

# Multi-Robot 6D Graph SLAM Connecting Decoupled Local Reference Filters

Martin J. Schuster<sup>1</sup>, Christoph Brand<sup>1</sup>, Heiko Hirschmüller<sup>1</sup>, Michael Suppa<sup>1</sup> and Michael Beetz<sup>2</sup>

**Abstract**—Teams of mobile robots can be deployed in search and rescue missions to explore previously unknown environments. Methods for joint localization and mapping constitute the basis for (semi-)autonomous cooperative action, in particular when navigating in GPS-denied areas. As communication losses may occur, a decentralized solution is required. With these challenges in mind, we designed a submap-based SLAM system that relies on inertial measurements and stereo-vision to create multi-robot dense 3D maps. For online pose and map estimation, we integrate the results of keyframe-based local reference filters through incremental graph SLAM. To the best of our knowledge, we are the first to combine these two methods to benefit from their particular advantages for 6D multi-robot localization and mapping: Local reference filters on each robot provide real-time, long-term stable state estimates that are required for stabilization, control and fast obstacle avoidance, whereas online graph optimization provides global multi-robot pose and map estimates needed for cooperative planning. We propose a novel graph topology for a decoupled integration of local filter estimates from multiple robots into a SLAM graph according to the filters' uncertainty estimates and independence assumptions and evaluated its benefits on two different robots in indoor, outdoor and mixed scenarios. Further, we performed two extended experiments in a multi-robot setup to evaluate the full SLAM system, including visual robot detections and submap matches as inter-robot loop closure constraints.

## I. INTRODUCTION

Search and rescue (SAR) missions following up disaster events often take place in environments that are hard or dangerous to access for humans. Teams of mobile robots can be deployed to raise the rescue workers' situational awareness while keeping them out of danger. These robots have to operate in previously unknown semi- or unstructured environments where external methods for localization like (D)GPS are not always available. In order to operate in such challenging environments, local reference filters can provide real-time local state estimates on each system to allow for fast obstacle avoidance as well as the stabilization of highly dynamic systems. Our novel SLAM framework connects these estimates for global joint localization and mapping, which acts as the basis for coordinated cooperative actions within a team of robots. We therefore designed a decentralized solution and perform all computation online and on board the individual systems in order to gain robustness w.r.t.

<sup>1</sup> German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Department of Perception and Cognition, Münchner Str. 20, 82234 Wessling, Germany, {martin.schuster|christoph.brand|heiko.hirschmueller|michael.suppa}@dlr.de

<sup>2</sup> Institute for Artificial Intelligence and TZI (Center for Computing Technologies), University Bremen, Am Fallturm 1, 28359 Bremen, Germany, beetz@cs.uni-bremen.de

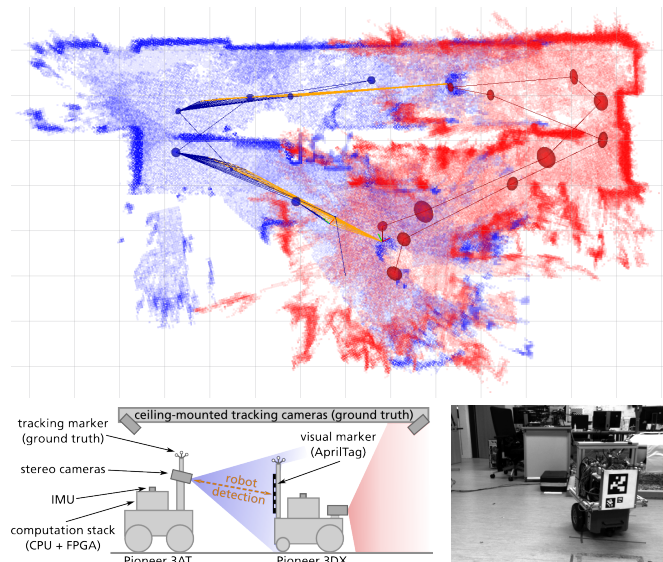


Fig. 1. Top: Multi-Robot map and graph for an indoor scenario with robot detections as inter-robot loop closure constraints. Bottom: Experimental setup and camera image of P3DX as seen by P3AT.

communication losses and failures of individual robots. Our robots are equipped with stereo cameras and we employ *semi-global stereo matching (SGM)* [1] to gather dense 3D data under varying light conditions in both indoor and outdoor environments. We follow a submapping approach to create a dense multi-robot 3D map and apply graph SLAM to both optimize the map as well as the 6D pose estimates for all participating robots. In Figure 1, we present the joint 3D map created by two rovers equipped with narrow- and wide-angle stereo camera systems and provide a sketch of our experimental setup. We exchange submaps between robots and integrate marker-based visual robot detections as well as submap matches to obtain intra- and inter-robot loop closure constraints. The contributions of this work are twofold: First, we propose a decoupled integration of keyframe-based local reference filters and online multi-robot graph SLAM with incremental optimization. To the best of our knowledge, we are the first to combine these two methods to benefit from their particular advantages for 6D multi-robot localization and mapping: Local reference filters on each robot provide real-time, long-term stable state estimates [2] that are required for stabilization, control and fast obstacle avoidance, whereas online graph optimization provides global multi-robot pose and map estimates needed for cooperative planning. We propose a novel graph topology for a decoupled integration according to the filters' uncertainty estimates and inde-

pendence assumptions. Second, we integrate the filter and SLAM components into our modular mapping architecture, combining them with visual robot detections and submap matching based on geometric features. This allows matches between robots with different camera systems to generate loop closure constraints in heterogeneous multi-robot teams. They are all accompanied by Gaussian uncertainty estimates, which we take into account during optimization.

We present two types of evaluations. First, we evaluate the impact of our novel integration of local reference filter estimates, showing an average improvement on localization accuracy of 15 % in three experiments performed in indoor, unstructured outdoor and mixed scenarios on two different robots. Second, we conducted two extended multi-robot mapping experiments in an indoor scenario with unknown initial robot poses. We evaluated the full SLAM system, including visual robot detections and submap matches as inter-robot loop closure constraints between two robots equipped with different stereo camera systems. We achieved an improvement for the joint localization accuracy compared to single-robot SLAM with submap matching of on average 32 %.

## II. RELATED WORK

Within the large body of related work on simultaneous localization and mapping (SLAM), three fundamental techniques can be identified: Extended Kalman Filters (EKF), Rao-Blackwellized particle filters (RBPF), and graph optimization. In EKFs, landmark-based maps are modeled as multivariate Gaussians, typically implying a quadratic growth of the computational effort with the number of landmarks [3]. RBFs tracking a multi-modal distribution over robot trajectories accompanied by grid maps are often employed for 2D mapping [4]. As the number of particles required for stable and robust estimation grows exponentially with the size of the state space [5], RBFs are computationally challenging for 6D SLAM [6]. While RBPF approaches exist for multi-robot 2D mapping, they pose additional restrictions, e.g. limiting information exchange between robots to rendezvous events [7] or requiring similar robot moving speeds to avoid particle depletion [8]. RBFs are thus not well suited for multi-robot 6D SLAM. Graph SLAM gained popularity since recent advances in sparse nonlinear incremental optimization [9] allow its application for online localization and mapping. A graph represents 6D poses, connected by measurements with Gaussian uncertainties. On loop closures, non-linear least-squares optimization is applied to reduce the overall error. Its worst-case computational effort depends on the size and complexity of the graph and thus typically grows with the traveled distance. This challenge can be approached by constraining the optimization to local regions [10] or removing nodes through marginalization [11]. We consider graph optimization to be the most promising technique for 6D multi-robot SLAM. Inter-robot measurements can easily be integrated as constraints connecting multiple subgraphs while keeping the complexity of the optimization manageable [12].

The creation of submaps is a technique to handle large quantities of sensor data, e.g. dense 3D information, by

locally aggregating them into multiple maps of limited size [13] [14] and adding their origins as nodes to the slam graph. The sparse structure of the resulting graph allows for efficient optimization steps while the submaps themselves can keep more sensor information than key-frame based systems that sparsely sample the incoming sensor data [15]. Submapping is in particular suitable for multi-robot systems, as under communication constraints, an exchange of submaps is more viable than an exchange of the complete raw sensor data.

The integration of filter and graph SLAM methods allows to fulfill both local real-time requirements as well as to provide global multi-robot estimates. An adequate integration of filter estimates instead of sequential odometry measurements, however, requires an adaption of the graph topology, as we discuss in this work. Leishman et al. [15] combine a keyframe-based local reference filter with a pose graph for global localization, representing all keyframes as nodes in the graph. In contrast to their work, we decouple the graph from internal filter states and thus can trigger the creation of new local reference frames from the higher-level mapping modules. This allows us to keep the graph size independent of the number of keyframes, which can be orders of magnitude larger than the number of submaps required for mapping. Furthermore, Leishman et al. track only one keyframe at a time, did not yet incorporate loop closure constraints and solely consider single-robot setups. Williams et al. [11] split a graph that contains all measurements into a real-time filter for the most recent data and a slower smoother for past states. Running these in parallel, they are able to recover the solution of full batch optimization. This however requires a tight coupling between filter and smoother, which exchange state information in both directions. While both work on the same types of sensor data, we propose to process measurements at different levels of abstraction in our filter and SLAM by solely adding aggregated pose information to the graph. As Williams et al. [11] feed loop closure results back into their filter, they cannot guarantee it to produce a smooth trajectory, which however is required for stabilization and control of highly dynamic systems.

We are aware of several multi-robot graph SLAM approaches [16] [14] [13] [17], however each of them has its own limitations. They either assume unlimited communication [16], restrict the enforcement of loop closures [14], require a master node to decide on inconsistencies [13] or solely provide simulation results created under simplifying assumptions [17]. Both Kim et al. [16] and Cunningham et al. [17] connect pose graphs created by multiple robots through nodes specifying frame-of-reference constraints, the latter additionally introducing anti-factors to exchange condensed graphs between robots while avoiding double counting of information under delayed communication. With our proposed system, double counting cannot occur as each robot constructs its graph separately, thereby adding each measurement only once. As our combination of local reference filters and submapping keeps the graph size small in the first place, we do not expect major benefits from exchanging optimized partial graphs.

### III. MULTI-ROBOT MAPPING ARCHITECTURE

We present an overview of our software architecture as a block diagram in Figure 2. It shows the components and data flows within our localization and mapping framework as well as their connections to a robot’s sensors and further software modules. We employ ROS as a middleware to connect the individual software components.

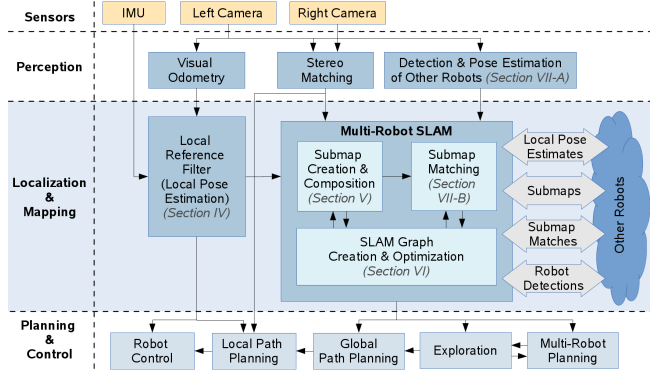


Fig. 2. Software architecture block diagram with four layers

For the experiments presented in this work, our sensor setup solely consists of a pair of stereo cameras as well as an inertial measurement unit (IMU). The local reference filter as well as our graph-based SLAM system however can easily be extended to include measurements from further sensors, e.g. barometer-based height updates or GPS fixes where available. The perception layer contains dense stereo matching [1], visual odometry computation [18] as well as marker-based detection of other robots and estimation of their poses [19]. For both visual odometry and robot detections, we compute estimates for the measurement uncertainty and pass them to the subsequent filter and SLAM components.

Our focus lies on the localization and mapping layer. It consists of two major components. First, IMU and visual odometry data are fused in a local reference filter [2], which takes the respective measurement uncertainties into account. Second, our SLAM framework, consisting of modules for incremental SLAM graph creation and optimization, submap creation and composition as well as submap matching. These modules run in a decentralized fashion on-board all robots within our multi-robot system and exchange measurement and submap information whenever communication links between robots can be established. Thus, losses of communication are allowed to occur at any time, the only impact being a delay of the integration of measurements until the link is re-established. Furthermore, no initial knowledge about the robots’ starting positions is required. Each robot constructs and optimizes its own SLAM graph as we do not introduce a shared global coordinate frame. However, each robot includes the other robots’ pose estimates and submap data within its own environment model as soon as connections between them are created through measurements.

In the planning and control layer shown in Figure 2, we illustrate exemplary uses of the local and global pose and map estimates by further subsystems. The local reference

filter provides hard real-time local state estimates that can directly be utilized for control, including the stabilization of highly dynamic systems like quadrotors [20]. Together with the stereo data, fast local obstacle avoidance can be realized as well. We in particular do not feed back data from the SLAM into the real-time filter in order to keep this system-critical component decoupled from higher-level modules. As our SLAM components run online but at a slower rate, their output is suitable for global path and exploration planning. In a multi-robot context, they provide pose estimates for all robots in addition to a joint map, thus laying the basis for coordinating cooperative actions.

### IV. LOCAL REFERENCE FILTER

For hard real-time local state estimation, we employ an Extended Kalman Filter (EKF) with time-delay compensation for visual odometry measurements [20] as a local reference filter [2]. Internally, a fixed number of  $n = 5$  keyframes are used to improve its precision and allow for locally drift-free estimation when the robot is standing or hovering in place. Due to its implementation as a square root UD filter, it can also run on low-end processing boards with single-precision FPUs, as often used on lightweight micro aerial vehicles. These might run the filter for fast local estimation on-board, while global estimates can be computed by more powerful systems. The local reference filter can be triggered to switch its frame of reference at arbitrary points in time without losing keyframe information. By switching into a robot’s current pose, the local pose uncertainty estimates are reset to zero. In contrast to a full restart of the filter, other state variables like velocities and past keyframes are kept and transformed into the new frame of reference [2]. The benefits of local reference filters are threefold. First, frame switches allow unbounded covariances for unobservable states within the filter to be reset in a consistent way. This is important to guarantee long-term numerical stability. Second, unbounded errors violate assumptions on small-angle approximations that are performed during filter-internal linearizations and thus can lead to inconsistencies. Third, local reference frames anchored in our submaps allow for a better integration of the filter uncertainty estimates into our SLAM graph, as we describe in detail in Section VI-B.2. A full reference switch however is not always practical as the control of flying robots typically requires the navigation frame’s  $x/y$ -plane to be orthogonal to the gravity vector. We thus keep the *pitch* and *roll* estimates during a switch. From a filter perspective, this is unproblematic as both angles are well observable, their errors thereby being bounded.

### V. SUBMAPPING CONCEPT

In order to create dense 3D maps, we generate pointcloud-based submaps of limited size by locally integrating the 3D data resulting from dense stereo estimation along the trajectory estimated by the local reference filter, similar to our previous work [21]. On the creation of a new submap, we first switch the filter into a new reference frame. We then use this coordinate system as the origin of the novel

submap and represent it by a node in the SLAM graph, as described in detail in Section VI. As we employ a keyframe-based filter, see Section IV, drift within a submap can only occur during movements. We keep the drift within a submap small by triggering a new one when passing a threshold on the robot's integrated rotation of  $100^\circ$  or its driven distance of 2.5 m/2.0 m for our narrow/wide angle cameras respectively. Distance based thresholds have the advantage to lead to submaps of similar size, which is desirable for submap matching. In future work, we apply an additional threshold to limit the local uncertainty, which is estimated by the filter. Once a submap is completed, its 3D data is shared with all robots within communication range. Application-driven on-demand exchange or preprocessing of the submap data in order to reduce the network load represent important topics for future work, in particular w.r.t. to larger multi-robot systems. We compose all available submaps based on the latest global estimates of their origins, which we compute during SLAM graph optimization. The resulting full multi-robot 3D map can be used for visualization, navigation and planning. In addition, we generate intra- as well as inter-robot loop closure constraints for the SLAM graph through submap matching, see Section VII-B. Systems with limited computational resources might also employ sparser map representations or omit the computation of 3D maps while still taking advantage of the global pose estimates gained from graph optimization.

## VI. INCREMENTAL MULTI-ROBOT GRAPH SLAM

The SLAM problem can be formalized as a bipartite factor graph that represents the relevant 6D poses as variable nodes  $\theta_i \in \Theta$  and measurements as factor nodes  $f_i$ , i.e. constraints connecting subsets  $\Theta_i$  of them [9]. The graph defines a goal function  $f(\Theta) = \prod_i f_i(\Theta_i)$  over its variable nodes, the maximization of which coincides with a search for the most likely assignment of poses under the given measurements that constrain them. This can be formulated as a non-linear least squares minimization problem

$$\arg \min_{\Theta} (-\log f(\Theta)) = \arg \min_{\Theta} \frac{1}{2} \sum_i \|h_i(\Theta_i) - z_i\|_{\Sigma_i}^2 \quad (1)$$

with  $z_i$  denoting measurements and  $\Sigma_i$  their Gaussian noise as covariance matrices. The measurement functions  $h_i$  are typically non-linear in the context of 6D pose composition.

Graph SLAM systems can be divided into two parts. While the *back-end* is concerned with the minimization of the non-linear quadratic error function, the *front-end* deals with the construction of the graph. This means solving the data association problem and, by deciding on a graph topology, asserting (in)dependencies between the variable nodes [22].

### A. SLAM Back-End: Incremental Optimization

The computational effort of batch optimization makes it unsuitable for online application. We thus employ the incremental iSAM2 optimizer [9] to achieve fast updates on average, limiting slower optimization steps to the rare events of large loop closures. An open source implementation is

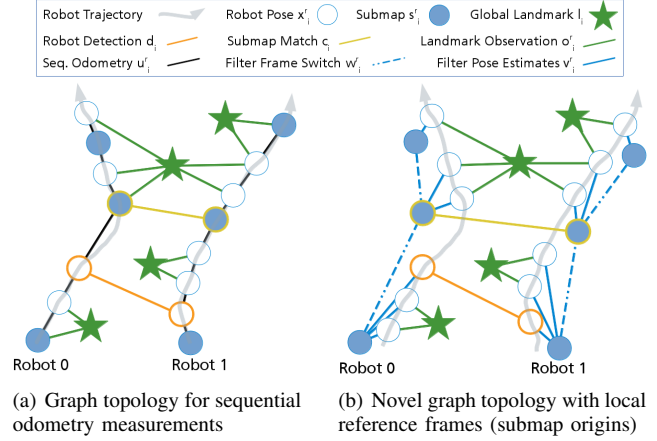


Fig. 3. Comparison of SLAM graph topologies

freely available as part of the GTSAM 3.2.0 library.<sup>1</sup> For our multi-robot experiments, we utilize the GTSAM implementation of M-Estimators, which replace the quadratic error term in Equation 1 with a robust error function. Thereby, we can mitigate the influence of potential errors and incorrect data associations in our robot detection and submap matching estimates. In particular, we employ the Cauchy function, as it is suitable to suppress outliers with large errors, which otherwise can corrupt the entire optimization result [23].

### B. SLAM Front-End: Multi-Robot Graph Topology

Considering a multi-robot setup with  $R$  robots, we include three different types of variable nodes: robot poses  $\{x_i^r\}$  and submap poses  $\{s_i^r\}$  for each robot  $r \in \{0, \dots, R-1\}$  as well as optionally poses of globally identifiable landmarks  $\{l_i\}$ . As all dense 3D data is attached to the submaps, we do not require a dense sampling of robot poses, but solely include them where needed to connect measurement factors. By keeping the size of the graph small, we allow for fast optimization steps. We can compute a global estimate for the pose of each robot  $r$  at any time by concatenating the most recent SLAM estimate of its current submap  $s_i^r$  with the latest local filter estimate, which is given relative to the submap's origin. As measurements, we consider robot detections  $\{d_i\}$  between pairs of robots, submap matches  $\{c_i\}$  between pairs of submaps as well as relative landmark observations  $\{o_i^r\}$ , all of which can lead to loop closure constraints. In addition, the robot and submap poses are connected via the filter estimates. In the following, we distinguish between two different graph topologies, as sketched out in Figure 3. As we only consider measurement factors connecting exactly two nodes, we collapsed them onto the edges for a more succinct graphical representation.

1) *Graph with Sequential Odometry Measurements*: With the graph topology depicted in Figure 3(a), as typically found in SLAM literature, submaps are attached to a subset of the robot poses, i.e.  $\{s_i^r\} \in \{x_j^r\}$ . All of these poses are connected sequentially via relative incremental robot ego motion estimates  $\{u_i^r\}$ . The placement of these constraints solely between consecutive robot poses is based on the

<sup>1</sup><https://collab.cc.gatech.edu/borg/gtsam>



assumption of their independence w.r.t. prior states. This approximation is reasonable for pure odometry measurements, e.g. wheel odometry or sequential scan-matching. These exhibit at most indirect dependencies through environment characteristics, which are hard to quantify in any case. When using local reference filters however, all estimates and their Gaussian uncertainties are computed relative to a local reference frame and may depend on each other through filter-internal states, such as keyframes. In our previous work [21], we approximated  $u_i^r$  as delta measurements by computing  $u_i^r = x_{i+1}^r \ominus x_i^r$ , which denotes the relative transformation between two consecutive robot poses. We employed  $\Sigma_{u_i^r} = \max(\Sigma_{x_{i+1}^r} - \Sigma_{x_i^r}, 10^{-10} \cdot I_6)$  as an approximation of its covariance matrix.  $\Sigma_{u_i^r}$  is guaranteed to be non-negative and above an experimentally determined threshold to ensure numerical stability during optimization. This rough approximation however completely ignores the aforementioned state dependencies. In the following, we therefore propose an adaption of the graph topology to allow a more suitable integration of the estimates computed by local reference filters.

2) *Graph with Local Reference Filter Estimates:* In our novel graph topology, as shown in Figure 3(b), we replace the approximated sequential odometry measurements  $\{u_i^r\}$  with two types of constraints that are directly estimated by the filter: First, transformations between submaps  $\{w_i^r\}$ , which refer to switches of reference frames in the filter. Second, robot pose estimates  $\{v_i^r\}$ , which are computed by the filter relative to their local reference frames ( $\hat{=}$  submap origins). Note that due to the alignment of local reference frames to the gravity vector, as explained in Section IV, the 6D submap origins  $s_i^r$  do not necessarily coincide anymore with the robot poses  $x_i$ . By directly integrating local reference filter estimates, our novel graph topology better represents the underlying probabilistic structure. Within a reference frame, it introduces no additional independence assumptions. Overestimations of the covariances however can still occur when a filter switches its local reference frame and afterwards integrates a visual odometry measurement that refers to a keyframe measured within a previous frame. In order to avoid these cases, the SLAM would require knowledge about the keyframes themselves. This however would prevent a decoupling of the SLAM from filter-internal states, which is important to keep the graph compact while allowing the filter to use any kind of internal state representation as well as the integration of high-frequency measurements. We thereby achieve more design freedom for the filter compared to a tightly coupled solution like Concurrent Filtering and Smoothing [11]. Our novel graph topology allows a straightforward addition of pose estimates and delayed measurements at any point in time. A structure suitable for incremental online optimization is maintained without requiring additional methods to remove constraints from the graph or to avoid double-counting of information, such as anti-factors [17]. In Section VIII-B, we compare both graph topologies in our experiments, showing an improved localization accuracy for our novel approach.

## VII. INTER-ROBOT MEASUREMENTS

Inter-robot measurements, comprising robot detections and submap matches, represent direct connections between poses or submaps of multiple robots. Additional links can be established through landmark observations, as shown in Figure 3.

### A. Robot Detections

We utilize the images from our stereo cameras to detect planar visual markers that are attached to robots in our multi-robot team, as shown in Figure 1. We employ AprilTag markers, which have been shown to outperform similar 2D bar code style markers w.r.t. accuracy and detection range [19]. Several open source implementations of AprilTag detectors are freely available.<sup>2</sup> These detect the marker in an image and estimate a 6D transformation between the camera and the marker's origin. We however experienced that the quality of these estimates highly depends on the distance and angle of the observation. Thus, simply assuming a fixed covariance is not adequate. In order to get more accurate results, we analyzed the behavior of the detector, derived a camera-dependent model for its errors and compute a worst-case estimate for the uncertainty of each observation. We plan a separate publication for an in-depth description as it would go beyond the scope of this paper. In order to add robot detections to the SLAM graph, it is necessary to know the poses of both robots at the point in time for which their relative transformation has been estimated. To cope with interrupted or delayed communication, each robot keeps a buffer of its own pose estimates to facilitate a lookup in case it had been detected in the past. Its memory footprint however is negligible compared to the 3D submap data. In our experimental comparison of graph topologies, see Section VIII-B, we utilized the same marker detector and uncertainty estimation to observe static landmarks.

### B. Submap Matches

In previous work [4], we have shown that arrangements of obstacles described by their surface topology are valuable features for indoor as well as unstructured outdoor environments, since they are robust w.r.t. changes in viewpoint and light conditions. We therefore compute pointcloud-based local obstacle maps, considering terrain roughness, slope and step height for obstacle classification as described in [4] and select potential matches by considering the overlap of their bounding boxes (min. 6 m<sup>2</sup>), arranged according to their current pose estimates obtained from graph optimization. We organize the matching candidates in a priority working queue. They are ordered by the expected impact of a match, for which the difference in variance estimates between two submaps' origins serves as a heuristic. The matching itself is then executed as a low-priority background process that can easily be parallelized. We match submaps by searching for similarities within their geometric structure by applying CSHOT [24] as a 3D feature descriptor that is rotational invariant, robust to noise and clutter and includes texture

<sup>2</sup><http://april.eecs.umich.edu>, <http://people.csail.mit.edu/kaess/apriltags/>

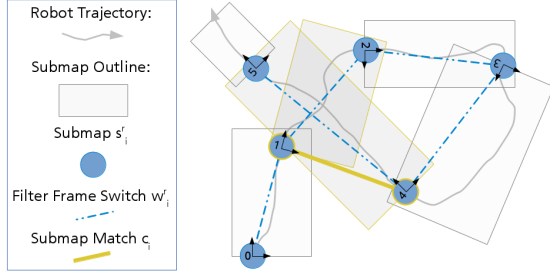


Fig. 4. Submap matching: The two highlighted submaps match and lead to a loop closure constraint (yellow edge) in the graph.

information to improve the matching accuracy. The resulting 6D transformation estimate is refined by applying the Iterative Closest Point (ICP) algorithm on the full pointcloud, which provides accurate results, but requires close-enough initial estimates. We filter matches between  $s_i$  and  $s_j$  as outliers that exceed the combined estimated submap pose uncertainties in any dimension. In order to accurately integrate the resulting loop closure constraints into the SLAM graph, see Figure 4, we estimate the 6D variance considering the submaps resolution and the alignment errors estimated during the ICP step. We present further details on the submap matching itself in a separate publication [21].

## VIII. EVALUATION

### A. Experimental Setup

We performed experiments on three different robotic platforms: Our all-terrain Lightweight Rover Unit (LRU) with a pan-tilt sensor head for outdoors and both a Pioneer 3-AT (P3AT) and a Pioneer 3-DX (P3DX) for indoor experiments. All robots are equipped with a Xsens MTi-10 IMU and a stereo camera system (baseline: 9 cm), allowing us to gather dense depth data both indoors and outdoors as it is robust to the effects of bright sunlight. Our camera setup is:

- *LRU*: Guppy PRO F-125B, Narrow-angle:  $f = 5$  mm
- *P3AT*: Guppy F-080B, Narrow-angle:  $f = 5$  mm
- *P3DX*: Guppy PRO F-125B, Wide-angle:  $f = 1.28$  mm

While we employ an on-board Spartan 6 LX75 FPGA for dense stereo matching ( $1024 \times 508$  px at 14.6 Hz), all other computation is performed on a quadcore CPU. In our experiments, we shared all measurements, pose estimates and submap pointclouds between the robots. In communication settings with limited bandwidth, it would however be sufficient to limit the communication to the typically sparser set of estimates depicted in Figure 3(b) and transfer pointcloud map data solely on demand. For our outdoor experiment, we acquired ground truth position data through a Leica total station and aligned its initial reference frame to our map frame as described in [4]. For the mixed and indoor experiments, we received ground truth trajectories from a ceiling-mounted Advanced Realtime Tracking (ART) system. As its tracking area is limited to approx.  $3 \text{ m} \times 4 \text{ m}$ , we could solely evaluate the respective partial trajectories. Note that for trajectory evaluation, we always use the sequentially logged SLAM estimates at each particular point in time, not an afterwards fully optimized trajectory. Hence, before

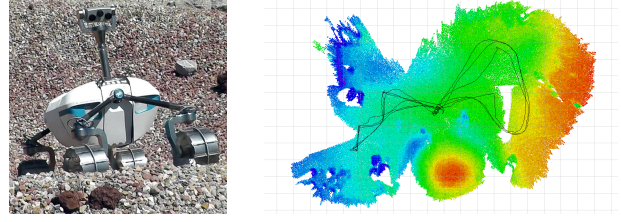


Fig. 5. Outdoor experiment: Photo of LRU and height-colored 3D map of outdoor scenario and SLAM path (see also Table I).

Scenario	indoor		mixed		outdoor	
Robot	P3AT		P3AT		LRU	
Driven distance	69 m		320 m		106 m	
Ground truth available	25 m		10 m		106 m	
3D trajectory error	$\mu$ [m]	$\sigma$ [m]	$\mu$ [m]	$\sigma$ [m]	$\mu$ [m]	$\sigma$ [m]
Filter only	0.327	0.109	1.324	0.886	0.402	0.192
Seq. odom. SLAM [VI-B.1]	0.177	0.060	0.121	0.082	0.153	0.083
Local ref. SLAM [VI-B.2]	<b>0.146</b>	<b>0.054</b>	<b>0.096</b>	<b>0.060</b>	<b>0.142</b>	<b>0.079</b>

TABLE I

COMPARISON OF SLAM GRAPH TOPOLOGIES

the first loop closure, the SLAM trajectory is equal to the filter estimate. We chose this evaluation criterion, as for autonomous robots, only the past and current estimates are available at each particular point in time.

### B. Comparison of SLAM Graph Topologies

First, we evaluated the impact on the localization accuracy of our novel SLAM graph topology for local reference filters, see Section VI-B.2. Therefore, we compare it to the standard graph topology for sequential odometry that we previously used as an approximation as described in Section VI-B.1. We performed this evaluation in single-robot setups on data gathered by two different robots in indoor, mixed and unstructured outdoor environments similar to the scenarios described in [21]. An impression of the outdoor testbed is given in Figure 5. As we conducted this evaluation prior to the full integration of our submap matching components, we employed AprilTag markers as artificial landmarks to generate loop closure constraints and ran both SLAM implementations in parallel on the same input data and filter estimates. A direct evaluation of the uncertainty estimates represented by the two graph topologies is impossible due to a lack of corresponding ground truth data. However, a better approximation of the underlying probabilistic structure is expected to lead to more accurate graph optimization results. Therefore, we employ the SLAM trajectories as an indirect measure for the accuracy of the covariance estimates and (in)dependencies asserted by the graph topology. We summarize the results in Table I, comparing the 3D trajectory errors between both SLAM graph topologies. All three experiments exhibit an improvement of the 3D localization accuracy, on average of 15 %. The results thus indicate the benefit of adapting the graph topology to the underlying structure of local reference filter estimates. While a comparison to full batch optimization and an evaluation of the consistency of the proposed sub-optimal data fusion method are important topics for future work, the consistency of the local reference filter itself has already been investigated in [2]. For a

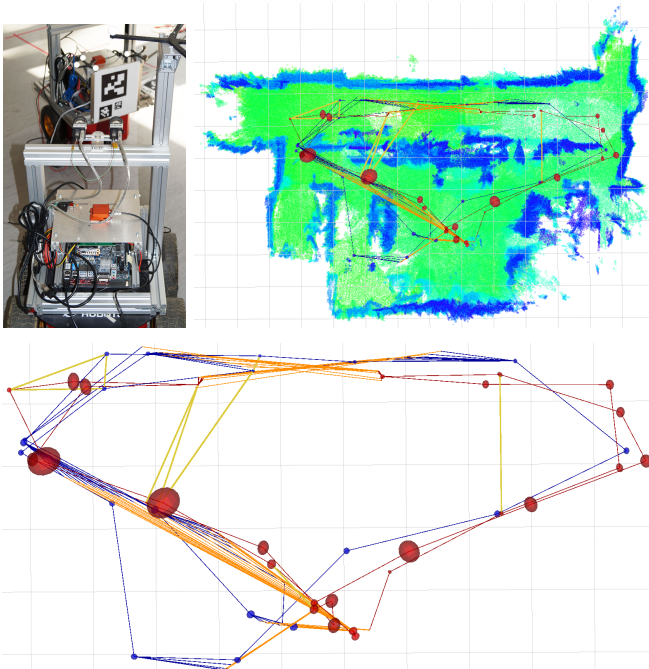


Fig. 7. Multi-robot experiment, top-down view of final 3D map (height-based coloring) and graph. Ellipsoids show the submap origins (P3AT: blue, P3DX: red) and are scaled to their respective position standard deviation estimates. Orange edges in the graph represent robot detections (P3AT detects P3DX1), yellow edges submap matches. See Table II for details.

simulated 24h quadrotor flight, its covariance estimates are shown to be conservative for position, velocity and attitude.

Updates of the SLAM graph on new measurements (avg: 0.03 s, max: 0.12 s), which include the incremental graph optimization itself (avg: 0.008 s, max: 0.08 s) as well as the computation of all submap pose covariances through marginalization, are fast enough for online operation. In addition, our code had not yet been optimized w.r.t. its runtime, thus we expect room for significant improvements.

### C. Collaborative Localization and Mapping

For the evaluation of the full SLAM system, we performed two extended multi-robot experiments with two rovers in our indoor lab environment (approx. 100 m<sup>2</sup> area), as shown in Figure 1. It consists of a long hallway (top) and a large room (bottom), in which the tracking system is set up. We could not find any publicly available multi-robot dataset with stereo images and inter-robot detections for additional evaluation. For the first experiment, the robots have no initial knowledge about their relative positions. In Figure 6, we present a time-series of maps, in which robot detections serve as inter-robot measurements. The first ones, at a range greater than 5 m, are imprecise and lead to an error in *yaw* that is visible in the top part of Figure 6(b). The integration of further detections into the graph optimization however allows a more accurate alignment, as can be seen in Figure 6(c). No artificial static landmarks have been used in our multi-robot experiments.

For the second experiment, we integrated submap matches as additional loop closure constraints. Both robots observe most parts of the environment multiple times. In Figure 7, we show a photo of our two robots and present the final map

	Multi-Robot		Single-Robot	
	P3AT	P3DX1	P3AT	P3DX1
Number of robot poses $x_i^r$	72	72	0	0
Number of submaps $s_i^r$	21	27	21	27
Number of submap matches $c_i$	4	3	2	7
per robot inter-robot	4		0	
Number of robot detections $d_i$	72	0	0	0
Total number of nodes $\theta_i$	192		21	27
Total number of factors $f_i$	274		30	36
Driven distance [m]	46.49	44.56	46.49	44.56
Ground truth available [m]	9.55	12.48	9.55	12.48
Mean 3D trajectory error $\mu$ [m]	<b>0.167</b>	<b>0.210</b>	0.215	0.360

TABLE II

COMPARISON OF MULTI-ROBOT AND SINGLE-ROBOT SLAM

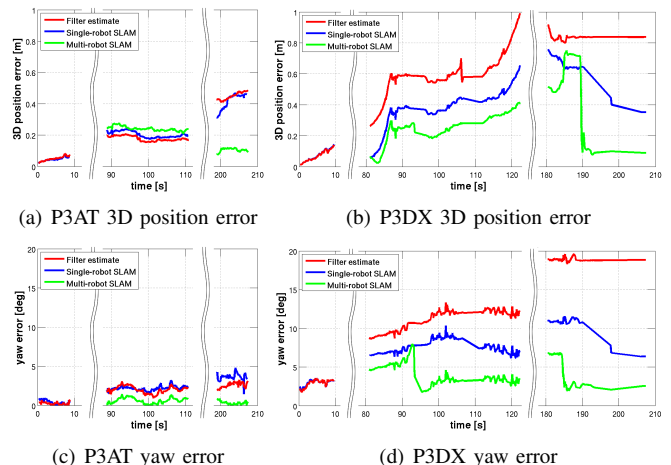


Fig. 8. Pose errors for the multi-robot SLAM experiment presented in Figure 7 and Table II. Plots show the partial trajectories for which ground truth is available. All estimates refer to the sequentially logged data available at each particular point in time, not an afterwards fully optimized trajectory.

and SLAM graph. In Table II and Figure 8, we compare our multi-robot SLAM system to single-robot SLAM on the same dataset. The estimates for the multi-robot setup are on average by 32 % more accurate and thus indicate the benefit of joint graph optimization compared to estimating a single relative transformation between the robots' coordinate frames to connect their maps. Inter-robot submap matches and robot detections add additional loop-closure constraints compared to single-robot SLAM. The robots serve as virtual "moving landmarks" to improve each other's localization, in particular when the quality of their intermediate local pose estimates differs between consecutive robot detections. Due to its wide-angle stereo setup, P3DX exhibits a larger visual odometry error than P3AT [4], represented by greater covariances in the graph, see Figure 7. In our experiment, the robot detections thus primarily improve the localization for P3DX, as can be observed in Figure 8. The quadratic error minimization during graph optimization leads to a temporary slight degradation of the SLAM position estimate for P3AT in the middle part of our evaluation, see Figure 8(a). At this point in time only few submaps have been matched due to a small overlap of the robots' trajectories. Additional matches later on however significantly improve both robot's estimates. The final maps show a consistent representations of the walls and doorways of our indoor scenario. It constitutes a challenging environment for stereo vision with texture-less

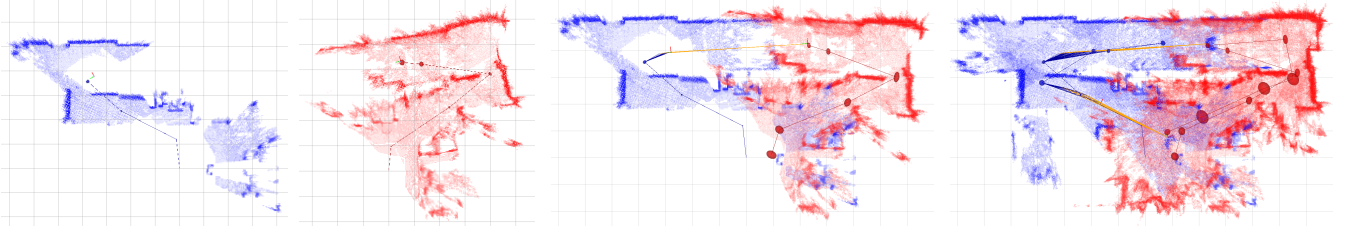


Fig. 6. Multi-robot experiment with two robots (P3AT: blue, P3DX: red) and unknown starting positions. Separate maps computed by both robots before (a) and after (b) the first connection as well as after two series of inter-robot detections (c). The ellipsoids show the submap origins and are scaled to their respective position standard deviation estimates. Blue and red edges connect submaps and robot poses, orange edges represent robot detections.

walls, reflecting surfaces and regular patterns (e.g. radiators) that lead to visual odometry and depth errors, which can be observed as noise in our maps. Their representations do not include any assumptions about structure in the environment, i.e. no biases towards straight walls or even floors. We show the mapping process for both multi-robot experiments in the accompanying video.

## IX. CONCLUSION AND FUTURE WORK

In this work, we have presented a multi-robot 6D SLAM system that employs a novel graph topology to incorporate the results of decoupled local reference filters according to their uncertainty estimates. By aggregating sensor information into submaps attached to local reference frames, we keep the effort for graph optimization small and are able to online generate dense 3D maps from stereo data. Further, we include visual robot detections and submap matches as inter-robot loop closure constraints and integrated all components into our modular mapping architecture. We performed experiments, both to evaluate the impact of our novel integration of local reference filters on localization accuracy and to evaluate the application of our full SLAM framework on a multi-robot system, thereby combining the data of stereo systems with different perspectives and angles of view. For future work, we plan to evaluate our approach with a team of flying and driving robots to further increase the diversity. While we manually controlled the robots for this work, we intend to employ our maps for collaborative autonomous exploration in the future. Another open challenge is the merging of submaps when no further improvement of their relative transformation can be expected.

## ACKNOWLEDGMENT

We thank the members of the Mobile Robots Group at DLR-RMC, especially Korbinian Schmid and Felix Ruess for many fruitful discussions and their support for the local reference filter. We thank Sebastian Vetter for his work on the uncertainty estimation for marker-based robot detections and his assistance with the experiments.

## REFERENCES

- [1] H. Hirschmüller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IPAMI*, vol. 30, no. 2, pp. 328–341, Feb. 2008.
- [2] K. Schmid, F. Ruess, and D. Burschka, "Local Reference Filter for Life-Long Vision Aided Inertial Navigation," in *FUSION*, 2014.
- [3] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping : Part I," *RAM*, vol. 13, no. 2, pp. 99–110, 2006.
- [4] C. Brand, M. J. Schuster, H. Hirschmüller, and M. Suppa, "Stereo-Vision Based Obstacle Mapping for Indoor / Outdoor SLAM," in *IROS*, Chicago, Illinois, USA, 2014.
- [5] P. B. Quang, C. Musso, and F. Le Gland, "An Insight into the Issue of Dimensionality in Particle Filtering," in *FUSION*, 2010.
- [6] J. Welle, D. Schulz, T. Bachran, and A. B. Cremers, "Optimization Techniques for Laser-Based 3D Particle Filter SLAM," in *ICRA*, Anchorage, Alaska, USA, 2010.
- [7] L. Carlone, M. K. Ng, B. Bona, and M. Indri, "Rao-Blackwellized Particle Filters Multi Robot SLAM with Unknown Initial Correspondences and Limited Communication," *ICRA*, May 2010.
- [8] A. Howard, "Multi-robot Simultaneous Localization and Mapping using Particle Filters," *IJRR*, vol. 25, no. 12, pp. 1243–1256, 2006.
- [9] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2 : Incremental Smoothing and Mapping Using the Bayes Tree," *IJRR*, vol. 31, pp. 217–236, 2012.
- [10] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "RSLAM: A System for Large-Scale Mapping in Constant-Time Using Stereo," *IJCV*, vol. 94, no. 2, pp. 198–214, June 2010.
- [11] S. Williams, V. Indelman, M. Kaess, R. Roberts, J. J. Leonard, and F. Dellaert, "Concurrent Filtering and Smoothing : A Parallel Architecture for Real-Time Navigation and Full Smoothing," *IJRR*, vol. 33, no. 12, pp. 1–47, 2014.
- [12] A. Ahmad, G. D. Tipaldi, P. Lima, and W. Burgard, "Cooperative Robot Localization and Target Tracking based on Least Squares Minimization," in *ICRA*, 2013.
- [13] R. Reid and T. Bräunl, "Large-scale Multi-robot Mapping in MAGIC 2010," *RAM*, 2011.
- [14] T. A. Vidal-Calleja, C. Berger, J. Solà, and S. Lacroix, "Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain," *RAS*, vol. 59, no. 9, pp. 654–674, Sept. 2011.
- [15] R. C. Leishman, T. W. McLain, and R. W. Beard, "Relative Navigation Approach for Vision-Based Aerial GPS-Denied Navigation," *ICUAS*, 2013.
- [16] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple Relative Pose Graphs for Robust Cooperative Mapping," in *ICRA*, 2010.
- [17] A. Cunningham, V. Indelman, and F. Dellaert, "DDF-SAM 2.0: Consistent Distributed Smoothing and Mapping," in *ICRA*, 2013.
- [18] H. Hirschmüller, P. Innocent, and J. Garibaldi, "Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics," *ICARCV*, vol. 2, pp. 1099–1104, 2002.
- [19] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *ICRA*, 2011.
- [20] K. Schmid, F. Ruess, M. Suppa, and D. Burschka, "State Estimation for highly dynamic flying Systems using Key Frame Odometry with varying Time Delays," in *IROS*, 2012.
- [21] C. Brand, M. J. Schuster, H. Hirschmüller, and M. Suppa, "Submap Matching for Stereo-Vision Based Indoor/Outdoor SLAM," in *IROS*, Hamburg, Germany, 2015.
- [22] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM," *ITSM*, vol. 2, no. 4, pp. 31–43, 2010.
- [23] G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Robust Pose-Graph Loop-Closures with Expectation-Maximization," in *IROS*, 2013.
- [24] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3d feature matching," in *ICIP*, 2011.