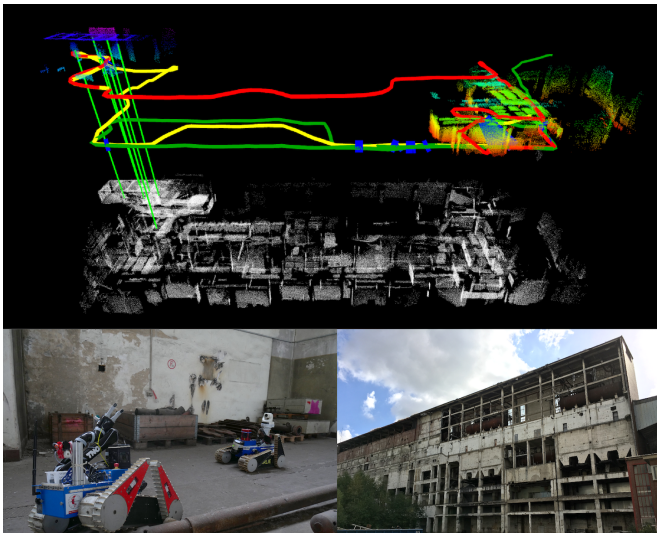


Fig. 1: Top: An illustration of the presented multi-robot SLAM system. The trajectories of three UGVs are estimated in real-time and shown in green, yellow and red. The target map M_t is shown below in white and an inter-robot place recognition is depicted with vertical green lines indicating segment matches. Bottom: Two skid-steering UGVs equipped with encoders, IMUs and rotating LiDARs and the decommissioned two-floors building considered in the power plant experiment of Section V-E.

*Authors are with the Autonomous Systems Lab, ETH, Zurich authors@mavt.ethz.ch.

This work was supported by the European Union's Seventh Framework Programme for research, technological development and demonstration under the TRADR project No. FP7-ICT-609763.

significant implementation efforts, in order to achieve a full working system. The *SegMatch* technique is formed on the basis of partitioning point clouds into sets of segments which efficiently represent the environment by compact yet discriminative features. This compact representation is crucial for multi-robot applications as it reduces the required communication bandwidth as well as the complexity and the memory requirement of the overall system. This is reflected in the proposed system which offers real-time performance for the experiments considered in this paper.

To summarize, this paper presents the following contributions:

- A fully-integrated online multi-robot SLAM system for 3D LiDARs.
- An evaluation of the entire system in real-world, multi-robot automotive and disaster scenario experiments.
- An open-source implementation accompanied with easy-to-run demonstrations[1].

The remainder of the paper is structured as follows: Section II provides an overview of the related work in the field of 3D LiDAR-based SLAM and multi-robot SLAM. Section III and IV describe our online multi-robot 3D pose-graph SLAM system. The full system is evaluated in Section V, and Section VI finally concludes with a short discussion.

II. RELATED WORK

This section gives an overview of the related work in single-robot 3D LiDAR-based SLAM, with a focus on pose-graph based approaches, and present current solutions to the multi-robot mapping problem.

A. Single robot pose-graph SLAM

The recent survey of Cadena et al. [12] reviews common approaches to the SLAM problem. This work considers the pose-graph approach which was pioneered by Lu and Milios [13] and became increasingly popular in recent years with an active community performing research in this direction [14–19].

Several works propose to apply the pose-graph approach to perform 3D SLAM for single robots equipped with LiDAR sensors [15–19]. These works mainly differ in terms of the technique used for matching new 3D scans to previous ones. For example, Pathak et al. [15] propose to register subsequent 3D scans on the basis of large planar surfaces which leads to robust estimation of rotations and simplifies the pose-graph relaxation to handle 3D translations only. This assumption of an explicit plane model would however typically not hold for unstructured environments. Droeschel et al. [16] introduce a multi-resolution surface element representation for the 3D scans which are obtained by accumulating scans from a rotating 2D LiDAR sensor. Matching the scans through this surface elements representation allows for efficient registrations. The systems proposed in [17–19], and ours are all based on Iterative Closest Point (ICP) for registering successive 3D scans and for augmenting the pose-graph with the corresponding scan-matching constraints.

Although related to this work, none of the aforementioned approaches explicitly deal with loop-closures. Instead, scan matching is performed against *submaps* containing nodes in the vicinity of the robot, assuming that only little drift occurred. In our system, place recognitions are explicitly dealt with, allowing the fusion of maps from independent workers and enabling the joint exploration of larger environments.

B. Multi-robot pose-graph SLAM

A thorough survey on multi-robot SLAM can be found in [3]. There is a significant amount of works proposing solutions to the SLAM problem for robots equipped with cameras or 2D LiDAR but much fewer works consider 3D LiDAR sensors [5, 6, 20].

Nagatani et al. [5] propose to merge digital elevation maps obtained from three robots where inter-robot constraints are found on the basis of *submap* matching by assuming little drift and a known good estimate of the relative transformation between the robots. The map-merging strategy is performed offline and the experiment only consider a small environment. Michael et al. [6] present a strategy for generating a 3D map of a building damaged by an earthquake. The maps are locally built on each robot using a technique which assumes the environment to be composed of walls and horizontal ground planes. The two maps are merged afterwards, providing a good initial guess for the relative robot transformation which is then refined by ICP. The two aforementioned solutions are not applicable to online multi-robot SLAM and cannot correct for drift which might occur in the single maps. Finally, Kurazume et al. [20] propose to model large buildings using multiple robots, one of which is equipped with a 3D LiDAR. The other robots are used to improve the sensor's localization by using direct inter-robot detection. Although the idea is interesting, this system differs from ours in that a single 3D LiDAR is used.

As we could not find a single work presenting an online SLAM system for multiple robots equipped with 3D LiDARs, we briefly introduce major multi-robot systems based on vision, with a focus on the back-end particularities. The work of Kim et al. [21] addresses the multi-robot mapping problem based on incremental optimization of multiple relative pose-graphs. These graphs are linked through *anchor nodes* which allow a relative formulation of the inter-robot encounter detections using april-tags. *Anchor nodes* which are the equivalent of *base nodes* first introduced in [22] were also used in a multi-session vision-based SLAM system [23]. Anchor nodes are agnostic to the sensor used and help in the convergence of the back-end pose-graph optimization. Contrastingly, Konolige and Bowman [24] introduced *weak-links* to allow each robot's pose-graph to grow independently while keeping the problem constrained in the absence of inter-robot constraints. Our system is based on *weak-links*

[1]The source code of the proposed system is open-sourced and demonstrations including a newly available dataset for multi-robot mapping in SaR scenarios are available at `https://github.com/ethz-asl/segmatch`. A video demonstration is available at todo `https://www.youtube.com/watch?v=iddCgYbgpjE`

and could easily integrate *anchor nodes* instead if slow convergence is noticed during optimizations following inter-robot place recognitions.

III. System Architecture

This section introduces our multi-robot localization and mapping system based on 3D LiDAR and displacement measurements. As illustrated in Figure 2, the system is centralized such that a *master agent* is responsible of merging sensor information transmitted by multiple robots. At the core of the master agent lies an incremental pose-graph optimization back-end which is responsible for estimating the robot trajectories. On the other hand, the front-end is distributed between the robots and the master agent and is responsible for providing constraints to the optimization problem. For instance, each robot is responsible of computing sequential constraints, and of pre-processing the laser point clouds in order to compress the information to be transmitted over a communication channel of limited bandwidth.

The remainder of the section presents the back-end of the proposed multi-robot 3D SLAM system whereas Section IV details the front-end with the computation of LiDAR odometry, loop-closure and robot-association constraints.

A. Pose-graph formulation

The system is based on a pose-graph optimization approach [25] where a factor graph $G{=}(\mathcal{F}, \boldsymbol{\Theta}, \mathcal{E})$ is the underlying bipartite graph which connects all relevant elements of the system. It consists of factor nodes $\boldsymbol{f}_i{\in}\mathcal{F}$, variable nodes $\boldsymbol{\theta}_i{\in}\boldsymbol{\Theta}$ and edges $\epsilon_i{\in}\mathcal{E}$, that connect factor nodes with variable nodes. Variable nodes $\boldsymbol{\theta}_i$ are the states of the system and represent robot poses, i.e., $\boldsymbol{\theta}_i \in SE(3)$. Factor nodes $\boldsymbol{f}_i$ on the other hand are constraints between several poses.

The factor graph then defines the factorization of a function $\boldsymbol{f}(\boldsymbol{\Theta})$

$$\boldsymbol{f}(\boldsymbol{\Theta}) = \prod_i \boldsymbol{f}_i(\boldsymbol{\Theta}_i) \tag{1}$$

with $\boldsymbol{\Theta}_i$ being the subset of variables adjacent to the factor $\boldsymbol{f}_i$. Our system implements three different types of factor nodes, i.e., prior factors $\boldsymbol{f}_{prior}(\boldsymbol{\Theta}_0)$, sequential factors $\boldsymbol{f}_{seq,i}(\boldsymbol{\Theta}_i)$, and place recognition factors $\boldsymbol{f}_{PR,i}(\boldsymbol{\Theta}_i)$ which include both intra-robot loop-closures and inter-robot associations. Both $\boldsymbol{f}_{seq,i}(\boldsymbol{\Theta}_i)$ and $\boldsymbol{f}_{PR,i}(\boldsymbol{\Theta}_i)$ are always expressed as relative pose measurements which is particularly practical for multi-robot applications as the measurements are entirely independent from the fixed frame of reference. Finally, each factor is expressed in full 6 Degrees of Freedom (DOF).

B. Sparse incremental optimization

The aim of the back-end is to compute a Maximum A Posteriori (MAP) estimate of $\boldsymbol{f}(\boldsymbol{\Theta})$ given its observations $\tilde{\boldsymbol{z}}_i$ that minimizes a negative log-posterior $\boldsymbol{E}$. Assuming a Gaussian measurement model leads to Equations 2, 3 and 4.

$$\boldsymbol{f}_i(\boldsymbol{\Theta}_i) \propto \exp\left(-\frac{1}{2}||\boldsymbol{z}_i(\boldsymbol{\Theta}_i) - \tilde{\boldsymbol{z}}_i||^2_{\boldsymbol{\Omega}_i}\right) \tag{2}$$

$$\boldsymbol{E} = -\log \boldsymbol{f}(\boldsymbol{\Theta}) \tag{3}$$

$$\arg\min_{\boldsymbol{\Theta}}(\boldsymbol{E}) = \arg\min_{\boldsymbol{\Theta}}(\sum_i \boldsymbol{e}_i^T \boldsymbol{\Omega}_i \boldsymbol{e}_i) \tag{4}$$

where $e_i = \boldsymbol{z}_i(\boldsymbol{\Theta_i}) - \tilde{\boldsymbol{z}}_i$ is the error between the prediction function $\boldsymbol{z}_i(\boldsymbol{\Theta_i})$ and a measurement $\tilde{\boldsymbol{z}}_i$, with the information matrix $\boldsymbol{\Omega}_i$. In order to robustify the optimizer against false place recognitions, we add a Cauchy function as M-estimator to the place recognition factors as described in [26] which down-weights the effect of possibly wrong factors on the optimization objective $\boldsymbol{E}$.

As the prediction function $\boldsymbol{z}_i(\boldsymbol{\Theta})$ is nonlinear, we minimize the error $\boldsymbol{E}$ using nonlinear optimization with the Gauss-Newton algorithm. Specifically, we perform incremental update and optimization of the pose-graph using the iSAM2 algorithm [14] which allows for efficient variable re-ordering and relinearization using the Bayes tree.

Given that we use *weak links*, introduced by Konolige and Bowman [24], through the prior factor $\boldsymbol{f}_{prior}(\boldsymbol{\theta}_0)$, the Bayes tree approach is particularly suitable for updating the pose-graph resulting from the multi-robot problem. Once an inter-robot place recognition is detected, one prior is removed from the graph and iSAM2 allows for efficient re-ordering and relinearization of the variables.

IV. System front-end

In this section, we show one possible front-end configuration where sequential factors are created using ICP and displacement measurements and where the *SegMatch* algorithm is used for generating place recognition factors. As illustrated in the block diagram of Figure 2 the system is modular so that one could select and integrate different techniques for factors generation modules.

A. Sequential factors

This first front-end module is responsible for transforming 3D LiDAR and proprioceptive sensors measurements into sequential factors $\boldsymbol{f}_{seq,i}(\boldsymbol{\Theta}_i)$. In order to keep this module robust to failures when doing scan matching, we separate its contribution to the graph in two different types of factors i.e., odometry factors $\boldsymbol{f}_{odom,i}(\boldsymbol{\Theta}_i)$ and scan-matching factors $\boldsymbol{f}_{scan,i}(\boldsymbol{\Theta}_i)$.

A pose node $\boldsymbol{\theta}_i$ is added to the graph, along with one odometry and one scan-matching factor for every 3D laser-scan S_i, given that the robot travelled a minimum distance d_{min}. This minimum travel distance is a standard practice to help avoiding un-informative accumulation of data, and the growth of the factor graph, when the robot is not moving.

1) Odometry factors: Odometry factors $\boldsymbol{f}_{odom,i}(\boldsymbol{\Theta}_i)$ are based on displacements between consecutive robot poses as stated in Eq. 5 and are built using Eq. 2.

$$\boldsymbol{z}_{odom,i}(\boldsymbol{\Theta}_i) = \boldsymbol{\theta}_{i-1}^{-1} \oplus \boldsymbol{\theta}_i \tag{5}$$

These odometry measurements can be obtained from different sources of proprioceptive sensors. For instance, for the UGVs illustrated in Figure 1, these constraints are obtained fusing wheel encoders and IMU data using an extended Kalman filter as proposed by Kubelka et al. [27].

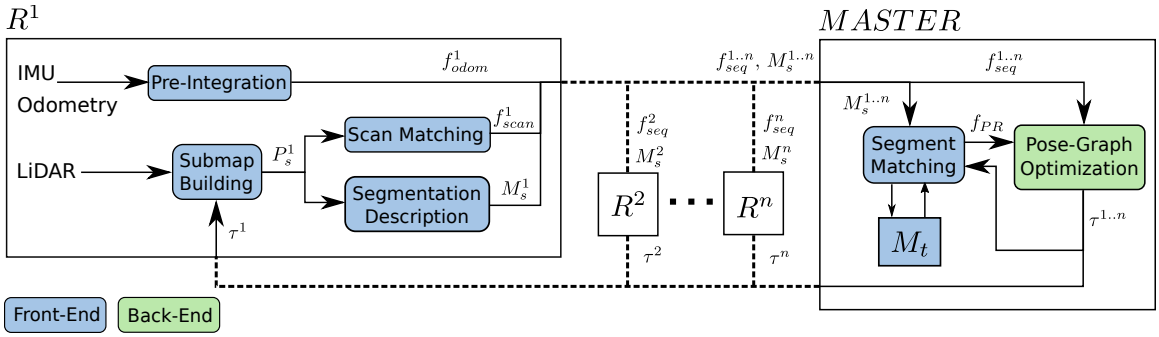


Fig. 2: A configuration of the proposed centralized multi-robot SLAM system where place recognition is based on a segment matching algorithm. The robots R^i locally compute odometry and scan-matching factors f^i_{odom} and f^i_{scan}. The 3D source point clouds P^i_s are segmented into maps M^i_s which are used by the master for generating place recognition factors f_{PR}. Once the pose-graph is optimized by the master, the trajectories updates τ^i are used to maintain the target map M_t and transferred back to the robots. For simplicity, modules acting locally on each robot have been expanded only once.

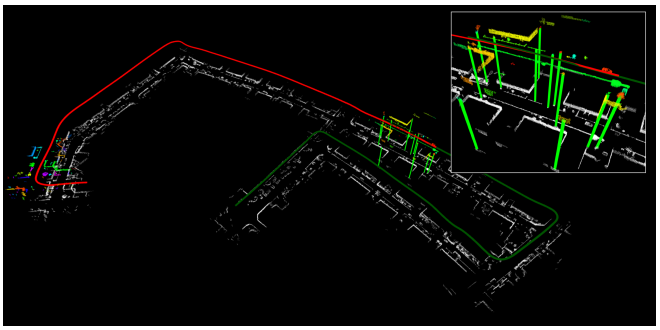
2) Scan matching factors: The 3D LiDAR data is used to compute scan-matching factors $\boldsymbol{f}_{scan,i}(\boldsymbol{\Theta}_i)$ between adjacent nodes in the graph. These factors are obtained using ICP by registering the current scan against a *submap* composed of the m previous scans, expressed in the frame of the previous pose. The *submap* size is controlled by m, the number of previous scans and d_{min}, the minimum travelled distance between consecutive scans.

Performing ICP against this *submap* helps dealing with sparsity in the scans, eg. when working with Velodyne data, and helps making the system more robust. The ICP registration step between the *submap* and the new scan results in the 6 DOF rigid transformation ${}_iT_{ij}$ which is a transformation from pose θ_i to θ_j expressed in the frame of θ_i. This transformation is directly used to build a laser scan-matching factor $\boldsymbol{f}_{scan,i}(\boldsymbol{\Theta}_i)$ using the distance between the relative transformation prediction and measurement as an error function, i.e. $e = d({}_iT_{ij}, {}_i\tilde{T}_{ij})$.

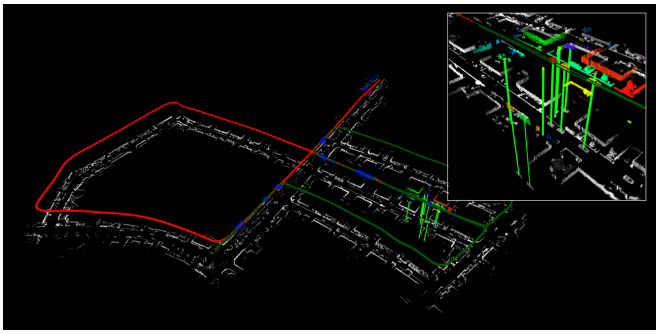
B. Place recognition factors

Our multi-robot SLAM system has a flexible implementation which can receive place recognition candidates from different sources. In the present work, we propose one configuration where associations are generated using the *SegMatch* algorithm introduced in [11]. This algorithm takes as input (i) a source point cloud P_s representing local LiDAR measurements and (ii) a target point cloud P_t against which place recognition is performed. Segments are extracted and described from both P_s and P_t to respectively generate a source map M_s and a target map M_t which contain lists of low dimensional segments descriptors. In this work we consider two region growing algorithms for extracting segments from 3D point clouds: one based on Euclidean distance [11] and the other based on smoothness constraints [28]. Compact eigenvalue based features are used for describing the segments [11]. Associations are made between M_s and M_t and a geometrical verification step is used to identify clusters of matches which represent place recognitions as illustrated in Figure 3.

The remainder of this section will detail how segment matching is used within the full system and the reader is encouraged to consult our prior work for more information about the *SegMatch* algorithm [11].



(a) First inter-vehicle association.



(b) Last intra-vehicle association.

Fig. 3: Two autonomous vehicles mapping KITTI sequence 05 using LiDAR information only. The trajectories are shown in red and green and the source map segments are shown in colors. The optimized associations are show in blue with the latest association shown by segment matches with vertical green lines. The target map M_t is shown in white below.

1) Source map generation: As illustrated in Figure 2, each robot in the system is responsible of generating its own segment source map M^j_s where j denotes the robot's unique identifier. M^j_s is, along with the sequential factors $\boldsymbol{f}^j_{odom,i}(\boldsymbol{\Theta}_i)$ and $\boldsymbol{f}^j_{scan,i}(\boldsymbol{\Theta}_i)$, the only information communicated to the centralized master. Converting the raw point cloud P^j_s into M^j_s induces a high level of compression which is a key advantage for reducing the required communication bandwidth in multi-robot systems.

The intermediate source point cloud representation P^j_s is created by accumulating the 3D scans S_i once the corresponding nodes $\boldsymbol{\theta}_i$ are optimized by the back-end. Noisy

data is filtered using a voxel grid of resolution res_{voxel} with n_{min}, a minimum number of points per voxel to consider them as occupied. An octomap [29] can also be used if one further wishes to filter dynamics. The growth of P_s^j is limited by extracting a cylindrical neighbourhood of radius R, centred around the current robot location. Applying this cylindrical filter inevitably results in cut objects, which then results in 'incomplete segments' in M_s^j that can interfere with 'complete views' in the target map M_t. These 'incomplete segments' are detected by filtering P_s^j with a smaller radius $r = R - b$, where b is the thickness of the outer zone. Segments containing points within that zone are discarded from M_s^j which is transferred to the master and used for both matching and building the target map M_t.

2) Incremental target map management: Given our centralized approach, the master agent is responsible of incorporating incoming source maps M_s^j into a single target map representation M_t. For each segment in M_s^j, we check for a duplicate in M_t, i.e. a segment resulting from the same object part, but extracted at different times. As single-robot odometry is locally accurate, these 'duplicate segments' can efficiently be detected by comparing the distances to the closest segments centroids in M_t with a minimum distance d_{seg}. As we prefer to keep the latest view of a segment, we choose to remove the oldest of these duplicates. Future work will include techniques for merging these 'duplicate segments'.

In event of place recognitions, M_t is updated with a similar approach. Given the updated robot trajectories, the positions of the target segments are first refreshed, knowing the origin of their segmentation relative to the trajectories. In case of successful place recognitions, segments of the target map will correctly align and can safely be filtered for removing duplicates as described above.

3) Place recognition factor generation: Given M_s and M_t, segment matching is performed through k-Nearest Neighbors (k-NN) retrieval. Afterwards, a geometric verification step based on RANSAC identifies clusters containing at least min_{RANSAC} segment matches that are geometrically consistent with a resolution res_{RANSAC}. In order to convert these segment matches in the form of place recognition factors $\boldsymbol{f}_{PR,i}(\boldsymbol{\Theta}_i)$, the nodes to be constrained by the factors are first identified. For both the source and the target, we select the trajectory node that is, on average, the closest to all corresponding segments and which lies within a time window defined by the segments' timestamp.

As the segments centroids are represented in world frame when doing geometrical verification, the resulting relative transformation will also be given in world frame, i.e. ${}_wT_{ij}$. This transformation can be expressed in the frame of the first node θ_i using Eq. 6 and converted into a factor as described in Section IV-A.

$$ {}_iT_{ij} = {\theta_i}^{-1} \oplus_w T_{ij} \oplus \theta_j \tag{6} $$

Given these new factors, the pose-graph is incrementally optimized and M_t is updated as explained in IV-B.2. As shown in Figure 2, the resulting trajectory updates τ^j are transmitted back to the individual robots which then update P_s^j to follow the transformation applied to the trajectory head. The performances of this *SegMatch* based place-recognition module are evaluated in the following section.

V. EXPERIMENTS

In this section we demonstrate the performance of the proposed system through two different multi-robot experiments. In the first experiment, we adapt sequence 05 of the KITTI odometry dataset [30] in order to generate a multi-robot scenario. The second experiment is based on data collected during a SaR mission performed by the "Long-Term Human-Robot Teaming for Robots Assisted Disaster Response" (TRADR) consortium[2] at the Gustav Knepper Power Station in Dortmund, Germany.

A. Implementation details

This section briefly presents implementation details which can be relevant when exploring the system. The system is built on multiple available libraries, as for instance, the incremental optimization back-end which is based on the iSAM2 implementation of the GTSAM library[3]. The factor's Jacobians are evaluated using the block automatic differentiation functionality of GTSAM. For creating scan-matching factors we use the ICP implementation of $libpointmatcher$[4]. The place recognition implementation is based on the *SegMatch* library[5] which itself uses PCL[6] for voxel grid filtering and geometric verification functionalities and $libnabo$ [31] for segment matching with fast k-NN search in low dimensional space. The system has a full ROS interface with integration to the TF tree for publishing the estimated robot poses.

B. Experiments setup

The two following experiments are performed on a single computer equipped with an Intel i7-4900MQ CPU @ 2.80GHz and 32 GB of DDR3 RAM. In order to realize the multi-robot scenarios, the system is implemented with multiple threads. One thread per robot is used for computing the sequential factors and the local maps whereas the place recognition, pose-graph optimization and target map management functionalities are all running on a separate thread. For real missions, the latter thread would run on the master agent and the computational load would be distributed amongst the multiple computers.

C. System parametrization

The front-end parameters used for both experiments are resumed in Table I. For the place recognition module, segment retrieval is always performed using k-NN. However, different segmentation algorithms and segment descriptors are used which reflects the difference in sensor configuration, vehicle

[2] http://www.tradr-project.eu/
[3] https://research.cc.gatech.edu/borg/gtsam
[4] https://github.com/ethz-asl/libpointmatcher
[5] https://github.com/ethz-asl/segmatch
[6] http://pointclouds.org/

TABLE I: Front-end parameters for the two experiments.

Parameter	KITTI	Power plant
Min. distance between poses (d_{min})	0.05 meter	0.1 meter
Number of scans per *submap* (m)	5	10
Voxel grid resolution (res_{voxel})	0.1 meter	0.1 meter
Min. point count per voxel (n_{min})	1	2
Source map radius (R)	60 meters	25 meters
Number of neighbors (k)	45	20
Min. RANSAC cluster size (min_{RANSAC})	5	5
RANSAC resolution (res_{RANSAC})	0.45 meter	0.55 meter
Min. segment distance (d_{seg})	2 meters	2 meters

TABLE II: Timing of each module (in ms).

Module (Agent)	KITTI	Power plant
Scan-matching (R)	78.9	916.8
Trajectories estimation (M)	3.7	4.3
Trajectories estimation after PR (M)	43.1	24.9
Segmentation and description (R)	600.1	1176.0
Segment matching (M)	8.1	6.0
Geometric verification (M)	57.2	13.8
Duplicates removal (M)	91.8	21.2

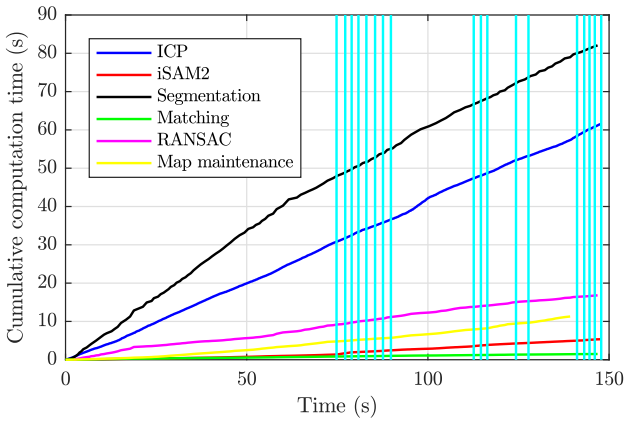


Fig. 4: Cumulative computation time for each module during the LiDAR-only multi-robot KITTI experiment. The 18 valid place recognitions are shown with vertical lines. During this experiment, no false place recognition was detected.

speed and environment between the two experiments. For the multi-robot KITTI experiment, Euclidean segmentation and eigenvalue based features are adopted, as described in [11]. For the power plant experiment, segments are obtained based on region growing with smoothness constraints as described in [28]. Normals are computed with a radius of 0.5 meters and the 15 nearest neighbours of each seed are considered for growing segments. The threshold on the difference of normals is set to 8 degrees and we keep only segments having a minimum of 75 points. In addition to the eigenvalue based features, we use the fact that the power plant environment contains many vertical and horizontal planar surfaces and make a distinction between these by adding a single value feature describing its orientation. This can efficiently be computed by comparing the x, y, and z dimensions of the segments.

For the two platforms considered in the experiments, the noise models could easily be adjusted on different datasets in order to yield good localization results.

D. LiDAR-only multi-vehicular KITTI

In this first experiment, we split sequence 05 of the KITTI odometry dataset which lasts 2.2 km and 287 seconds into two sequences of equal duration. A multi-robot scenario is simulated by simultaneously playing back the two sequences.

With this experiment, we aim to show the capabilities of our system for performing online multi-robot SLAM by using LiDAR information only. In other words, the scan-matching factors $\boldsymbol{f}_{scan,i}(\boldsymbol{\Theta}_i)$ are the only sequential factors considered in this experiment. A constant velocity model is adopted for generating initial guesses to the ICP based registrations, acting as measurement functions for these factors. The rest of the parametrization is introduced in V-C.

On this multi-robot sequence and using LiDAR measurements only, our system could in real-time detect 18 valid intra and inter-robot global associations. The first and last associations are illustrated in Figure 3 along with the two estimated trajectories. The timings for each module are stated in Table II with an indication whether this module should be executed by the robot (R) or by the master agent (M). Cumulative computation times are given in Figure 4.

During this experiment, 154 source point clouds P_s^i were processed for place recognition and contained in average 96200 points (after downsampling, voxelization and cylindrical filtering). With each point defined by three doubles and assuming a typical size of 8 bytes per double, communicating the source clouds to the master computer would require the transmission of 356 MB of data. In comparison, 5MB would be transmitted if one would select 100 keypoints per source cloud and describe these with Fast Point Feature Histograms (FPFH) [32] of dimension 1x33. With our segment approach, the source maps M_s^i contained in average 30.4 $\pm$ 9.9 descriptors for a total of 4682 descriptors transmitted to the master computer. With the compact representation of eigenvalue based features [1x7], this results in only 562 kB of data to be transmitted over 143 seconds of operation. Note that for these computations, six doubles and two unsigned int are additionally required to link each descriptor to the trajectories. We also only treat the *useful data* and do not consider the data transfer overhead.

E. Gustav Knepper Power Station

For this experiment, we use data collected during a SaR mission performed by the TRADR consortium at the decommissioned Gustav Knepper Power Station. The experiment took place in one large two-floors utility building measuring 100m long by 25m wide illustrated in Figure 1. During the exercise held in November 2016, three UGVs equipped with multiple encoders, an Xsens MTI-G IMU and a rotating 2D SICK LMS-151 LiDAR were teleoperated by firemen end-users in order to efficiently explore the scenario. With their skid-steering climbing capabilities, the TRADR UGVs can traverse and map challenging 3D environments.

The power plant mission lasted 950 seconds with the three

robots starting at different locations and traversing different paths with a cumulative distance of 694 meters. As can be seen in Figure 1, UGV_{green} began its mission outside, entered the building and performed a loop in clock-wise direction of the ground floor. Simultaneously, UGV_{red} first climbed challenging metal stairs at the right side of the ground floor in order to reach and explore the upper level whereas UGV_{yellow} started at the left side of the upper floor, went down and explored the ground floor.

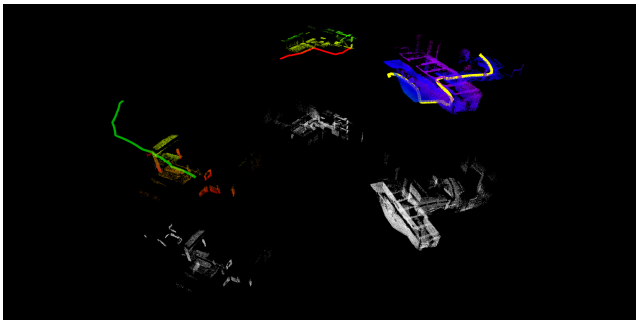
For this experiment, the odometry factors $\boldsymbol{f}_{odom,i}(\boldsymbol{\Theta}_i)$ are obtained by fusing encoders and IMU measurements with the technique presented by Kubelka et al. [27]. This odometry information is also used for accumulating measurements from the rotating 2D LiDAR sensor into dense 3D point clouds which are used to compute the scan-matching factors $\boldsymbol{f}_{scan,i}(\boldsymbol{\Theta}_i)$ as described in Section IV-A.2. The parametrization of the other modules is defined in Section V-C and Table I.

During the three-robot power plant experiment, our SLAM system with the presented configuration was capable of detecting 20 valid place recognitions, in real-time and on one single computer. Eight of these successful *SegMatch* detections were in challenging scenarios where the UGVs drove in opposite directions. The three trajectories, as estimated at different moments of the mission, are depicted in Figure 5. Figures 5b and 5c specifically illustrate the case where a global prior factor $\boldsymbol{f}_{prior}(\boldsymbol{\Theta}_0)$ is removed from the pose-graph due to a first inter-robot association, as described in Section III-B. Two of the eight place recognitions in areas visited in opposite directions are shown in Figure 5c and 5d. The final trajectories are illustrated in Figure 1 and a top-down view is given in Figure 6. As for the previous experiment, the timings are stated in Table II.

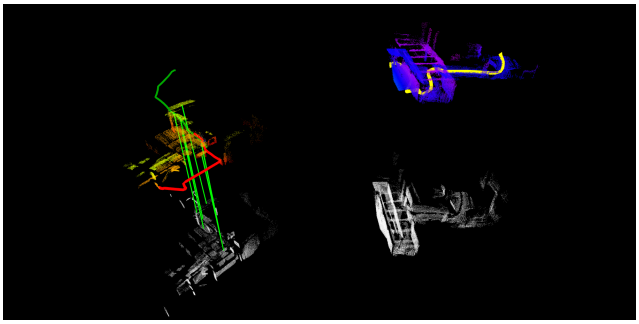
VI. CONCLUSION

This paper presented a SLAM system for multiple robots equipped with 3D LiDAR sensors. After important development efforts, our system successfully integrates different state of the art modules that are advantageous for multi-robot systems with regards to closing loops, computational complexity, and communication bandwidth requirements.

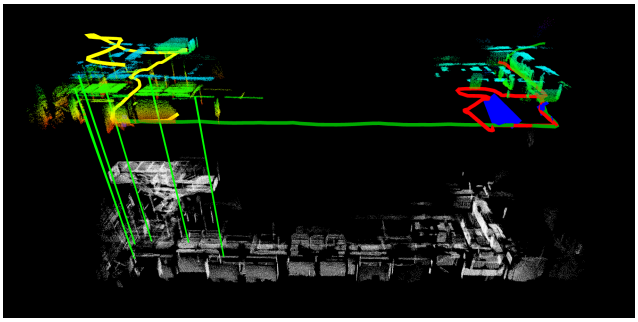
The system's back-end is based on a pose-graph optimization approach where updates and inference are performed incrementally using the Bayes tree. We showed how this architecture can easily be configured to handle multiple trajectories without any prior information on their relative position. The front-end is responsible of providing the graphical model with LiDAR odometry and place recognition constraints. The LiDAR odometry constraints relate successive nodes using ICP scan-matching between the latest scan and a *submap* of previous scans. Place recognition is performed by leveraging a segment extraction and matching algorithm. We demonstrated through two experiments in different scenarios that the presented system enabled multiple-robots to jointly map large areas in real-time and with a high rate of successful place recognitions. During these experiments, we found that



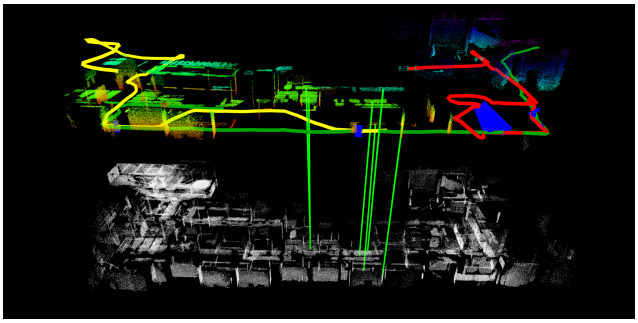
(a) Start of the mission with each robot in its own frame of reference.



(b) At $t = 180s$, first global association between UGV_{red} and UGV_{green}.



(c) At $t = 380s$, first encounter of UGV_{green} and UGV_{yellow} when exploring the lower floor in opposite directions.



(d) At $t = 476s$, another association is made between the same UGVs in a corridor traversed in opposite directions.

Fig. 5: A demonstration of our multi-robot SLAM system based on data collected in a two-floor building of the Gustav Knepper Power Station in Dortmund, Germany. The trajectories of UGV_{red}, UGV_{green} and UGV_{yellow} are estimated in real-time. The vertical green lines represent segment matches which resulted in one of the 20 inter-robot place recognitions. For visualization purposes, the source maps are coloured by height and the target map is illustrated in white below. The final trajectories are illustrated in Figure 1. The reader is encouraged to consult the video demonstration for a better visualization.

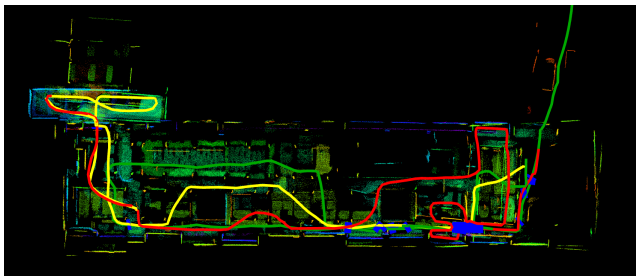


Fig. 6: A top down view of the multi-robot experiment in the Gustav Knepper Power Station. The target map M_t is coloured by height.

identifying valid and accurate place recognitions is a crucial capability when dealing with multiple robots.

In the context of multi-robot systems acting in difficult scenarios and under limited communication bandwidth, one important challenge is to limit the data to be transmitted for finding robot associations [3]. One key advantage of the proposed system is to detect these associations on the basis of segments that offer a high level of descriptiveness and information compression.

Whereas this paper proposed a centralized system, future work could include a distributed system which would also directly benefit from these advantages. To this end, the implementation of the complete system is available online with easy to run demonstrations, along with the new SaR dataset introduced in this work.

References

[1] D. Tardioli, D. Sicignano *et al.*, "Robot teams for intervention in confined and structured environments," *Journal of Field Robotics*, 2015.

[2] I. Kruijff-Korbayová, F. Colas *et al.*, "Tradr project: Long-term human-robot teaming for robot assisted disaster response," *KI-Künstliche Intelligenz*, vol. 29, no. 2, pp. 193–201, 2015.

[3] S. Saeedi, M. Trentini *et al.*, "Multiple-robot simultaneous localization and mapping: A review," *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.

[4] S. Lowry, N. Sunderhauf *et al.*, "Visual place recognition: A survey," *IEEE Trans. on Robotics*, 2016.

[5] K. Nagatani, Y. Okada *et al.*, "Multirobot exploration for search and rescue missions: A report on map building in robocuprescue 2009," *Journal of Field Robotics*, vol. 28, no. 3, pp. 373–387, 2011.

[6] N. Michael, S. Shen *et al.*, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.

[7] M. Bosse and R. Zlot, "Place recognition using keypoint voting in large 3D lidar datasets," in *IEEE Int. Conf. on Robotics and Automation*, 2013.

[8] Y. Zhuang, N. Jiang *et al.*, "3-d-laser-based scene measurement and place recognition for mobile robots in dynamic indoor environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 2, pp. 438–450, 2013.

[9] T. Röhling, J. Mack, and D. Schulz, "A fast histogram-based similarity measure for detecting loop closures in 3-d lidar data," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015.

[10] K. Granström, T. B. Schön *et al.*, "Learning to close loops from range data," *The Int. Journal of Robotics Research*, vol. 30, no. 14, pp. 1728–1754, 2011.

[11] R. Dubé, D. Dugas *et al.*, "Segmatch: Segment based place recognition in 3d point clouds," in *IEEE Int. Conf. on Robotics and Automation*, 2017.

[12] C. Cadena, L. Carlone *et al.*, "Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age," *IEEE Trans. on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[13] F. Lu and E. Milios, "Globally Consistent Range Scan Alignment for Environment Mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.

[14] M. Kaess, H. Johannsson *et al.*, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The Int. Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[15] K. Pathak, A. Birk *et al.*, "Online three-dimensional slam by registration of large planar surface segments and closed-form pose-graph relaxation," *Journal of Field Robotics*, vol. 27, no. 1, pp. 52–84, 2010.

[16] D. Droeschel, M. Schwarz, and S. Behnke, "Continuous mapping and localization for autonomous navigation in rough terrain using a 3d laser scanner," *Robotics and Autonomous Systems*, 2016.

[17] E. Mendes, P. Koch, and S. Lacroix, "Icp-based pose-graph slam," in *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 195–200.

[18] D. Borrmann, J. Elseberg *et al.*, "Globally consistent 3d mapping with scan matching," *Robotics and Autonomous Systems*, vol. 56, no. 2, pp. 130–142, 2008.

[19] A. Nüchter, K. Lingemann *et al.*, "6d slam—3d mapping outdoor environments," *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 699–722, 2007.

[20] R. Kurazume, S. Oshima *et al.*, "Automatic large-scale three dimensional modeling using cooperative multiple robots," *Computer Vision and Image Understanding*, 2016.

[21] B. Kim, M. Kaess *et al.*, "Multiple relative pose graphs for robust cooperative mapping," in *IEEE Int. Conf. on Robotics and Automation*, 2010.

[22] K. Ni, D. Steedly, and F. Dellaert, "Tectonic sam: Exact, out-of-core, submap-based slam," in *IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 1678–1685.

[23] J. McDonald, M. Kaess *et al.*, "Real-time 6-dof multi-session visual slam over large-scale environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1144–1158, 2013.

[24] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 1156–1163.

[25] G. Grisetti, R. Kümmerle *et al.*, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[26] G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Robust pose-graph loop-closures with expectation-maximization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 556–563.

[27] V. Kubelka, L. Oswald *et al.*, "Robust data fusion of multimodal sensory information for mobile robots," *Journal of Field Robotics*, vol. 32, no. 4, pp. 447–473, 2015.

[28] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 5, pp. 248–253, 2006.

[29] A. Hornung, K. M. Wurm *et al.*, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[30] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[31] J. Elseberg, S. Magnenat *et al.*, "Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration," *Journal of Software Engineering for*

Robotics, vol. 3, no. 1, pp. 2–12, 2012.

[32] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 3212–3217.