

Map Management Approach for SLAM in Large-Scale Indoor and Outdoor Areas

Simon F. G. Ehlers, Marvin Stuede, Kathrin Nuelle and Tobias Ortmaier

Abstract— This work presents a map management approach for various environments by creating multiple maps with different SLAM (simultaneous localization and mapping) configurations. A modular map structure allows to add, modify or delete maps without influencing other maps of different areas. The hierarchy level of our algorithm is above the utilized SLAM method, since it is able to automatically trigger new maps (e.g. after the detection of passing a doorway). The appropriate SLAM configuration for the next map is chosen by evaluating laser scan data. Single independent maps are connected by link-points which are located in an overlapping zone of both maps, enabling global navigation over several maps. Loop-closures between maps are detected by an appearance-based method using feature matching and ICP registration between point clouds. The number of possible loop-closure locations is limited to the number of link-points. Based on the arrangement of maps and link-points, a topological graph is extracted for navigation purposes and tracking the global robot's position over several maps. Our approach is evaluated by mapping a university campus with multiple indoor and outdoor areas and abstracting a metrical-topological graph. It is also compared to a single map running with different SLAM configurations. Our approach enhances the overall map quality compared to the single map approaches by automatically choosing appropriate SLAM configurations for different environmental setups.

I. INTRODUCTION

SLAM is a large field of research in mobile robotics [1] with many approaches for different sensor types (e.g. range sensors or cameras) and different strategies [2][3]. However, most SLAM algorithms generate a map based on a specified sensor data and parameter configuration, which has to be specified by a human supervisor in advance. This configuration is then suitable for a specific environment, but once the environment changes while mapping, for example an indoor/outdoor change, the SLAM algorithm could produce suboptimal results. Furthermore, mobile robots operating in very large buildings like airports or hospitals with small and large rooms, corridors and outdoor areas, creating a single map with one specific sensor and parameter configuration seems not reasonable. Also, navigation over multiple floors connected by an elevator is hard to solve by a single map approach. This paper addresses this issue by presenting a mapping approach, which automatically detects new rooms or an indoor/outdoor change by recognition of a doorway passing and evaluation of range sensor data. After the robot enters a new room/environment, a new map is created. In addition, the algorithm is able to choose automatically the

The authors are with the Institute of Mechatronic Systems, Faculty of Mechanical Engineering, Leibniz University Hannover, D-30167 Hannover, Germany simon.ehlers@stud.uni-hannover.de, marvin.stuede@imes.uni-hannover.de

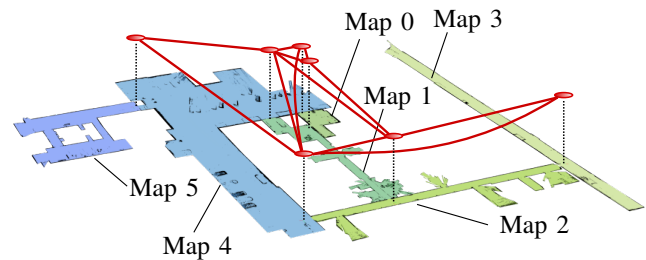


Fig. 1. Metrical-topological graph structure from the map manager approach: The creation of new maps is automatically triggered after door passing. Maps are then connected by link-points (red circles). Link-points are connected by edges (red lines). The greenish and blueish (on the left side) maps are generated with different SLAM configurations.

optimal sensor- and SLAM configuration for mapping the new environment. Our algorithm is not a SLAM approach but a map management algorithm to autonomously organize and arrange maps in a topological-metric form which are created by common SLAM algorithms, as presented in Fig. 1. The modular map structure makes it possible to maintain the map for every room or outdoor area independently. The requirements for the map management system are segmentation of certain areas/rooms by detecting doors, choosing an appropriate SLAM configuration for the environment, finding loop closures with existing maps, and abstracting a metrical-topological graph.

A. Related Works

To create a metrical-topological graph, methods based on sub-mapping techniques and appearance-based SLAM methods [3] are often utilized. Sub-mapping SLAM approaches create local metrical maps of the environment which are connected by a topological graph [4][5][6]. In [4], the topological graph is created by using the origins of the local metric maps as vertices and the transformations between local map origins as edges, to avoid the propagation of uncertainty of the positions between the local maps. In a hierarchical SLAM approach, loop closures are found efficiently on the global (topological) level [5]. A computationally efficient navigation method through large-scale metric-topological maps is presented in [6]. This is done by path planning on the topological level while only the current local metrical map is used for the occupancy grid. However, the local maps are created in similar environments, for example rooms/offices or large corridors, positioned directly next to each other and limited by e.g. size or number of features in these approaches.

To detect the start of a new map automatically, methods for online room segmentation are necessary. Ekvall et al. [7] combine a classical SLAM method with object detection. For labeling certain rooms, map partitioning was implemented that automatically detects doors while mapping is performed based on laser scan data. Wurm et al. [8] use a room segmentation detecting doors based on a Voronoi graph during mapping with multiple robots. Due to the room segmentation, redundant mapping of areas can be reduced to optimize the mapping-time. A different approach for creating sub-maps and abstracting a metric-topological hierarchy is to use spectral clustering and machine learning techniques, as presented by Brunskill et. al [9]. But in contrast to the other approaches, the locations of segmentation are doorways and additionally corners. Room segmentation and place categorization techniques are summarized in [10]. However, all the segmentation approaches presented above have in common, that after a door was detected, the mapping continues with the same configuration and parameters.

A multi mapping approach is presented by Stricker et al. [11] for a mobile service robot that guides persons through a multi-story building. A metrical-topological map structure with multiple maps is used. A Node represents specific locations and is organized in multiple hierarchies, where the floors on the highest topological level can be subdivided in sub-nodes like aisles or corridors which again consist of sub-nodes (single rooms). The nodes are connected by gateways (e.g. doors or elevators). However, due to the multiple hierarchy structure, the navigation becomes more complicated. Also, the method was only used indoors and does not allow multiple SLAM configurations.

Our approach overcomes the previously mentioned limitations by segmenting the area, which can be indoor and outdoor, in multiple maps with its appropriate SLAM configuration connected by link-points. The link-points are arranged in a simple topological graph which allows us to use the Dijkstra shortest path algorithm [12] for later navigation, path planning and global position tracking purposes. We categorize this approach to the field of semantic mapping, as discussed in IV.

B. Robot System for Evaluation

The mobile robot used for evaluation is designed as an information and guiding service system for an university campus. The tasks are providing information about buildings, faculty members and events as well as a guiding visitors to appropriate locations [13]. Since the operational area of the robot includes large outdoor areas as well as indoor areas of several buildings with multiple floors, an eligible multi-map management algorithm¹ has to be implemented and validated with respect to efficient long-term and large-scale operation capabilities. Fig. 2 shows the ROS-based robot with the used sensor systems.

The remainder of this paper is structured as follows: In section II we present our method for mapping in different

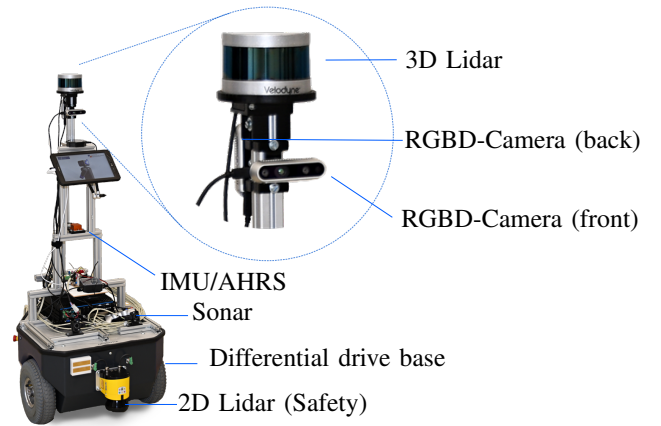


Fig. 2. Guiding robot with the base (Neobotix MP-500) equipped with a 2D laser scanner (SICK) and five ultrasonic sensors for collision avoidance. On top of the base, an IMU/AHRS (Attitude Heading Reference System, XSense MTi-30), two RGB-D-Cameras (Intel Realsense D435, front and back) and a 3D LiDAR (Velodyne VLP-16) are connected to the main computer.

environments. The experimental evaluation is shown in section III. Finally, the discussion and conclusion are presented in section IV and V.

II. MODULAR MULTI-ENVIRONMENT MAPPING

Since our approach is not a SLAM method but a mapping management on a higher level, the maps can be created by different SLAM methods. Theoretically, even the SLAM methods between each map can vary. The presented method is applicable to all types of mobile robots which are equipped with one front and rear RGB-camera as well as range sensors covering the front and rear (2D or 3D).

A. Graph Structure

Generally, our map management creates a graph structure on the highest level that can be described as a hybrid map $H = \langle \mathcal{M}, \mathcal{L}, \mathcal{E} \rangle$ [14]. $\mathcal{M} = \{M_0, \dots, M_n\}$ is a set of Maps, where a single map represents one room, corridor or outdoor environment separated by doors. Adjacent maps are connected by link-points $\mathcal{L} = \{L_0, \dots, L_m\}$, which are the graph nodes and placed in front or behind doors in an area where both maps slightly overlap. Each link-point L_i is a pair of $\langle \mathbf{m}_j, \mathbf{m}_k \rangle$ where \mathbf{m}_j and \mathbf{m}_k represent poses on the maps M_j and M_k (with $j \neq k$) and are positioned in the overlap zone of both maps, so $\mathbf{m}_j, \mathbf{m}_k \in M_j \cap M_k$. Both poses are equal relative to the global frame: ${}_{(0)}\mathbf{m}_j = {}_{(0)}\mathbf{m}_k$. In the nodes, the poses ${}_{(j)}\mathbf{m}_j = {}_{(j)}(x_j, y_j, \theta_j)^T$ and ${}_{(k)}\mathbf{m}_k = {}_{(k)}(x_k, y_k, \theta_k)^T$ of the link-point L_i relative to the origins of $(CF)_j$ and $(CF)_k$ of adjacent maps are saved. The link-points are connected by edges $\mathcal{E} = \{E_0, \dots, E_p\} \subseteq \mathcal{L} \times \mathcal{L}$. If two link-points (e.g. L_f and L_h) share the same map, $\{L_f, L_h\} \in M_i$, the link-points are connected. As a result, all link-points on the same map are fully connected. Note that one link-point belongs to two different maps.

¹The code to this work is provided at github.com/friedrichgerhard/map_manager

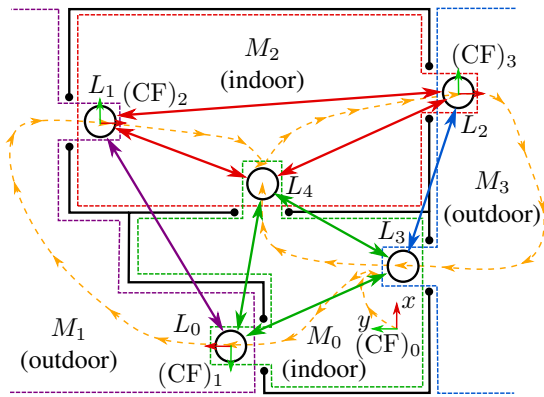


Fig. 3. Principle topological structure with link-points L_i on four different maps. The colored dashed lines represent the different maps M_i created by a SLAM method. At every link-point (and doorway) is an overlap between the maps. The color of edges emphasizes the map it corresponds to. The dashed orange line shows the path the robot traveled to map the area.

The resulting simple topological structure, shown in Fig. 3, enables the use of the Dijkstra shortest path algorithm for later navigation or global tracking purposes.

B. Door and Environment Appearance Detection

To detect the beginning of a new map, the robot needs to recognize passing a doorway. In our approach, three criteria are considered where at least two have to be fulfilled to detect the door robustly, namely (1) front passway detection, (2) back passway detection and (3) change in wall distances. For passway detection, the spatial derivative of the laser scan ranges with respect to the laser scan angle is calculated. When the robot stands in front of an opened door, two characteristic derivations within a specific range and angle relative to the robot can be detected (see Fig. 4 a) to fulfill criterion (1). After passing the door, again two high values of the derivative can be detected in the back (criterion (2), see Fig. 4 b) within a certain time and distance to the first detection from the front. Most doors fulfill criteria (1) and (2), resulting in a successfully door detection. However, some doors are close to orthogonal walls, which results in a small derivative in laser scan data. To enhance the robustness of door detection, criterion (3) is introduced: The 360° laser-scan data are average-low-pass filtered and transformed to the map frame. So the rough shape of the room is available and invariant regarding the robot's orientation. Then a fictive bounding box is placed around the scan data by extracting the minimum x_{\min}^t and maximum x_{\max}^t values relative to the map origin $(CF)_i$. If the derivative of the values changes over a specific time, criterion (3) is fulfilled and can replace either the door front or back detection.

After the detection of a door passing and before starting mapping the new map, the SLAM configuration for the new map has to be determined. Therefore, the low-pass filtered range data are used: If three of the four min/max values x_{\min} , x_{\max} , y_{\min} and y_{\max} of the range data are over a specific threshold of 15 m, the configuration for large rooms and high distances is used. Other possible criteria are discussed in IV.

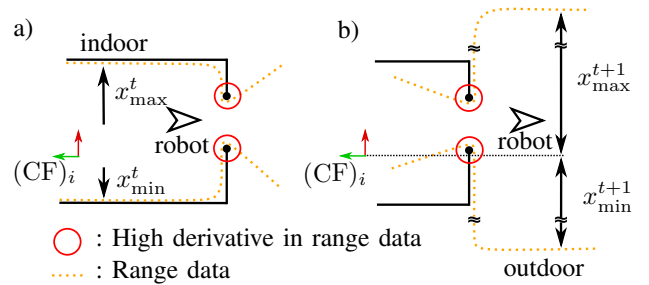


Fig. 4. Criteria to detect a door passing and an environment change: In a) the door is detected by the derivative at the doorposts (criterion 1). While the robot is inside the room the min/max range data x_{\max}^t and x_{\min}^t relative to $(CF)_i$ are low. After the robot passed the door in b), the doorpost can be detected again in the back (criterion 2). In addition, the values x_{\max}^{t+1} and x_{\min}^{t+1} are very large now. So the derivative of x_{\max} and x_{\min} with respect to time becomes very large. Criterion 3 is fulfilled, too. Note that in this example only the x values are shown for simplicity. Generally, y values are considered as well. Also, note that the detection of doorposts and the determination of min/max range data are not processed with exactly the same range data. The data for detecting the min/max values are additionally average-low-pass filtered.

C. Appearance-Based Loop Closure Detection

The appearance-based detection of loop closures in our approach is inspired by algorithms of appearance-based mapping where loop closures are found either by evaluating global features, local features (used here) or bag-of-word (BoW) methods [15][16].

To detect a topological loop closure, it is necessary to mark all the locations (link-point candidates) for a potential link-point with another map by placing the robot in front of an opened door, to detect the doorposts. Then the rear camera takes an image of the environment and the range data is saved as a point cloud of the link-point. When a new link-point and a new map are created, the cameras record the environment with one picture on each side of the door. Once the robot passes a door, all link-points and link-point candidates are checked for loop closure. This is done by feature matching between the current camera image and the saved ones in the link-point node. As feature-detector-descriptors SIFT [17] and/or A-KAZE [18] are used. The application of other descriptors is possible, depending on the environment and use case [19]. A loop closure is detected when the number of matching features exceeds a threshold of 30 matches. To improve robustness to noise, multiple images are used for feature matching and the median of all matching results is taken.

If a loop closure with an already existing link-point occurs, the stored map is loaded and the position of the link-point is given as the initial pose guess for the map to the SLAM method. If the loop closure occurs with a link-point candidate, the current point cloud is registered to the saved point cloud by ICP (iterative closest point) algorithm [20]. The transform between both point clouds is used to set the link-point at the current position of the robot in the coordinate frame of the old map. After a new link-point has been created and the topological connection was added to the graph, the corresponding link-point candidate is deleted.

By segmenting the maps into rooms or outdoor areas using doorways, the number of potential locations for topological loop-closures is very limited, which makes the loop closure detection efficient.

D. Global Position Tracking and Topological Graph Optimization

To increase the efficiency of the appearance-based loop closure detection, it is possible to track the global robot position over multiple maps and only consider the link-points and link-point candidates that are within a certain search radius r around the robot (see Fig. 5). For this, it is necessary to compute a metrical relation between the maps. The poses of the link-point relative to the map-origins are saved in the graph nodes, respectively. Thus, it is possible to compute the transformation between both map-origins. Let L_i be the link-point which connects map M_i and map M_{i+1} , and ${}^i\mathbf{T}_{L_i}$ is the homogeneous transform matrix between the origin $(\text{CF})_i$ of map M_i and L_i as well as ${}^{i+1}\mathbf{T}_{L_i}$ describes the transform between $(\text{CF})_{i+1}$ of map M_{i+1} and L_i . The transformation between both origins can be calculated as

$${}^i\mathbf{T}_{i+1} = {}^i\mathbf{T}_{L_i} ({}^{i+1}\mathbf{T}_{L_i})^{-1}. \quad (1)$$

The transform between the robot pose ${}_{(k)}\mathbf{p}_r = {}_{(k)}(x_r, y_r, \theta_r)^T$ on map M_k to the world origin $(\text{CF})_0$, which is usually the origin of map M_0 , can be determined by

$${}_{(0)}\mathbf{p}_r = {}^0\mathbf{T}_k {}_{(k)}\mathbf{p}_r = {}^0\mathbf{T}_{v(1)} \dots {}^{v(l-1)}\mathbf{T}_k {}_{(k)}\mathbf{p}_r, \quad (2)$$

where $v \in \mathbb{N}^l$ is a vector of length l which contains the map-indices on the shortest path to $(\text{CF})_k$. The shortest path is computed by the Dijkstra algorithm based on the graph structure presented in sec. II-A. The algorithm returns the sequence of link-points being used to compute the transformation between the map origins (see eq. 1) to have minimal uncertainty. The robot pose in the local map ${}_{(k)}\mathbf{p}_r$ is provided by the SLAM algorithm.

Although, the global robot position is only used for detecting loop closures during mapping, it is important to consider the propagation of uncertainties in the global positioning of maps. To estimate the uncertainty of the link-point position, either an estimation provided by the SLAM algorithm or an empirical estimate can be used in form of a covariance matrix. The uncertainty for every transform is computed by the method in [21]. The search radius r for finding eligible link-point candidates is adaptive to the uncertainties of the robot position:

$$r = r_{\text{orig}} + g_{\text{max}} \quad \text{with} \quad g_{\text{max}} = 2\sqrt{5.991\lambda_{\text{max}}}. \quad (3)$$

The value g_{max} is the length of the largest major axis of the 95% confidence ellipse with the largest eigenvalue λ_{max} of the covariance matrix [22]. The addition of g_{max} to the original search radius r_{orig} ensures a safe estimation of the new search radius. Note that solely x - and y -values are considered for the uncertainty in position.

Once a topological loop closure was newly detected, it is possible to correct the global positions of the link-points by a

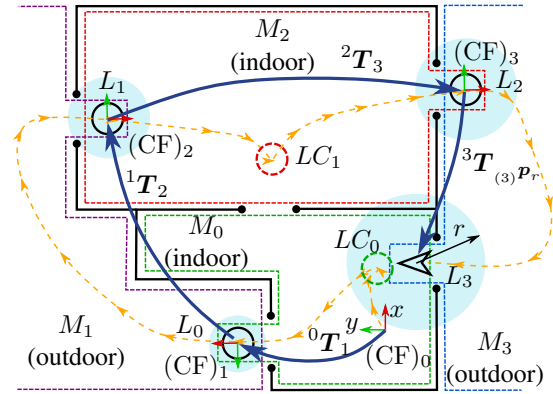


Fig. 5. Visualization of the global tracking of the robot's position to find eligible loop-closures. In this case the loop-closure between map M_3 and M_0 with link-point candidate LC_0 is detected. Since only one topological connection between the robot and world origin exists yet, the global pose has to be calculated by composing transformations from $(\text{CF})_0$ to $(\text{CF})_1$, $(\text{CF})_2$ and $(\text{CF})_3$ to ${}_{(3)}\mathbf{p}_r$. The search radius for loop-closure candidates (light blue circle) becomes larger the higher the uncertainty is. In this example only the link-point candidate LC_0 is checked for an appearance-based loop closure, since it is the only one within the search radius. This yields a more efficient and robust loop closure detection. The graph shows a previous time step to the graph presented in Fig. 3.

graph optimization algorithm. In our approach, g2o (general framework for graph optimization) [23] is used, where the robot positions are the link-point nodes and the observed landmark is the link-point candidate that the loop closure was detected with. Thus, the error due to uncertainty propagation can be reduced.

III. EXPERIMENTAL EVALUATION

In the present implementation, the SLAM method RTAB-Map (real-time appearance-based mapping) [24] is used in ROS, since it supports several camera sensor types (e.g. stereo, RGB-D) and also 2D and 3D Lidar. Furthermore, it is suitable for long-term and large-scale mapping due to its advanced memory-management [15]. In addition, RTAB-Map supports multi-session mapping [25] which allows us to simply continue mapping on an already existing map at another time. This perfectly fits in our philosophy of maintaining the modular map structure. However, RTAB-Map needs to be configured for each type of environment differently. The most important settings and differences between both configurations are presented in table I. Generally,

TABLE I
RTAB-MAP CONFIGURATIONS FOR EXPERIMENTAL EVALUATION

Parameter/Method	Configurations	
	Small Range (1)	High Range (2)
Depth source	Realsense D435	Velodyne VLP-16
Vis/FeatureType	GFTT/BRIEF	SIFT
Grid/RangeMax	10 m	30 m
Grid/MaxObstacleHeight	1.7 m	3.5 m
Mem/STMSize	30	60
Reg/Force3DoF	true	true
Grid/FromDepth	false	false
Rtabmap/TimeThr	750 ms	750 ms

both configurations have in common that 2D-SLAM is used (Reg/Force3DoF) and a 750ms limit for computation time (Rtabmap/TimeThr) is set to ensure real-time capabilities. For the high range configuration 2, the maximum distance of range data (Grid/RangeMax) considered for mapping is larger to detect obstacles at higher distance which is not necessary indoors and saves memory. Also the maximum obstacle height and STM size (short term memory for loop closure detection) is configured for higher distances and larger maps. The biggest difference is the source of the depth image for appearance-based loop closure detection. Since the RGB-D camera (Realsense D435) has only good performance for a distance under 7 meters and suffers from sunlight [26], the depth image is reconstructed from the point cloud of the 3D lidar for the high range configuration. The points of the lidar are projected into the camera sensor frame. Continuous depth information is generated by interpolating between the single points, resulting in a small resolution. However, features can be extracted from large distances up to 100 m (e.g. house walls) which enhances the appearance-based loop closure detection.

In the following evaluation, the decision which configuration to use for a new map is only based on the criterion presented in sec. II-B. For both configurations, the odometry source is an extended Kalman filter (EKF) which fuses the wheel odometry with IMU data [27]. Our approach is presented by mapping the ground floor of an university building. In the first subsection, the focus is on the graph structure and the configuration settings. In the second subsection, our method is compared to the single map approach with RTAB-Map in the small and high range configuration.

A. Modular Mapping of an university campus

For evaluation, part of the university campus was mapped with our map management approach and RTAB-Map as the SLAM method. In total, six maps were created including two buildings and two outdoor areas presented in Fig. 6 a). Every map has its own color, where the greenish maps are based on configuration 1 (small distances) and the blueish maps are created with configuration 2 (high distances). The link-points $L_0 \dots L_6$ are marked with red circles, the map origins (CF) $_0 \dots (CF)_5$ are also visualized by coordinate frames. A total distance of 1410 m was traveled while mapping the environment. All doors and environment changes were detected automatically. Even a glass door with metal frame at L_4 was detected by the door detection based on the derivatives. Also the SLAM configuration was chosen autonomously before starting a new map. Note, that the configuration setting does not distinguish between indoor and outdoor. Although, M_3 is an outdoor map (narrow corridor between two buildings), the algorithm decided to use configuration 1, since the distances to the walls left and right are small and features can be extracted precisely using the depth image from the RGB-D camera. In contrast, map M_5 is an indoor map (large entrance hall of a multi-story building), which was mapped with configuration 2 for large distances using depth information based on the 3D Lidar.

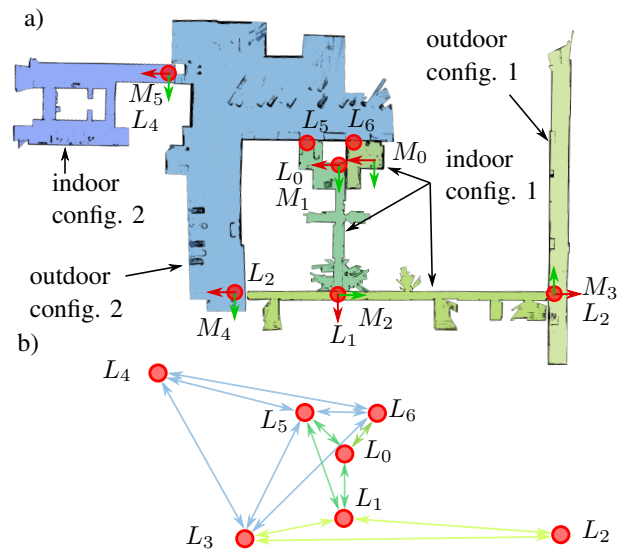


Fig. 6. a): Part of the university campus mapped with our map management approach. The greenish maps are created based on configuration 1 (small distances) and the blueish with configuration 2 (high distances). b): Topological graph structure created based on the map structure in a). The colors of the edges correlate with the maps.

This demonstration shows the mapping results of a large area with indoor and outdoor maps of different shape and appearance using our approach. In addition, maps can be added, modified or deleted later easily. The prerequisites for efficient navigation over multiple maps are given by connecting the maps over link-points in an overlap-area.

B. Comparison to Single Map Approach

Our multi-map approach is compared to the single-map approach using only RTAB-Map. For this, a large loop was traveled twice with an approximated distance of 506 m to detect loop closures with RTAB-Map. In total, three different mapping approaches were conducted on the same data. Our multi-map approach presented in Fig. 7 a), and two single-map approaches with RTAB-Map using only configuration 1 (see Fig. 7 b) or configuration 2 (see Fig. 7 c). Generally, all approaches in Fig. 7 show a consistent map. However, some advantages due to the separation into multiple maps and the usage of multiple SLAM configurations occur. While the mapped indoor area in Fig. 7 b) has about the same map quality compared to Fig. 7 a), mapping the outdoor area with configuration 1 leads to a smaller map with less range. The depth image from the RGB-D camera used is not suitable for the large distances to objects outside. As a result, less loop closures can be found outdoors and the localization of the robot suffers, which can lead to map errors as presented in Fig. 7 b). Using configuration 2 in hallways, parts of the ceiling and door frames are detected as obstacles, as shown in Fig. 7 c). All these map errors can be solved by using an appropriate configuration for the environment, as presented in Fig. 7 a).

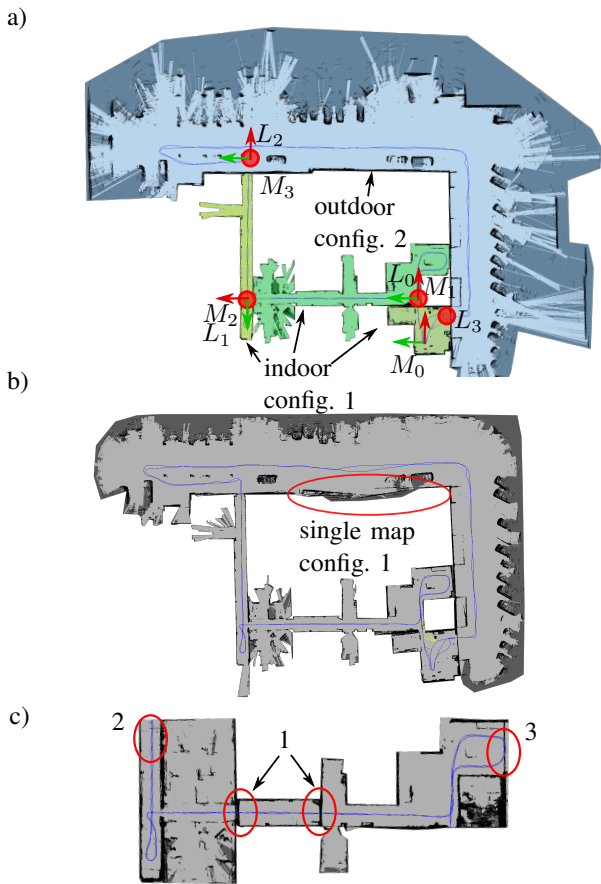


Fig. 7. a): Large loop with the introduced map managing approach. b): Large loop with a single map using RTAB-Map with configuration 1. c): Cropped single map from indoor environment mapped with configuration 2. For all three approaches the same data was used. The traveled path is visualized by the blue line. During mapping, the same path was traveled twice for finding loop closures with RTAB-Map. In b) a map error caused by inappropriate SLAM configuration is marked red. In c) the disadvantages of using the configuration 2 indoors is shown. Due to the maximum obstacle height of 3.5 m, the wall above doors and part of the ceiling are detected as obstacles (see 1 and 2). At position 3 the traveled path (RTAB-Map graph) collides with the wall which could be caused by a too complex graph optimization problem. The complexity of this optimization problem could be reduced by segmenting the area, as in a).

IV. DISCUSSION

In addition to choose the appropriate environmental SLAM settings, another advantage of our map management approach is the more efficient detection of large loop closures by our appearance-based method using link-point candidates at doors, rather than loading an already existing map to detect loop closures. Furthermore, we suppose that mapping with RTAB-Map benefits from using multiple small maps rather than one large map, since the number of nodes in the long-term memory not considered for loop closures is reduced on a smaller map. So, our approach gives RTAB-Map a preselection of eligible nodes for localization and loop-closing. However, only one criterion presented in sec. II-B can be used for determining the new SLAM configuration for a new map. As future work, multiple criteria could be

fused by machine learning algorithms, for example PSNR (peak signal-to-noise-ratio) and SSIM (structural similarity) [28] to detect the noise in RGB- and depth images or a CNN based on the Places365 data set [29] to categorize the environment (indoor/outdoor). Furthermore, a false detection of loop closures leads to a massive error in the topological graph structure. The loop closure detection could become more robust by additionally using other criteria like range data [30] in future applications.

Regarding the question whether our approach belongs to the category of semantic mapping or not: In [31], the minimal criteria for semantic mapping are defined as (i) the existence of a metric map, (ii) the ability to cognize the signification of places, (iii) the arrangement and abstraction of the mapped environment in form of a topological graph and (optionally) (iv) the usage of recognition methods to let the robot understand its environment in a "human compatible manner" [31]. In our understanding all these criteria are fulfilled, even though our approach differs from almost all other semantic mapping methods: We are not creating a single but multiple metric maps, which are arranged in a topological graph where the link-points, which connect the maps, are the nodes and the spatial transformations between the link-points are the edges. The semantic information about the certain places is stored in the metrical map itself, by determining which SLAM algorithm/sensor configuration is used for mapping (e.g. small/high ranges configuration), dependent on the appearance of the environment. So unlike many other semantic mapping approaches, the focus is not on human-robot interaction (HRI) [31] but on enabling the robot to map environments with different appearances to get optimal mapping results.

V. CONCLUSIONS

In this work, an approach for mapping large-scale environments with different SLAM configurations for various appearances of the environment is presented. By triggering new map creation and analyzing the environment after the automatically detection of doorway passing, the appropriate SLAM configuration is automatically chosen. The single maps are connected by link-points located in an overlapping zone of connecting maps. Loop-closures are detected by an appearance-based method using feature matching of images at link-point locations and ICP registration of point clouds. The number of possible loop-closure locations is limited to the number of link-points to make the loop-closure detection efficient. Based on the arrangement of link-points, a topological graph is extracted which can be used for navigation or tracking of the global robot's position over multiple maps.

For evaluation, a large area of a campus was mapped with our map management approach and directly compared to a single map approach. The results show that our approach enables a mobile robot to get appropriate mapping results for different types of environment by creating maps with different SLAM configurations. This makes it superior to single map solutions for large-scale areas regarding the mapping results.

REFERENCES

- [1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [2] S. Thrun *et al.*, "Robotic mapping: A survey," *Exploring artificial intelligence in the new millennium*, vol. 1, no. 1, pp. 1–35, 2002.
- [3] E. Zamora and W. Yu, "Recent advances on simultaneous localization and mapping for mobile robots," *IETE Technical Review*, vol. 30, no. 6, pp. 490–496, 2013.
- [4] M. Bosse, P. Newman, J. Leonard, and S. Teller, "Simultaneous localization and map building in large-scale cyclic environments using the atlas framework," *The International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, 2004.
- [5] C. Estrada, J. Neira, and J. D. Tardós, "Hierarchical slam: Real-time accurate mapping of large environments," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 588–596, 2005.
- [6] K. Konolige, E. Marder-Eppstein, and B. Marthi, "Navigation in hybrid metric-topological maps," pp. 3041–3047, 2011.
- [7] S. Ekvall, D. Kragic, and P. Jensfelt, "Object detection and mapping for service robot tasks," *Robotica*, vol. 25, no. 2, pp. 175–187, 2007.
- [8] K. M. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment," pp. 1160–1165, 2008.
- [9] E. Brunskill, T. Kollar, and N. Roy, "Topological mapping using spectral clustering and classification," pp. 3491–3496, 2007.
- [10] R. Bormann, F. Jordan, W. Li, J. Hampp, and M. Hägele, "Room segmentation: Survey, implementation, and analysis," pp. 1019–1026, 2016.
- [11] R. Stricker, S. Müller, E. Einhorn, C. Schröter, M. Volkhardt, K. Debes, and H.-M. Gross, "Interactive mobile robots guiding visitors in a university building," pp. 695–700, 2012.
- [12] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [13] M. Stüde, J. Wilkening, S. Tappe, and T. Ortmaier, "Voice recognition and processing interface for an interactive guide robot in an university scenario," *IEEE International Conference on Control, Automation and Systems, Accepted for Conference*, 2019.
- [14] P. Buschka and A. Saffiotti, "Some notes on the use of hybrid maps for mobile robots," pp. 547–556, 2004.
- [15] M. Labbé and F. Michaud, "Memory management for real-time appearance-based loop closure detection," pp. 1271–1276, 2011.
- [16] E. Garcia-Fidalgo and A. Ortiz, "Vision-based topological mapping and localization methods: A survey," *Robotics and Autonomous Systems*, vol. 64, pp. 1–20, 2015.
- [17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [18] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, 2011.
- [19] S. A. K. Tareen and Z. Saleem, "A comparative analysis of sift, surf, kaze, akaze, orb, and brisk," pp. 1–10, 2018.
- [20] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," vol. 1611, pp. 586–606, 1992.
- [21] J.-L. Blanco, "A tutorial on se (3) transformation parameterizations and on-manifold optimization," *University of Malaga, Technical Report*, vol. 3, 2010.
- [22] W. E. Hoover and M. Rockville, *Algorithms for confidence circles and ellipses*. Citeseer, 1984.
- [23] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," pp. 3607–3613, 2011.
- [24] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.
- [25] M. Labbé and F. Michaud, "Long-term online multi-session graph-based splam with memory management," *Autonomous Robots*, vol. 42, no. 6, pp. 1133–1150, 2018.
- [26] G. Halmetschlager-Funek, M. Suchi, M. Kampel, and M. Vincze, "An empirical evaluation of ten depth cameras: Bias, precision, lateral noise, different lighting conditions and materials, and multiple sensor setups in indoor environments," *IEEE Robotics & Automation Magazine*, vol. 26, no. 1, pp. 67–77, 2018.
- [27] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," July 2014.
- [28] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [29] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [30] K. Granström, T. B. Schön, J. I. Nieto, and F. T. Ramos, "Learning to close loops from range data," *The international journal of robotics research*, vol. 30, no. 14, pp. 1728–1754, 2011.
- [31] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86–103, 2015.