SLAM", 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2015)

Submap Matching for Stereo-Vision Based Indoor/Outdoor SLAM

Christoph Brand, Martin J. Schuster, Heiko Hirschmüller and Michael Suppa

***Abstract*— Autonomous robots operating in semi- or unstructured environments, e.g. during search and rescue missions, require methods for online on-board creation of maps to support path planning and obstacle avoidance. Perception based on stereo cameras is well suited for mixed indoor/outdoor environments. The creation of full 3D maps in GPS-denied areas however is still a challenging task for current robot systems, in particular due to depth errors resulting from stereo reconstruction. State-of-the-art 6D SLAM approaches employ graph-based optimization on the relative transformations between keyframes or local submaps. To achieve loop closures, correct data association is crucial, in particular for sensor input received at different points in time. In order to approach this challenge, we propose a novel method for submap matching. It is based on robust keypoints, which we derive from local obstacle classification. By describing geometrical 3D features, we achieve invariance to changing viewpoints and varying light conditions. We performed experiments in indoor, outdoor and mixed environments. In all three scenarios we achieved a final 3D position error of less than 0.23% of the full trajectory. In addition, we compared our approach with a 3D RBPF SLAM from previous work, achieving an improvement of at least 27% in mean 2D localization accuracy in different scenarios.**

I. Introduction

In search and rescue (SAR) scenarios, supporting the situational awareness of the rescue workers is crucial in order to improve their efficiency as well as to keep them out of danger. As the mission environments after disasters typically include areas that are hard or dangerous to access, mobile robots can be deployed to support them. In such partially destroyed, semi- or unstructured indoor and outdoor environments, (D)GPS-like global methods for accurate external localization might not be available at all times. Furthermore, communication delays and failures are to be expected to occur during the mission. (Semi-)autonomous operation of the robots is thus required to deal with these challenges and to relieve operators of tedious low-level control tasks. In order for the robots to operate in a previously unknown environments, local and global localization and mapping has to be performed online and on-board the individual systems. In SAR scenarios, performing full 6D localization and generating a 3D map can be advantageous, in particular when operating in caves or multi-story buildings with indoor areas as well as when deploying aerial robots like quadrotors.We therefore designed a mapping framework for robots operating in GPS-denied, previously unknown indoor as well as rough-terrain outdoor environments. *Semi-global stereo matching (SGM)* [1] on images gathered by a pair of cameras provides us with dense depth information under varying light conditions, even in mixed indoor/outdoor environments. For global optimization, we employ incremental graph SLAM methods to minimize the quadratic error on the robot's estimated trajectory. These can run online and on-board the robot to provide a global pose and map estimate at any time. A particular challenge is the association of sensor data to generate loop closures by matching features extracted from the environment model. Compared to laser scanners, stereo cameras typically have a narrow angle of view. This complicates the crucial data association between a measurement and a (partial) map. We compensate this by following a submapping approach that locally integrates multiple measurements from an area of limited size.

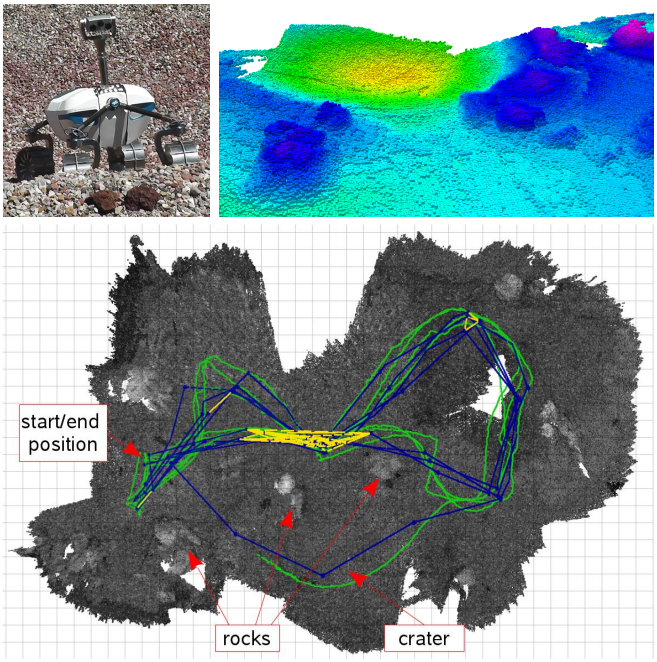

Fig. 1. Top: Lightweight Rover Unit (LRU) with 3D representation of a crater in our outdoor testbed, corresponding to the central part of the map displayed below. Bottom: Top-down view on resulting 3D map generated by our SLAM system (green path: SLAM estimates available to the robot at its respective positions, blue: SLAM graph edges between subsequent submaps from sensor fusion, yellow: edges representing submap matches).

As the central contribution of this work, we developed an algorithm to select and match pairs of stereo-vision based submaps, thereby computing an estimate for their relative transformation as well as for its uncertainty. In order to generate accurate and robust matches, we introduce multiple filtering and selection steps. As a starting point, during submap creation, we compute local obstacle maps. These contain the 2.5D results from a stereo-error adaptive obstacle classification, which we presented in [2]. The depth error of

The authors are with the German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Department of Perception and Cognition, Münchner Str. 20, 82234 Wessling, Germany
{christoph.brand|martin.schuster|heiko.hirschmueller|michael.suppa} at dlr.de

any stereo algorithm grows quadratically with the distance to the cameras. Therefore we have to consider it early on in our mapping pipeline, as the association between camera viewpoints and depth data will be lost after the integration into submaps. The local obstacle maps thereby allow the computation of robust keypoints, which is the first step of the submap matching process. We characterize them with CSHOT descriptors [3], which combine geometrical and visual characteristics. Features based on the 3D geometry of the environment are particularly invariant to a robot's viewpoint as well as to varying light conditions. We thereby overcome a typical limitation of purely image-based features like SIFT, while still having the option to include texture information. As the second step, we select and rank potential matches based on the expected success rate of the matching process as well as on their expected impact on the graph optimization. This creates a prioritized work queue for the remaining parts of the matching algorithm, which as the third step performs a keypoint matching. On success, we employ the resulting transformation as the initial estimate for an Iterative Closest Point (ICP) optimization, which constitutes the fourth step. We perform the ICP on the full pointcloud that, in contrast to the local obstacle maps, also includes the traversable ground. This ensures a high precision of the final transformation, in particular w.r.t. the z-axis as well as roll and pitch angles. The fifth and final step of our matching pipeline constitutes a rating of the uncertainty of the resulting transformation as well as an outlier filtering. We integrated our novel submap matching algorithm with our SLAM framework and performed experiments in our outdoor testbed (see Figure 1), in an indoor lab environment as well as in a mixed indoor/outdoor setting. We thereby demonstrated the applicability of our methods to different scenarios and their robustness to varying environments and light conditions. With our mapping system, we were able to achieve a final 3D position error of less than 0.23% of the full trajectory in all three scenarios. In addition, we compared our novel approach to a 3D RBPF SLAM from our previous work [2], achieving an improvement of at least 27 % in mean 2D localization accuracy in different scenarios.

II. Related Work

The large body of related work concerned with the task of simultaneous localization and mapping (SLAM) can be partitioned into three major techniques: Extended Kalman Filters (EKF), Rao-Blackwellized particle filters (RBPF), and graph optimization approaches. For a general overview, see [4], [5] and [6]. EKF approaches model their landmark-based maps as multivariate Gaussians. Their major drawback is the quadratic growth of the computational effort with the number of landmarks [5], however variations with lower complexity exist [6]. RBPFs [7] [8] are typically employed to optimize a distribution over robot trajectories along with grid maps. They yield robust solutions for planar localization and mapping, as we recently demonstrated for stereo vision data with appropriate preprocessing [2]. However, extensions from three to six degrees of freedom are computationally challenging w.r.t. runtime and memory requirements [9], as the number of particles needs to grow exponentially with the size of the state space to avoid weight collapse [10]. RBPFs are thus not well suited for 6D SLAM.

While graph SLAM approaches have started out as batch methods [11] for offline use, recent advances in incremental graph optimization [12] allow their application for online localization and mapping. They thus currently appear as the most promising method for 6D SLAM. Their graph represents robot poses and (optional) landmarks as nodes, interconnected by their associated measurements. These edges, weighted by the Gaussian measurement uncertainty, serve as constraints for global optimization, which is then applied to minimize their quadratic error. This process is sensitive to overconfident false measurements, resulting e.g. from erroneous data associations. However several methods that are robust to outliers have been developed [13]. The size of the graph and thus the worst-case computational effort on loop closures grows with the traveled distance. Constraining the optimization to local regions [14] or removing nodes through marginalization [6] are techniques to deal with this challenge. While graph optimization constitutes the back-end of a SLAM framework, the front-end is concerned with solving the data association task. Established methods are the identification of landmarks, e.g. in form of image features [15], the matching of visual key-frames [16], as well as the registration of depth data through Iterative Closest Point (ICP) techniques [17].

Submapping approaches [18] aggregate local sensor data into multiple maps of limited size and attach their origins as poses to the graph. This allows for a sparse graph structure and thereby efficient optimization steps while keeping more information compared to key-frame approaches. In addition, they are also suitable for multi-robot scenarios, for example by matching and merging 2D occupancy grid maps using Hough transforms [19]. They however require linear features, typically lacking in unstructured environments. Reid et al. [18] designed a graph-based multi-robot SLAM system, using a brute-force GPU-based correlation search on 2D occupancy maps to find matches. Labbé and Michaud [20] employ graph SLAM for multi-session mapping, using a laser rangefinder and a Kinect. Loop closures are detected through matching of SURF features, which however are not robust to changes in viewpoint and illumination. Kinect fusion [17] is a popular algorithm for the generation of consistent 3D RGBD maps and smooth 3D reconstruction, using a frame-to-frame ICP registration. However global loop closure optimization is not performed. As for 3D registration, ICP approaches often become trapped in minimums and fail to provide correct solutions, the use of 3D feature descriptors [21] has become popular in order to find correspondences within pointclouds. The central challenge is to select and describe robust 3D features. Yousif et al. [22] highlight the importance of robust keypoints by employing ranked order statistics to achieve improvements over other subsampling techniques for frame-to-frame registration in indoor scenarios. While SHOT [23] feature descriptors are used

for pointcloud matching in texture-less environments, we employ CSHOT [3] features to include texture information where available. Furthermore, in [2] we have shown that our local obstacle maps yield good geometric landmarks for both indoor and outdoor scenarios.

III. System Architecture

In Figure 2, we present an overview over our software architecture. We employ ROS as a middleware to connect the individual components. For the task at hand, we introduce a division into three layers: *perception*, *local and global mapping* as well as *planning and control*.

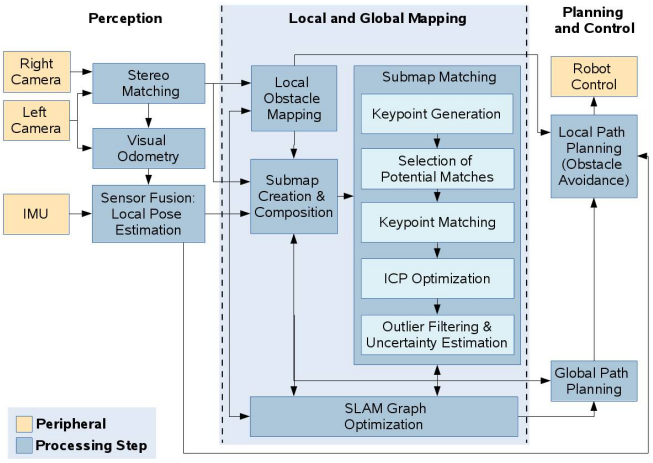

Fig. 2. Software architecture block diagram. The focus of this paper is on the local and global mapping part.

Our only sensors are a pair of cameras and an IMU, similar to our setup in [2]. We employ an FPGA implementation of the Semi-Global Matching (SGM) algorithm [1] for dense stereo matching and use the results to compute visual odometry estimates as outlined in [24] [25]. We fuse these with IMU data in an keyframe-based Extended Kalman Filter (EKF) with time-delay compensation for real-time robust local pose estimates [26]. For these steps, we employ the same implementations as on our flying robots [26]. The resulting pose estimates are utilized both for the integration of measurements in our local submaps as well as for relative transformation estimates between the submaps within our SLAM graph. The pose and map estimates can readily be utilized for path planning, for example by using the local and global obstacle maps as input to the ROS Navigation Stack[1] for obstacle avoidance and global planning. While we performed our mapping experiments presented in this work through teleoperation, we already conducted preliminary experiments on semi-autonomous waypoint-navigation with the full system.

IV. Mapping

A. Local Obstacle Mapping

We generate local obstacle maps from stereo data aligned by local pose estimates [2]. They can be used for fast local path planning and obstacle avoidance. The stereo error in the distance l_c from the camera $\Delta l_c \approx \Delta p \frac{P_z \, l_c}{f \, t} \sqrt{2}$ grows quadratically in the z-direction P_z and varies for different camera systems (with mean pixel error Δp, focal length f and stereo-baseline t) [25]. In order to receive robust obstacles, we thus consider the stereo error of our camera system for step and slope estimation. Furthermore, we detect negative edges, like cliffs or stairheads, directly in the depth image. These represent distinctive landmarks in the environment. As a final step, we filter outliers and perform a time-based probabilistic integration. Our obstacle mapping pipeline has proven to be robust to varying environmental conditions and sensor setups. The resulting obstacles constitute discriminative and robust geometric features for our submap matching approach, since we have already successfully applied them as input for a particle-filter based 3D SLAM [2]. In addition, we utilize them for a final alignment check. As we employ the obstacle maps also for local path planning, this processing step does not generate any additional computational overhead, in contrast to [22].

B. Submapper

Our submapping component generates and manages submaps of limited size. They can be composed to a full 3D map, e.g. for visualization and navigation purposes, as well as employed for submap matching. From the greyscale and depth images, we first compute 3D pointclouds. We only consider points within a maximum distance of $3.5\,\mathrm{m}$ from the camera, because for larger distances the stereo error would exceed the map voxel density. Each new submap is anchored with its origin at the robot's current pose, which is added as a node to the SLAM graph (see Section IV-D). During the creation of the submap, we use local pose estimates from the sensor fusion algorithm to integrate pointclouds into the local coordinate frame of the submap. In our current setup, a submap contains both, a full pointcloud, as well as an obstacle pointcloud, resulting from the aforementioned obstacle mapping step. In order to limit the drift within the submaps while still providing a sufficient submap size for matching, we empirically determined system-dependent criteria to trigger submap creation. We start new submaps after a maximum driven distance of $2.5\,\mathrm{m}$ or a maximum integrated rotation of $90\,^\circ$, whichever criterion is met first. Due to the keyframe-based fusion algorithm, the pose error is only increased while the camera is moving. In future work, we consider the estimation uncertainty within a submap as an additional criterion to start new submaps. To finalize a submap, we apply a voxel-grid filter with a resolution of $3\,\mathrm{cm}$ in order to reduce the impact of the radial distribution and the computational requirements for subsequent processing steps.

C. Submap Matching

In this section, we describe the processing pipeline of our submap matching component, as outlined in Figure 2.

1) Selection of Valid Submaps and Keypoints: The final submaps are sequentially sent to the matching module. Not all submaps are equally suitable for matching. There are

[1] http://wiki.ros.org/navigation

different criteria that can be applied a priori to select valid submaps. Initially, we dismiss submaps that are too small in size ($< 5\,\mathrm{m}^2$) or contain less than 2000 points. As most environments, especially indoor, contain areas with minor slope and roughness, like floors and straight walls, keypoints taken in those regions would result in indiscriminative descriptors and thus lead to wrong matches. In order to select distinctive geometric features, we employ the submap's precomputed obstacle pointcloud to define valuable keypoints for the matching, see Section IV-A. As we consider the stereo error within the depth image during obstacle map creation, they contain solely robust obstacles, in particular w.r.t. to changing viewpoints and camera distances. We get the corresponding keypoints within the full pointcloud by applying a nearest neighbor search. Considering lower resolutions and stereo errors, arrangements of obstacles provide more reliable landmarks in order to distinguish different submaps than a single obstacle. For example, individual stones in an outdoor environment can look quite similar, even from different viewpoints, depending on the resolution of the map. A lower resolution reduces the computational load, which is an issue for on-board computation on systems with limited resources, and thus worthwhile for our application. Hence, we define a minimum bounding box in the x/y-plane of $4\,\mathrm{m}^2$ for an obstacle formation with a resolution of $3\,\mathrm{cm}$, since feature descriptors of small separate obstacles might be ambiguous for lower resolutions. Stereo measurement errors can lead to sparse outliers, which corrupt the resulting pointcloud and lead to errors in its local characteristics like normals and curvatures, thereby affecting the map matching. We thus apply a statistical filter to remove outliers based on their distribution of distances to neighboring points [27]. The remaining submaps, including their precomputed keypoints, are cached in an indexed container, which is used to search for potential matching pairs. For multi-robot systems, we also include submaps created by different robots in order to look for inter-robot matches.

2) Selection of Potential Matching Pairs: As the matching step is time consuming and the whole SLAM system is running on-board and online, computational load is an issue. Despite the fact that the matching itself is not a time critical operation, a brute force approach would not be reasonable, since trying out all $\frac{n!}{2(n-2)!}$ combinations would computationally be impossible on a single machine within an acceptable timeframe. By a priori determining possible matches, we in addition exclude potential false positives. Therefore, it is important to score and rank potential matching pairs w.r.t. to their probability to match as well as the expected impact of the matches on global optimization. In particular, as matching is not performed in real-time, it can be implemented as a background process that always deals with the currently most promising pairs of maps, being executed only when sufficient resources are available. For a multi-robot scenario, including systems with limited computational power like light-weight UAVs, it would also be easily possible to outsource this computational step to another machine. We use thresholds on the minimum overlap of the bounding boxes of two submaps in the x/y-plane, taking into account the $2\,\sigma$ covariance bounds for their origins, which we obtain online from the graph SLAM. In addition, we require the bounding boxes of the obstacle keypoints and of the corresponding pointclouds to overlap by at least $3\,\mathrm{m}^2$ and $4\,\mathrm{m}^2$ respectively. In order to compute the expected impact of the matches on global optimization, we look at the time of creation of the submaps. Assuming a constant drift, matches of consecutive submaps are less valuable than matches bridging a long temporal distance. For future work, we plan to make use of the variance estimates between submaps to devise an improved heuristic for ranking. Furthermore, in order to obtain a consistent 3D map of the environment, we discovered that evenly distributed matches are preferable. A concentration within a relatively small area can result in a large error in distant parts of the map due to the influence of angular errors. Therefore, we organize the submaps on a 2D grid with a resolution of 3m, each cell representing a histogram bin that contains the number of matches performed on submaps located in the respective cell. Consequently, we prioritize submaps in cells with less votes.

3) Keypoint Matching: We start by retrieving the top element of our priority queue, which contains potential matches. First, we estimate the surface normals for each point in the submap as they are important properties of geometric surfaces. For this purpose, we execute a local least-squares plane fitting. As we are considering different camera systems and submap resolutions, we employ a resolution r_{res} adaptive point radius $r_N = \min(0.2\,\mathrm{m}, 4\,r_{res})$. Local surface properties, like normals and curvatures, are used to characterize a 3D feature. In order to locally describe the previously selected keypoints, we compute the 3D feature descriptor SHOT [23], which computes unique signatures of histograms of orientations. A spherical support structure is used to encode information about the topology (surface). For the final descriptor, all the local histograms are stitched together. SHOT is rotational invariant and robust to noise and clutter. It provides unique and unambiguous 3D features, while being computational efficient. As we are using cameras, we employ the extended descriptor CSHOT [3], which additionally include texture information to improve the accuracy of SHOT with limited impact on the computational effort. We cache the feature descriptors for our keypoints, since the corresponding final submaps will not change afterwards. Only the submap origins will be subject to optimization within the graph SLAM algorithm. The aforementioned initial preprocessing steps thus have to be computed only once for each submap and can be used for matching several times as part of different pairs of submaps.

The actual matching can easily be parallelized as it is a read-only operation on the cached submap data. We search for correspondences between the submaps, applying the Euclidean distance function to determine the similarity of two keypoints. We assume descriptors with a maximum distance of 0.2 as potentially describing the same geometric feature for the next step. In order to dismiss outliers from the set of matching keypoints, we utilize Random Sample Consensus

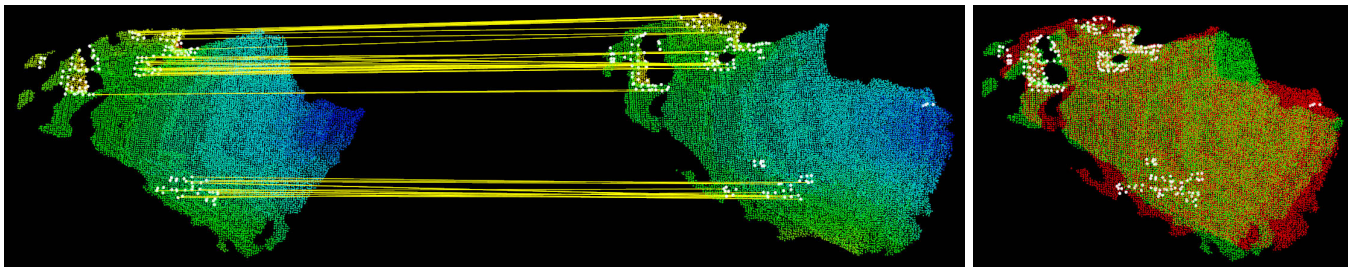

Fig. 3. Left: Keypoints (white) and correspondences (yellow) between two submaps. Right: Resulting final submap alignment after ICP optimization.

(RANSAC) on the resulting transformation estimates. As a final step, we perform Hough voting on the remaining pairs of keypoints, which is a 3D correspondence grouping that can deal with occlusions and clutter [28]. The algorithm allows for multiple instances of the model to be found. A relative position w.r.t. the model centroid is associated to each feature. Evidence for possible centroid positions in the current scene is accumulated with each corresponding feature casting a vote in a 3D Hough space. By associating each correspondence with a local reference frame, the remaining three degrees of freedom are taken into account. In particular, this method is effective for real-time stereo setups providing very noisy 3D data [28]. In case that more than one model is found within the estimated covariance bounds, we choose the one that exhibits the most correspondences. In Figure 3, we present an example for a successful match.

4) Full Pointcloud ICP Optimization: For a final refinement of the obtained 6D transformation, we employ the Iterative Closest Point (ICP) algorithm to minimize the metric matching error. ICP provides accurate results but requires close-enough initial estimates as input to avoid local minimums. The resulting transformation is estimated based on Singular Value Decomposition (SVD). The right image in Figure 3 shows the final alignment of two submaps after ICP optimization. In particular in environments where a large proportion of the map represents the ground plane, as for example an indoor floor, the ICP algorithm is able to significantly improve the transformation. As on the one hand such areas lack robust 3D features, on the other hand they work well for an ICP. By matching points on the ground, especially errors in roll, pitch and height can be compensated.

5) Outlier Filtering and Match Uncertainty Estimation: Before incorporating submap matches into the SLAM graph, it is important to filter false positives and matches that do not fit the corresponding variances. Already a single false connection between graph nodes can lead to map inconsistency and high pose errors. Although robust optimization methods exist [13], it is better to filter erroneous correspondences early on. We dismiss matches between pairs of submaps as outliers if their relative 6D transformation $\mathbf{t}^{n,m}$ exceeds the 2σ bounds of the difference of graph SLAM estimates for the submap origins $\mathbf{s}^n$ and $\mathbf{s}^m$. We compare the individual degrees of freedom as their uncertainties can differ greatly.

$$\Delta\mathbf{t}^{n,m} = \mathbf{t}^{n,m} \ominus (\mathbf{s}^m \ominus \mathbf{s}^n)$$
$$|\Delta t_\tau^{n,m}| < 2\sum(\sigma_\tau^n, \sigma_\tau^m) \quad \forall\, \tau \in \{x, y, z, roll, pitch, yaw\}$$

In particular the estimates for $roll$ and $pitch$ are very accurate due to the integration of IMU data, thus thresholds on these two angles yield valuable criteria for outlier rejection.

Second, we verify the resulting registration between submaps by validating the overlay of their local obstacle maps. Applying the estimated transform, we receive an overlapping region. Within this intersection, the arrangements of obstacles must be aligned very well if the right model has been found. We perform a nearest neighbor search for each obstacle point located within the submap intersection, employing the submap resolution as the maximum search radius. However due to occlusions and changes in viewpoint, not all obstacle keypoints can be aligned. If we find an alignment with a respective neighbor for more than 70% of all obstacle points, we assume the obstacle maps to be correctly aligned.

For the optimization steps done in the graph SLAM algorithm, it is essential to deliver an uncertainty measure for the estimated 6D transformation for each matched pair of submaps. Therefore, we estimate the variance for each of the six degrees of freedom from the root-mean-square error (RMSE) w.r.t. the nearest neighbor distances between the two aligned pointclouds:

$$\sigma_x = \sigma_y = \sigma_z = \mathrm{RMSE};\ \sigma_{roll} = \arctan\left(\frac{2\cdot\mathrm{RMSE}}{d_{yz}}\right)$$
$$\sigma_{pitch} = \arctan\left(\frac{2\cdot\mathrm{RMSE}}{d_{xz}}\right);\ \sigma_{yaw} = \arctan\left(\frac{2\cdot\mathrm{RMSE}}{d_{xy}}\right)$$

with $d_{yz},\ d_{xz},\ d_{xy}$ denoting the diameters of the overlap between the matched submaps in the respective planes.

D. Graph SLAM

For global optimization, we construct a graph containing the submaps as the only nodes, connected by their relative pose estimates. Thereby we differentiate two different types of edges in the graph. First, relative pose estimates between consecutive submap origins are available from our sensor fusion of visual odometry and IMU data, as described in Section III. Second, submap matches constitute additional connections between pairs of submaps that lead to loop closure constraints within the graph, as visualized in Figure 4. All edges are weighted by their estimated Gaussian uncertainty. We perform incremental, online graph optimization, employing the iSAM2 optimizer [12] from the GTSAM toolbox[2], which is freely available as open source software. By solely including submap origins as nodes, we construct a

[2]https://collab.cc.gatech.edu/borg/gtsam/

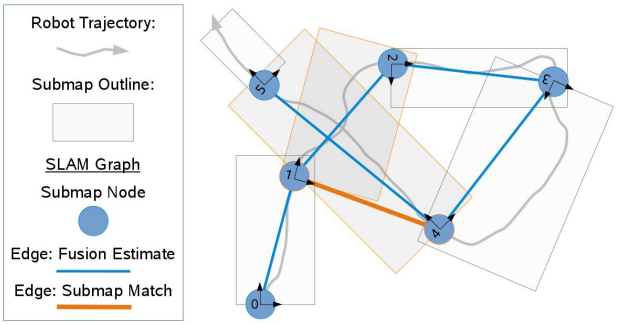

Fig. 4. Schematic of SLAM graph. The rectangles represent the submaps' bounding boxes. The two highlighted ones overlap and match, resulting in a corresponding edge in the graph. Through global graph optimization, the origins of all submaps are corrected on loop closures.

sparse graph, compared to SLAM approaches that include all robot pose estimates. The computational effort for its incremental optimization is at a negligible level compared to the processing of 3D pointclouds, which is performed in parallel. However, the graph can also easily be extended to include landmarks or object detections, GPS fixes and other relative or absolute low-frequency measurements that might be available. In [29] we extended this approach by introducing a novel graph topology to combine local reference filter estimates for online multi-robot SLAM.

V. Experiments

A. Robot Hardware Setup

For our experiments, we employed two different robots with similar sensor setup, our *Lightweight Rover Unit (LRU)*, see Figure 1, and a *Pioneer 3-AT (P3AT)* robot. We equipped both robots 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. On the P3AT we mounted Guppy F-080B cameras ($1/3$" chip size, resolution: 1032×778), on the LRU Guppy PRO F-125B cameras ($1/3$" chip size, resolution: 1292×964), both with $f = 5$ mm lens. Our computation stack consists of an Intel Core i7-3740QM CPU with 2.70 GHz and an additional Spartan 6 LX75 FPGA Eval Board, allowing us to perform dense stereo matching at 14.6 Hz with a resolution of 1024×508.

B. Experimental Scenarios

We evaluated our method in three different scenarios:

1) *Outdoor:* Unstructured environment with several types of gravel. It contains a small crater and rocks of different sizes, see Figure 1. Due to larger rocks and steep slopes, the crater can be entered only from one side. Robot: LRU, stereo framerate: 4.8 Hz.
2) *Indoor:* Lab environment with rooms and hallways, see Figure 7. Robot: P3AT, stereo framerate: 14.6 Hz.
3) *Mixed Indoor & Outdoor:* Our indoor scenario, extended by two loops around the building, see Figure 8. Robot: P3AT, stereo framerate: 14.6 Hz.

In all scenarios we used a 3D map resolution of 3 cm. For our outdoor tests, we purposely reduced the stereo framerate

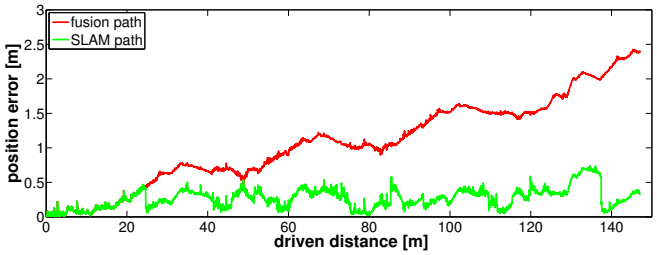

Fig. 5. 3D position errors w.r.t. ground truth in our outdoor scenario.

in order to highlight the robustness of our localization and mapping framework. We acquired ground truth position data through a Leica total station that tracks a prism attached to the robot. As the robot and the tachymeter use different reference systems, we performed an initial spatial (rigid transformation) and temporal path alignment. Therefore, we assume that the fusion error within the first 3 m is negligible and estimate the transformation of the corresponding trajectories by applying a least squares error minimization. In our indoor lab environment, we employed an Advanced Realtime Tracking (ART) system attached to the ceiling to receive ground truth for the robot trajectories. The tracking area however is limited to approx. $3\,\text{m} \times 4\,\text{m}$. We thus limit the evaluation for our indoor scenario to the partial trajectories for which ground truth is available (25 m out of 71 m). For the mixed indoor & outdoor scenario the ratio of the ground truth trajectory compared to the full path is very small. We therefore restrict the evaluation to the measured positions after each round and the final map quality, which depends heavily on the localization.

C. Results and Discussion

In this section we discuss the results of our experiments executed in the three aforementioned scenarios, which represent different challenges for our SLAM system. We compare the estimated SLAM paths to ground truth trajectories as well as to the fusion estimates, computed from visual odometry and IMU data. In addition, we perform a comparison of our novel approach with a 3D RBPF SLAM system from previous work [2]. It is important to note that for all of our evaluation results, we always use the sequentially logged 3D position estimates of the SLAM algorithm at each particular point in time, and not a afterwards fully optimized trajectory. Hence, before the first and in between submap matches, the SLAM trajectory is solely depending on the fusion estimate. This results in a larger overall trajectory error compared to a fully optimized path. However, for an autonomous robot, only the current estimate is available for navigation tasks at a particular point in time. We thus focus on this criterion for evaluation. Nevertheless, we achieve promising results compensating the drift of the fusion estimate, see Figure 5. The generated 3D maps presented in our evaluation however are based on the fully optimized graph.

1) Outdoor: In Figure 1 we show the final map (3 cm resolution) of the outdoor testbed, generated by our SLAM framework. We drove four rounds through the unstructured environment, including a passage through the crater. The map

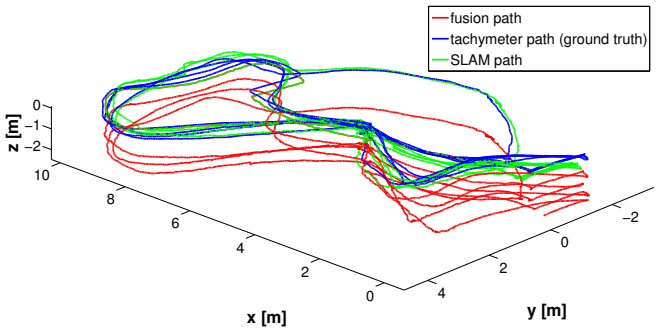

Fig. 6. Estimated robot trajectories in our outdoor scenario.

scenario		outdoor		indoor		mixed	
2D bounding box	[m]x[m]	11x17		8x13		25x37	
driven dist.	[m]	148.7		70.7		326.3	
mean linear vel.	[m/s]	0.355		0.304		0.329	
duration	[s]	652.8		295.2		1187.7	
num. of submap matches		12		10		16	
3D position error		**fusion**	**SLAM**	**fusion**	**SLAM**	**fusion**	**SLAM**
mean	[m]	1.10	0.26	0.32	0.13	-	-
std	[m]	0.62	0.14	0.17	0.05	-	-
rms	[m]	1.26	0.30	0.36	0.14	-	-
max	[m]	2.43	0.73	0.52	0.35	-	-
final	[m]	2.40	0.33	0.49	0.04	6.94	0.25

TABLE I

COMPARISON OF 3D LOCALIZATION ERRORS IN ALL THREE SCENARIOS (GROUND TRUTH FOR STATISTICAL EVAL. NOT AVAILABLE FOR MIXED SCENARIO)

represents an overlay of all generated submaps, positioned according to the graph SLAM pose estimates. All submaps visually appear correctly aligned to each other, indicating the accuracy of our SLAM system. The respective sequentially recorded robot trajectories are presented in Figure 6, which clearly shows the deviation of the fusion path from the actual robot trajectory. In Figure 5, we compare the resulting 3D position error (w.r.t. ground truth) of our SLAM approach with the fusion results. After a driven distance of more than 20 m, the first successful match is generated and incorporated as an edge into the SLAM graph. Up to this point, the SLAM estimate is equal to the fusion path. The correction of the position error highly depends on the quality of the matches and the locations of the matched pairs of submaps. Our results show that we achieve a strong improvement for the position estimate over the full distance. In Table I, we present an evaluation of the corresponding 3D trajectories w.r.t. ground truth. To avoid biases, we excluded consecutive measurements for time intervals, in which the robot did not move. In our outdoor scenario, we achieve a mean 3D position error of 0.26 m compared to 1.10 m for the fusion estimate. The final 3D position deviation is 0.22% w.r.t. the length of the full trajectory.

2) Indoor: Untextured and reflective surfaces, like white walls, as well as regular patterns, especially radiators, constitute challenges for the stereo algorithm in an indoor environment. This results in stereo mismatches and consequently in corrupted depth images and submaps. In order to test our system under such challenging circumstances, we drove three loops through a lab, including a floor passage. This corridor

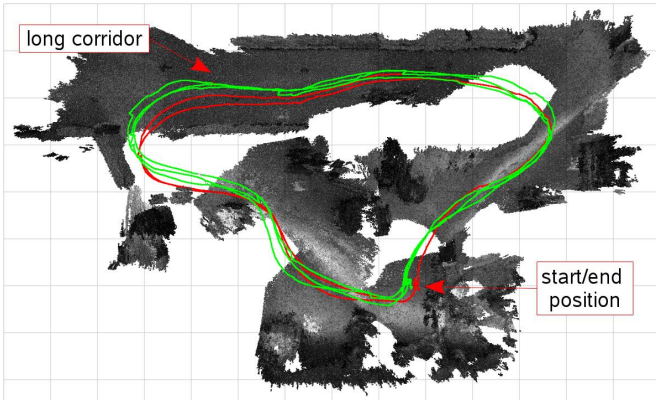

Fig. 7. Indoor run: Top-down view on final 3D map generated by our SLAM system (red: fusion path, green: SLAM path)

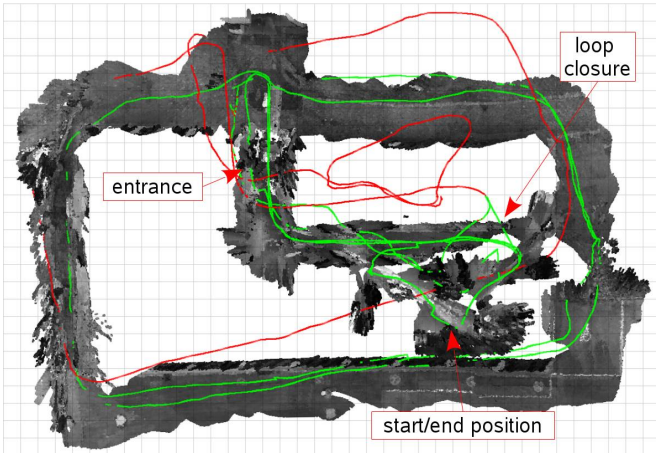

Fig. 8. Mixed indoor/outdoor run: Top-down view on final 3D map generated by our SLAM system (red: fusion path, green: SLAM path). All estimates refer to the sequentially logged data available at each particular point in time, not an afterwards fully optimized trajectory.

in particular constitutes tricky conditions, containing planar white walls and a partially reflective ground plane, i.e. few texture as well as almost no unique geometric features. In addition, jumps in the fusion estimate occur due to wrong stereo matches, In Figure 7 we present the indoor map generated by our SLAM system for this scenario. Despite all challenging conditions, the submap matching still provides good results and we receive a coherent 3D map. Compared to the ground truth acquired through our tracking system, we achieve a mean 3D trajectory error with our SLAM system of 0.13 m in contrast to 0.32 m for the fusion, as shown in the second column of Table I.

3) Mixed Indoor & Outdoor: In this scenario, we extended our indoor setup by driving two loops around the building. We thereby want to highlight the robustness of our approach, as it can cope with indoor and outdoor environments using the same set of parameters. We only evaluate the quality of the map as well as the final 3D positions after each round as the area of our tracking system is too small for meaningful evaluation of the full trajectory. After the first outdoor loop, the fusion and SLAM estimates result in the same position error of 3.17 m, since no submap match has been added to the SLAM graph yet. After the second loop and multiple submap matches, the final 3D position

outdoor scenario	2D position error [m]				
	mean	**std**	**rms**	**max**	**final**
3D RBPF SLAM (prev. work [2])	0.22	0.13	0.26	0.63	0.22
6D graph SLAM (Section IV)	0.16	0.10	0.19	0.41	0.08

indoor scenario	2D position error [m]				
	mean	**std**	**rms**	**max**	**final**
3D RBPF SLAM (prev. work [2])	0.14	0.08	0.17	0.36	0.19
6D graph SLAM (Section IV)	0.07	0.03	0.08	0.19	0.03

TABLE II

COMPARISON WITH A RBPF SLAM [2] IN DIFFERENT SCENARIOS

error of our SLAM framework is $0.25\,\mathrm{m}$ compared to a fusion error of $6.94\,\mathrm{m}$, see Table I. In Figure 8 we show the final map, after a driven distance of $326.3\,\mathrm{m}$, all created submaps are aligned according to their poses estimated by the graph SLAM algorithm. The outlined green path represents the sequentially logged SLAM estimates at each particular point in time, not a fully optimized trajectory. Hence, in areas where no loop-closures are incorporated it solely relies on the fusion estimate. Taking all three experiments into consideration, we have shown that our localization and mapping approach is capable to generate valid loop closures in both indoor, outdoor and mixed environments, resulting in consistent, globally optimized 3D maps.

4) Comparison to RBPF SLAM: Furthermore, we compared our submap based 6D SLAM approach with a 3D Rao-Blackwellized particle filter presented in our previous work [2] and show the results in Table II. We achieve an improvement of the mean 2D position error of $27\,\%$ and $50\,\%$ in the indoor and outdoor scenario respectively.

VI. CONCLUSION AND FUTURE WORK

In this work, we have presented a novel map matching technique for stereo-vision based submaps. We apply our previous work on local obstacle maps as one of multiple filtering steps in order to gain robust keypoints with discriminative geometric features for the matching process. We evaluated the localization accuracy of our novel submap matching pipeline within our SLAM framework. Therefore, we performed experiments in three different scenarios, thereby demonstrating its ability to achieve drift-free and accurate localization in previously unknown indoor, outdoor and mixed environments. In addition, we compare our novel approach to a 3D RBPF SLAM developed in previous work [2], showing a significant improvement on 2D localization accuracy. Furthermore, our approach generates high-resolution 3D maps ($3\,\mathrm{cm}$ voxel size) of the environment, containing both a full pointcloud for visualization and post-processing as well as the obstacle classification, which can directly be used for path planning. For future work, we plan to approach multi-robot scenarios that involve varying viewpoints as well as different sensors for the individual robots. We have shown that our novel approach to map matching already yields robust results w.r.t. changes in viewpoint and light conditions, for example in the mixed scenario. Another challenge for future research is the merging of submaps, once a good relative transformation estimate between them has been found. This is necessary to keep computational and memory requirements within a limited workspace independent of the runtime.

REFERENCES

[1] H. Hirschmüller, "Stereo Processing by Semi-Global Matching and Mutual Information," *IPAMI*, vol. 30, no. 2, pp. 328–341, 2008.

[2] C. Brand, M. J. Schuster, H. Hirschmüller, and M. Suppa, "Stereo-Vision Based Obstacle Mapping for Indoor / Outdoor SLAM," in *IROS*, 2014.

[3] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3d feature matching," in *ICIP*, 2011.

[4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 2005.

[5] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping : Part I," *RAM*, vol. 13, no. 2, pp. 99–110, 2006.

[6] T. Bailey and H. Durrant-Whyte, "Simultaneous Localization and Mapping (SLAM): Part II," *RAM*, vol. 13, no. 3, pp. 108–117, 2006.

[7] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *T-RO*, vol. 23, no. 1, Feb. 2007.

[8] B. D. Gouveia, D. Portugal, and L. Marques, "Speeding Up Rao-Blackwellized Particle Filter SLAM with a Multithreaded Architecture," in *IROS*, 2014.

[9] J. Welle, D. Schulz, T. Bachran, and A. B. Cremers, "Optimization Techniques for Laser-Based 3D Particle Filter SLAM," in *ICRA*, 2010.

[10] P. B. Quang, C. Musso, and F. Le Gland, "An Insight into the Issue of Dimensionality in Particle Filtering," in *FUSION*, 2010.

[11] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A General Framework for Graph Optimization," *ICRA*, 2011.

[12] 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.

[13] Y. Latif, C. Cadena, and J. Neira, "Robust Graph SLAM Back-ends: A Comparative Analysis," in *IROS*, 2014.

[14] 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.

[15] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An Evaluation of the RGB-D SLAM system," in *ICRA*, 2012.

[16] R. C. Leishman, T. W. McLain, and R. W. Beard, "Relative Navigation Approach for Vision-Based Aerial GPS-Denied Navigation," *ICUAS*, 2013.

[17] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-Time Dense Surface Mapping and Tracking," *ISMAR*, 2011.

[18] R. Reid and T. Bräunl, "Large-scale Multi-robot Mapping in MAGIC 2010," *RAM*, pp. 239–244, 2011.

[19] S. Carpin, "Fast and accurate map merging for multi-robot systems," *Autonomous Robots*, vol. 25, no. 3, pp. 305–316, July 2008.

[20] M. Labbé and F. Michaud, "Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM," in *IROS*, 2014.

[21] X. Li and I. Guskov, "Multiscale features for approximate alignment of point-based surfaces." in *Symposium on Geometry Processing*, 2005.

[22] K. Yousif, A. Bab-hadiashar, and R. Hoseinnezhad, "Real-Time RGB-D Registration and Mapping in Texture-less Environments Using Ranked Order Statistics," in *IROS*, 2014.

[23] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *ECCV*. Springer, 2010.

[24] H. Hirschmüller, P. Innocent, and J. Garibaldi, "Fast, Unconstrained Camera Motion Estimation from Stereo without Tracking and Robust Statistics," in *ICARCV*, 2002.

[25] H. Hirschmüller, "Stereo Vision Based Mapping and Immediate Virtual Walkthroughs," Ph.D. dissertation, De Montfort University, 2003.

[26] K. Schmid, F. Ruess, M. Suppa, and D. Burschka, "State Estimation for highly dynamic flying Systems using Key Frame Odometry with varying Time Delays," *IROS*, 2012.

[27] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *RAS*, vol. 56, no. 11, pp. 927 – 941, 2008.

[28] F. Tombari and L. Di Stefano, "Object recognition in 3d scenes with occlusions and clutter by hough voting," in *PSIVT*, 2010.

[29] M. J. Schuster, C. Brand, H. Hirschmüller, and M. Suppa, "Multi-Robot 6D Graph SLAM Connecting Decoupled Local Reference Filters," in *IROS*, 2015.