

Robust Exploration and Homing for Autonomous Robots

Daniel Perea Ström^a, Igor Bogoslavskyi^b, Cyril Stachniss^{b,*}

^aUniversidad de La Laguna, Departamento de Ingeniería Informática, Av. Francisco Sánchez, 38206 Tenerife, Spain.

^bUniversity of Bonn, Institute for Geodesy and Geoinformation, Nussallee 15, 53115 Bonn, Germany

Abstract

The ability to explore an unknown environment is an important prerequisite for building truly autonomous robots. Two central capabilities for autonomous exploration are the selection of the next view point(s) for gathering new observations and robust navigation. In this paper, we propose a novel exploration strategy that exploits background knowledge by considering previously seen environments to make better exploration decisions. We furthermore combine this approach with robust homing so that the robot can navigate back to its starting location even if the mapping system fails and does not produce a consistent map. We implemented the proposed approach in ROS and thoroughly evaluated it. The experiments indicate that our method improves the ability of a robot to explore challenging environments as well as the quality of the resulting maps. Furthermore, the robot is able to navigate back home, even if it cannot rely on its map.

Keywords: Exploration, Background knowledge, Homing, Navigation, SLAM

1. Introduction

Exploration is the task of selecting view points so that a robot can cover the environment with its sensors to build a map. The ability to robustly operate without user intervention is an important capability for exploration robots, especially if there is no means for communication between the robot and an operator. Most exploration robots always start assuming zero knowledge and do not exploit any background knowledge about the environment or typical environments. They build a map of the environment online and make all navigation decisions based on this map. As long as this map is consistent, the robot can perform autonomous navigation by planning the shortest path—for example using A*—from its current location to its next vantage point using the map. Although recent SLAM systems are fairly robust, there is a chance that they fail, for example, due to wrong data associations generated by the front-end. Even current state-of-the-art SLAM approaches cannot guarantee the consistency of the resulting map. Computing a path based on an inconsistent map, however, is likely to lead to a failure and possibly to loosing the robot if operating in a hazardous environment. Thus, exploring robots should always decide where to go next and at the same time verify if their map is still consistent (see sketch in Fig. 1). Considering existing approaches, however, it is fair to say that most exploration systems follow the paradigm that they (a) make their navigation and exploration decisions using the current map only and (b) assume that the map is consistent and thus can be used as the basis for path planning and navigation.

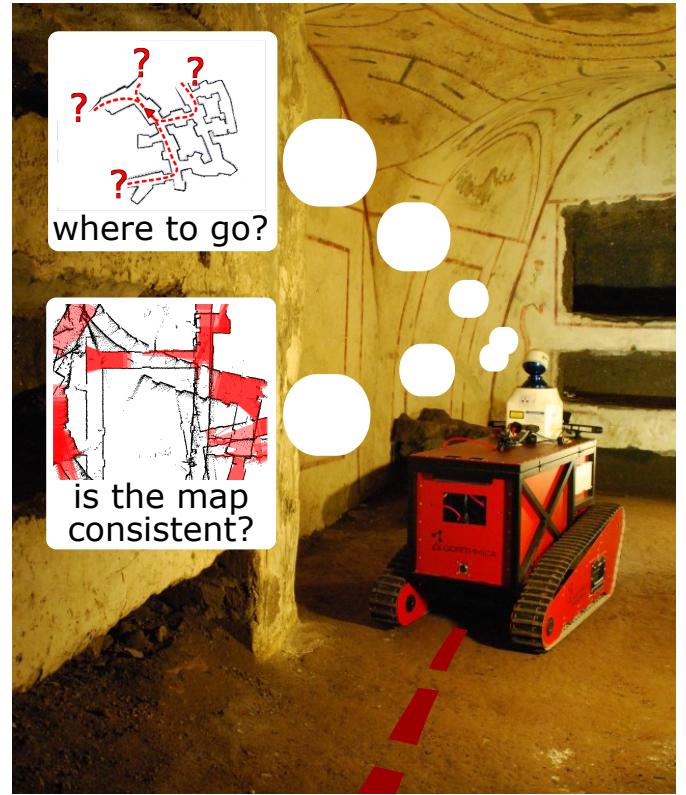


Figure 1: Mobile robot exploration has to answer the question: “Where to go next?”. Our approach exploits previously mapped environments to predict potential future loop closures and thus to select better target locations. When the statistical map consistency tester provides the robot with the information that the map is not consistent anymore the robot starts rewinding the trajectory using our robust homing method.

*Corresponding author

Email addresses: dani@isaatc.ul.es (Daniel Perea Ström), igor.bogoslavskyi@uni-bonn.de (Igor Bogoslavskyi), cyrill.stachniss@igg.uni-bonn.de (Cyrill Stachniss)

In this paper, we aim at relaxing these assumptions. The key idea is to consider the knowledge gained from previously conducted exploration missions to support the navigation system of the robot. This is motivated by the fact that selecting appropriate target locations during exploration supports the mapping process, and can increase the probability of building a consistent map. Furthermore, we want to be able to safely navigate our robot back to its starting location, even if the mapping process failed.

The first contribution of this paper is a novel approach to exploiting background knowledge while generating exploration behaviors to support mapping. The key idea is to use previously experienced environments to reason about what to find in the unknown parts of the world. To achieve this, we equip our robot with a database to store all acquired (local) maps and exploit this knowledge when selecting target locations. Our research is motivated by an exploration project for autonomously digitizing the Roman catacombs, which are complex underground environments with repetitive structures. To predict possible geometries of the environment the robot may experience during exploration, we exploit previously visited areas and consider the similarities with the area around the currently planned next view point. This allows the robot to actively seek for loop-closures and in this way actively reduce its pose uncertainty. Our experiments indicate that this approach is beneficiary for robots when comparing it to a standard frontier-based exploration method.

The second contribution is a robot homing approach with the goal of retrieving our robot even if the SLAM system failed to build a consistent map. To avoid that our robot gets lost, we propose a robust homing system consisting of two distinct parts. Part A performs a statistical analysis of the map and thus provides the information about its consistency. We build upon our previous work [1] for performing a cascade of pair-wise consistency checks using the observations perceiving the same areas. To avoid performing such checks on the overall map, we reduce the area to analyze by planning the shortest homing route for the robot assuming a consistent map. We then analyze the map consistency only along that path and can estimate on the fly if the map around this path is consistent or not with a given confidence level. If it is consistent, we navigate back on the verified homing path. Part B of our approach is responsible for driving the robot back to its starting location without a map. We achieve this by rewinding the trajectory that the robot took to reach its current pose. If the motions of the robot were perfect, i.e. would lead to the desired robot pose in the world frame, we would be able to simply invert the motion commands performed by the robot and could safely reach the starting location. Motion execution and odometry, however, are often noisy. As a result, simply following inverse motion commands will not bring the robot to the starting location in the real world in most cases. Therefore, we take into account the sensor information to guide the robot back by matching the observations with the past.

2. Related Work

The majority of techniques for mobile robot exploration focus on generating motion commands that minimize the time needed to cover the whole terrain. Several techniques also assume that an accurate position estimate is available during exploration [2, 3]. Whaite and Ferrie [4] present an approach that uses the entropy to measure the uncertainty in the geometry of objects that are scanned with a laser range sensor. Similar techniques have been applied to mobile robots [5, 6], but such approaches still assume to know the correct pose of the vehicle. Such approaches take the map but not the pose uncertainty into account when selecting the next vantage point. There are, however, exploration approaches that have been shown to be robust against uncertainties in the pose estimates [7, 8].

Besides the idea of navigating to the next frontier [3], techniques based on stochastic differential equations for goal-directed exploration have been proposed by Shen *et al.* [9]. Similar to that, constrained partial differential equations that provide a scalar field into unknown areas have been presented by Shade *et al.* [10]. An information-theoretic formulation that seeks to minimize the uncertainty in the belief about the map and the trajectory of the robot has been proposed by Stachniss *et al.* [11]. This approach builds upon the works of Makarenko *et al.* [12] and Bourgault *et al.* [13]. Both extract landmarks out of laser range scans and use an Extended Kalman Filter to solve the underlying SLAM problem. They furthermore introduce a utility function which trades-off the cost of exploring new terrain with the potential reduction of uncertainty by measuring at selected positions. A similar technique has been presented by Sim *et al.* [14], who consider actions to guide the robot back to a known place in order to reduce the pose uncertainty of the vehicle. Such information-driven techniques have also been used for perception selection to limit the complexity of the underlying optimization problems in SLAM [15].

In general, the computation of the expected entropy reductions is a complex problem, see Krause and Guestrin [16], and in all real world systems, approximations are needed. Suitable approximations often depend on the environment model, the sensor data, and the application. In some cases, efficient approximations can be found, for example in the context of monitoring lakes using autonomous boats [17].

Other approaches, especially in the context of autonomous micro aerial vehicles (MAVs), seek to estimate the expected feature density in the environment in order to plan a path through areas that support the helicopter localization [18]. This can be seen as related to information-theoretic approaches, although Sadat *et al.* [18] do not formulate their approach in this framework. A related approach to MAV exploration seeks to select new vantage points during exploration, so that the expected number of visible features is maximized, see Mostegel *et al.* [19].

An interesting approach by Fox *et al.* [20] aims at incorporating knowledge about *other* environments into a cooperative mapping and exploration system for multiple robots. This allows for predicting simplified laser scans of an unknown environment. This idea was an inspiration for our paper for predicting possible loop closures given the environment structure

explored so far. We use this approach for exploring ancient catacombs, which are repetitive underground environments, with a mobile platform, see Fig. 1. Chang *et al.* [21] propose an approach for predicting the environment using repetitive structures for SLAM. Other background knowledge about the environment, for example semantic information [22], can support the exploration process as shown by Wurm *et al.* [23], Stachniss *et al.* [24] as well as Holz *et al.* [25].

A central problem in robust exploration, however, is that in case of a SLAM failure, the map becomes inconsistent. This can prevent the robot from continuing its exploration mission and—even worse—from being able to navigate back. It is therefore important to be able to perform reliable navigation without relying on a map.

Sprung *et al.* [26] present a lidar-based teach-and-repeat method to follow a route given by the user. The approach relies on precise localization of the robot based on the lidar measurements with respect to a taught-in trajectory. Similarly, Furgale *et al.* [27] perform the ICP-based teach-and-repeat approach on an autonomous robot equipped with a high precision 3D spinning lidar. They extend the standard teach-and-repeat approach by adding a local motion planner to account for dynamic changes in the environment. Our method to rewind the trajectory is similar to the teach-and-repeat setup in this formulation. However, in contrast to the mentioned methods, we use a substantially less accurate robot and thus have to cope with somewhat larger deviation from the reference trajectory.

Vision methods are also popular for teach-and-repeat approaches. Furgale *et al.* [28] present a vision-based approach to teach-and-repeat for long range rover autonomy. During a learning phase, their system builds a manifold map of overlapping submaps as the rover is piloted along a route. The map is then used for localization as the rover repeats the route autonomously. They present an autonomous planetary rover that is able to navigate even non-planar terrain without relying on an accurate global reconstruction. Nitsche *et al.* [29] extend a standard teach-and-repeat approach by adding Monte Carlo localization to localize the robot with respect to the learned path. They present vision-based tests carried out both on a ground robot and an aerial drone. Battesti *et al.* [30] present an online localization approach. They use visual loop-closure techniques to create consistent topo-metric maps in real-time while the robot is teleoperated and localizes itself in such maps. This allows the robot to follow the predicted path successfully compensating the odometry drift. These visual methods, however, need substantial adaptation in order to be used in a setup similar to ours: using monocular cameras to localize through feature detection relies on having enough visual information, which is not the case in the typically dark catacombs. The work presented here is based on a conference publication [31], which described the idea of predictive exploration.

3. Robot and Sensor Setup

Our robot is a customized Mesa Element platform, see Fig. 1. It is equipped with a laser range finder scanning in a horizontal

2D plane around 60cm above the ground. The robot is additionally equipped with two ASUS Xtion depth cameras that observe the local area in front of the robot in 3D. Both cameras look forward, one slightly rotated to the left and the other one to the right with a minimal overlap in the middle. Our system relies on the 2D information for solving the exploration task in order to decide which parts of the scene have been explored, and where to move next. For the robust homing presented in Sec. 5, we take into account the 3D depth images from the Xtions as this allows for a more accurate alignment of the scans. Furthermore, a local traversability analysis is done in 3D based on the Xtions [32].

4. Environment Predictive Exploration

The central question in exploration is “Where to go?”. Several different cost functions for making the decision of where to go next can be defined. The most popular one goes back to Yamauchi [3], who guides the robot to the closest reachable unexplored location. Yamauchi introduces the concept of frontiers, which are the cells of an occupancy grid map at the boundary between the free and the unexplored space. In the standard setting, this approach seeks to minimize the time that is needed to cover the environment with the robot’s sensors and is a popular choice in mobile robotics. On the other hand, exploring hazardous environments requires trading time for a more robust navigation that supports the mapping system and avoids pose uncertainty.

4.1. Information-Driven Exploration

Given the fact that most real robots maintain a probabilistic belief about their pose and the map of the environment, an alternative approach is to select the target location that is expected to minimize the uncertainty in the belief of the robot. In this setting, the exploration problem can be formulated as follows. At each time step t , the robot has to decide which action a to execute (where to move next). During the execution of a , the robot obtains a sequence of observations z (for better readability, we neglect all time indices). Thus, we can define the expected information gain, also called mutual information, of selecting the action a as the expected change in entropy in the belief about the robot’s poses X and the map M :

$$I(X, M; Z^a) = H(M, X) - H(M, X | Z^a). \quad (1)$$

The second term in Eq. (1) is the conditional entropy and is defined as

$$H(M, X | Z^a) = \int p(z | a) H(M, X | Z^a = z) dz. \quad (2)$$

Unfortunately, reasoning about all potential observation sequences z in Eq. (2) is intractable in nearly all real world applications since the number of potential measurements grows exponentially with the dimension of the measurement space and with time. It is therefore crucial to approximate the integral



Figure 2: Example of the submap retrieval using FabMAP2. The left image shows the query map, the other ones the best four matches from the database.

of Eq. (2) so that it can be computed efficiently with sufficient accuracy.

A suitable approximation, however, depends on the environment model, the sensor data, and the application so that no general one-fits-all solution is available. In our previous work [11], we considered different types of actions: First, *exploration actions* that guide the robot to the closest frontier and reduce the map uncertainty. As we have no further information about the unseen area, it is difficult to distinguish two frontiers with respect to the expected uncertainty reduction. Second, *loop-closing and re-localization actions*, which are key to the uncertainty reduction about the robot’s pose.

In this work, we aim at combining these types of actions into a single one. We seek to predict what the so far unseen environment beyond a frontier *may* look like based on background knowledge of previously seen environments and select the frontier that potentially leads to a loop-closure. In this way, we maximize the expected uncertainty reduction in the belief of the robot about the world.

4.2. Utility Function for Exploration

Most exploration systems define a utility function to relate the expected gain in information with the cost of obtaining the information. As long as no constraints such as available energy or similar are considered, the distance that the robot has to travel to obtain its measurements is a standard choice. This yields a utility function of the form

$$U(a) = I(M, Z; Z^a) - \text{cost}(a), \quad (3)$$

so that the task of selecting the best action can be formulated as

$$a^* = \underset{a}{\operatorname{argmax}} I(M, Z; Z^a) - \text{cost}(a). \quad (4)$$

Throughout this work, we define the cost function $\text{cost}(a)$ as the path length corresponding to action a , i.e. the length of the trajectory from the current location of the robot to the designated target location.

As mentioned in the previous section, estimating the expected information gain is challenging and computationally demanding and thus we use the following approximation. We assume that actions can reduce the robot’s uncertainty about the map by exploring unseen areas and/or can reduce its uncertainty about the trajectory by closing a loop:

$$a^* = \underset{a}{\operatorname{argmax}} I_{\text{map}}(a) + I_{\text{traj}}(a) - \text{cost}(a). \quad (5)$$

As we do not know how large the unknown area and thus the number of unknown grid cells behind a frontier is, we may argue that all frontiers yield the same expected information gain with respect to the map uncertainty. Thus, we can simplify Eq. (5) as long as we consider only exploration actions to frontiers:

$$a^* = \underset{a}{\operatorname{argmax}} I_{\text{traj}}(a) - \text{cost}(a). \quad (6)$$

The expected information gain about the trajectory $I_{\text{traj}}(a)$ is mainly influenced by loop closures. The more likely a loop closure can be obtained when executing an exploration action a , the higher its expected gain. Thus, the remainder of this section addresses the problem of predicting possible loop closures.

4.3. Predictive Exploration

The key contribution here is to model the predictive belief describing what the environment may look like in the unexplored areas. To compute this belief, the robot exploits environment structures it has seen in the past—either in the environment explored so far or even from previous missions. Our exploration system uses this predictive belief to evaluate the frontiers as possible target locations for the exploration. This allows us to select the frontiers that are likely to lead to a loop-closure and thus to an active reduction of the uncertainty in the robot’s belief. As we show during the experimental evaluation, this approach outperforms the traditional frontier-based exploration system.

4.4. Querying for Similar Environment Structures

The key idea of this approach is to look for similarities between the known areas around a frontier and portions of previously mapped environments. Under the assumption that environments are not random but expose certain structures and that these structures tend to appear more than once, we can use the already mapped areas in order to predict what the environment beyond the frontier *may* look like.

The first step is to look for portions of the already mapped environments that are similar to the area around the frontier for which the prediction should be performed. To do this, we incrementally build a database storing all local grid maps that the robot experienced. To perform a similarity query, we compare our local maps with the maps stored in the database. To avoid a large number of expensive map-to-map comparisons to search for similar submaps, we rely on a bag-of-words inspired approach, a technique that is frequently used in computer vision to search for image similarities. More concretely, we apply FabMAP2 by Cummins and Newman [33], an appearance-based approach we can use to efficiently query our database.



Figure 3: Illustration of the loop closures prediction. Left: So far explored map with the frontier under consideration (blue circle). Middle: One map from the predictive belief (in red) superimposed on the map explored so far. Right: Voronoi diagram used for the path search.

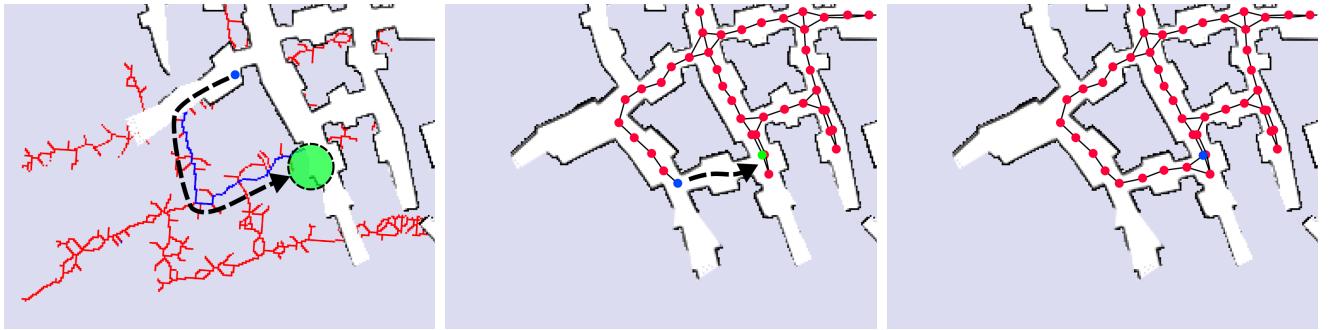


Figure 4: Illustration of the active loop closing. Left: prediction of the possible path with the loop closure shown in blue. Middle: the robot explores the path along the predicted loop closure and perceives the actual structure of the scene. The graph in the already explored environment shows the pose graph of the SLAM system. Right: successful loop closure Please note that the predicted environment is actually not identical with the real environment but reveals a similar structure. This similarity resulted in the shown loop closure.

Although FabMAP2 was originally designed to match camera images, it turns out that we can also use it to effectively search for local grid maps in a large database of maps. As FabMAP2 also provides a likelihood $l(m)$ for each match m , we can obtain a belief about possible environment structures. Fig. 2 shows an illustration of this procedure. The image on the left is a query image and the other images are the top 4 matches reported by FabMAP2.

4.5. Loop Closures Prediction

As we are mainly interested in the possible paths through the unknown environment in order to find loop closures and not necessarily the exact geometry, we reduce the maps reported by FabMAP2 to extended Voronoi graphs [34] and do all further computations on these graphs.

FabMAP2 provides us with candidates of matching maps but no geometric alignment between the query map and the reported ones. Thus, we align each map reported by FabMAP2 with our query map. This can be done in a robust manner through a RANSAC-based alignment of the Voronoi graphs using its junction points. Fig. 3 shows an example of a Voronoi graph aligned with the map explored so far.

The next step, is to search for possible loop closures, for which we use the extended Voronoi graph. Starting from the frontier point, we traverse the Voronoi graph in a breadth-first manner. During the traversal, we check if the Voronoi graph

leads to a position that is close to any other frontier in the map built so far. If this is the case, we regard that as a possible loop closure. Such a situation is illustrated in the left image of Fig. 4. This process is executed for each frontier.

4.6. Estimating the Probability of Closing a Loop

Each map reported by FabMAP2 comes with a likelihood. Thus, we can approximate the probability of closing a loop when executing an exploration action as

$$S_f = \sum_{m \in \mathcal{M}(f)} l(m) \sum_{c \in \mathcal{C}(f, m)} l(c | m) \quad (7)$$

Here, $\mathcal{M}(f)$ is the set of matches returned by FabMAP2 when querying with the frontier f , and $l(m)$ the likelihood of a match m . The term $\mathcal{C}(f, m)$ refers to the set of possible loop closures computed according to the breadth-first traversal explained above and $l(c | m)$ is the likelihood that the loop closures can be reached. We assume that $l(c | m)$ is proportional to the inverse length of the path of the predicted loop closure. This means that short loop closures are more likely than long ones.

Assuming that every executed loop closure through unknown areas of the map yields the same expected uncertainty reduction, we can approximate the expected information gain I_{traj} of Eq. (6) with the score S_f according to Eq. (7). This is clearly a strong assumption but we argue that a high score indicates a high expected gain from exploring the frontier.

5. Robust Homing Using Map Consistency Checks

Under the assumption that we can ensure the consistency of the current map, homing is a comparably easy task. It basically consists of computing a collision-free path from the current location to the starting location and following the planned path with a standard navigation pipeline. Such a navigation system would, for example, localize the robot in the map built so far and plan the shortest path towards home using A* or a similar approach. If the map, however, is not consistent because the underlying SLAM system has failed, this approach is likely to lead to a deadlock situations from which the robot cannot escape easily.

To ensure a robust exploration of the environment, we address the problem of robust homing in a two-stage approach. First, while mapping the environment, a path is computed from the current location towards home assuming the map is consistent. Then, we perform the recently proposed map consistency estimation approach by Mazuran *et al.* [1] to evaluate if the map is consistent with a given confidence level. If the path towards home is consistent, and we finished exploring the environment, we simply execute this plan. If the path towards home is not consistent, we aim at reversing the trajectory of the robot taken so far by aligning the current observation with the observations obtained on the way *from the starting location* to the current one. This yields a robust strategy to bring a robot home to its starting location.

5.1. Map Consistency Test

Our map consistency estimation approach proposed previously in [1] builds upon a pose-graph representation, i.e., the location of the robot from which individual observations have been taken. We start with evaluating the consistency of pairs of range readings. The approach of Mazuran *et al.* describes the discrepancy between two range scans by computing how much the two scans occlude each others free space.

To estimate the occlusion of the free space, we compute for each scan the polygon of the robot's pose and all end points of the range scan. Such polygons define the free space covered by the scan taken from the robot's pose. The intuition is that both scans are consistent with each other if none of the end points of the first scan lies inside the polygon of the second one and vice versa. In [1], we define an inconsistency distance $d(p)$ for a point p , which lies inside the polygon of another scan, as the Euclidean distance of a point p to the closest point on the polygon boundary of the other scan. Intuitively speaking, for a consistent map, we assume that the inconsistency distances $d(p)$ are in line with the sensor noise of the proximity sensor. Substantially larger values for $d(p)$ may indicate that the scans are not properly aligned and the map may be inconsistent in local neighborhood of the scans.

More concretely, we can expect that, under the assumption of a correct alignment of two scans, on average 50% of the end points from the first scan have an inconsistency distance $d(p) > 0$ in the second scan and vice versa. This is due to the sensor noise in the range measurements. According to [1], we can formulate a statistical test for the sum of inconsistency

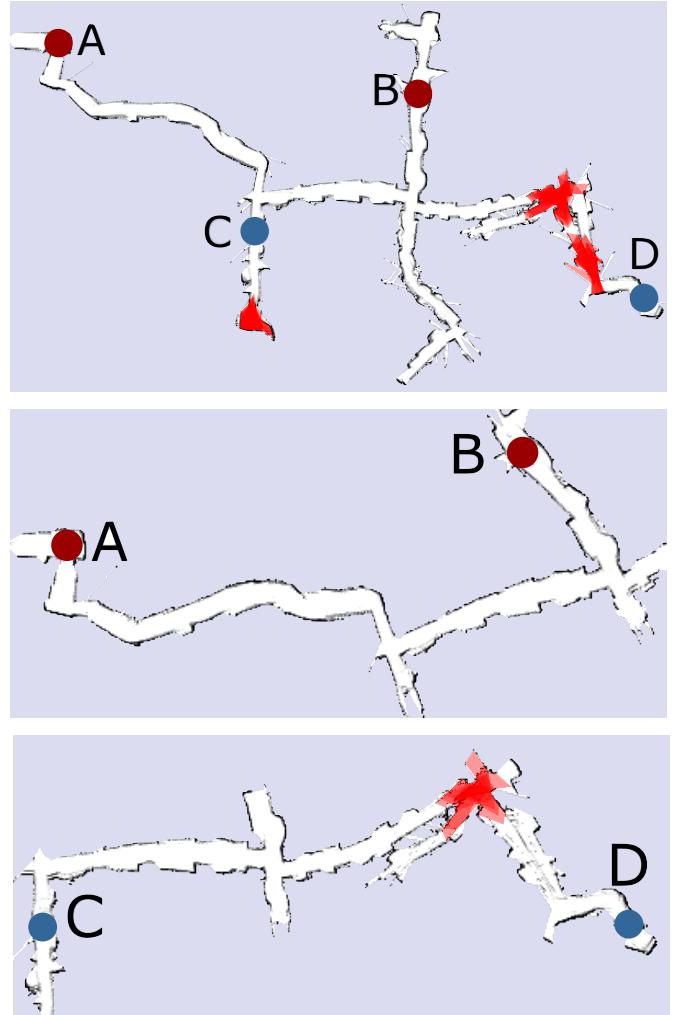


Figure 5: The top image shows the map built so far with the detected inconsistencies (inconsistent scans are shown in red). The middle one shows a submap that is built using only the scans recorded around the A* path from A to B computed in the full map. In this example, no inconsistencies are present and none are detected. The bottom image is done in the same way as the middle one, but the A* path is computed from C to D and here, the map inconsistencies are correctly detected.

distances $d(p)$. This test evaluates if pairs of scans are consistent given the sensor noise or reveal a larger error and thus an inconsistency.

To assess global map consistency, we could conduct this test for all pairs of scans and consider a map to be consistent if all tests are successful. The problem, however, is that a single statistical test will produce the wrong result with probability α . Thus, if we test a single scan, which overlaps with r other scans, this yields a type I error probability of $1 - (1 - \alpha)^r$ and thus renders the direct application of the pairwise approach unsuitable. The key trick is to model the outcome of the pairwise hypothesis test as a Bernoulli-distributed random variable with parameter α . As a result of that, the number of failed tests follows a binomial distribution with parameters α and r . Given that, we can compute the maximum number $\hat{\xi}$ of tests that are

allowed to fail at a confidence level $1 - \alpha'$ as

$$\hat{\xi} = \min_{0 \leq \xi \leq r} \left\{ \xi \mid \sum_{i=\xi+1}^r \binom{r}{i} \alpha^i (1-\alpha)^{r-i} \leq \alpha' \right\}. \quad (8)$$

This allows for computing a cascaded hypothesis test for all overlapping scans: We perform all pairwise hypothesis tests. If the number of failed tests is smaller than $\hat{\xi}$, the overall consistency test is positive otherwise negative. For more details, we refer the reader to [1].

5.2. Map Consistency Estimate for Finding the Way Home

Given the consistency test presented above, we can perform a mathematically sound statistical test to evaluate if a map is consistent or not. However, what the robot really needs to know is not the consistency of the full map. Instead, it is sufficient to know if it can safely move along a specific path through the environment to the starting location. Thus, we plan a path with A* assuming that the current map is consistent and extend our previous statistical consistency check to consider only the scans along that path. To achieve this, we select all recording locations that were closer than twice the maximum sensor range away from the trajectory planned with A*. Examples of such partial maps are depicted in Fig. 5. The top image shows an inconsistent 2D map of the Priscilla catacombs. Directly applying the approach described in [1] would label the whole map as inconsistent. In contrast to that, if the robot only takes into account the shortest route from A to B, he can still safely perform the navigation task, as shown in the middle image of the same figure. This is not the case if the robots wants to go from C to D as he will encounter an inconsistent part of the map on its way.

In terms of the persistent data structure that is used to store all the information, we use a generalization of a pose graph. Each node in the graph corresponds to a pose of the robot at time t . In addition to that, each node stores the original odometry pose X_t and the corresponding 3D point cloud c_t as well as the 2D scan. To efficiently represent this, the pose graph with the nodes X_t itself is kept in memory but the corresponding point clouds c_t are stored on disk and are loaded on demand.

5.3. Robust Homing by Rewinding the Trajectory

Once the consistency check has identified that the submap including the path is inconsistent, we need to perform the trajectory rewinding to bring the robot home safely. At this moment the robot does not have a consistent map, so it needs to rewind the trajectory without relying on a map. We assume that the environment remains static while the robot performs homing which is typically the case for the underground environments we explore. We can view the robot's forward trajectory as a series of 3D poses of the robot $\{X_0, \dots, X_n\}$. The task of rewinding the trajectory is to drive the robot from X_n to X_0 while correcting for the error in odometry. The correction of the odometry error is done by aligning the point clouds obtained while performing trajectory rewinding with the ones corresponding to poses from X_n to X_0 . Note that we subsample the trajectory in such way that each pose X_i is either 1 m

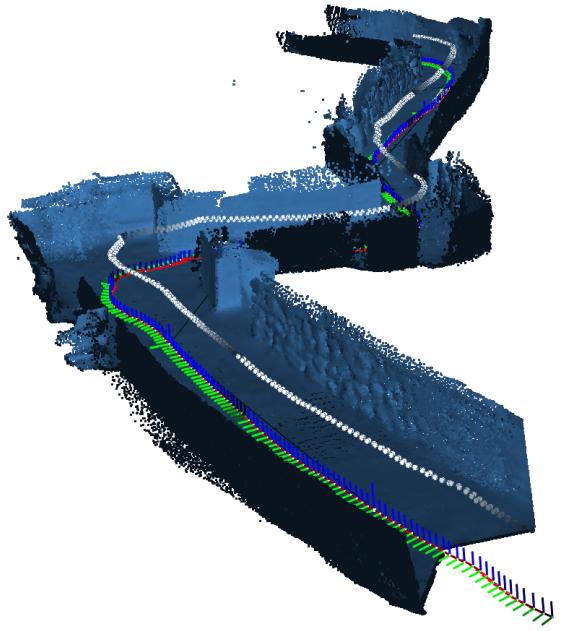


Figure 6: Partial view of the 3D model of the environment of the Priscilla catacombs built from two ASUS Xtion cameras.

away from the previous one or that there is a difference of at least 10° in yaw between these two poses.

Without loss of generality, let us consider that the robot has to carry out the action to move from X_i to X_j and to compensate for the error in odometry. To do that, the robot exploits the current point cloud c_{current} obtained after executing the movement from X_i to X_j . In an ideal world, the command should have brought the robot to the pose X_j . In reality, there is an error introduced by slippage, uneven ground etc. Thus, we align c_{current} with c_j . To achieve that, we use a recent robust variant of ICP called NICP [35] to find the discrepancy between the point cloud that the robot expects to perceive and what it actually perceives. The NICP method extends point-to-plane error metric proposed in Generalized ICP [36] by accounting not only for the metric distance between the points but also for the curvature of the underlying surface. The transformation between the point clouds provided by the matching algorithm can be viewed as the difference in the 3D poses at which the two point clouds c_{current} and c_j are obtained. The transformation reported by the NICP algorithm corresponds to T_Δ and thus leads to the relative position of c_{current} expressed in the local coordinate frame defined by X_j . Knowing the pose X_j and the pose of c_{current} relative to it through T_Δ enables us to compute the current position of the robot in the global odometry frame: $X_{\text{current}} = T_j T_\Delta$, where T_j is a transformation matrix that corresponds to the pose X_j in the world coordinate frame.

We use this new 3D pose X_{current} to generate a motion command to reach the next pose chosen from the recorded trajectory. As we have a wheeled platform that moves on the ground, we have no control over the height and attitude. Thus,

we generate 2D navigation commands for the robot. We continue the above-described process until the robot is within d_{\max} near its starting pose X_0 .

Note that our method relies on matching point clouds, typically seen from similar view points, i.e., no global search is needed. The vanilla ICP algorithm may converge to a local minimum while performing the optimization of the objective function. This can happen in very cluttered environments. Here, the objective function shows high variations with multiple local minima. On the contrary, this can also happen in places that are very feature-scarce as this may yield few distinct, narrow local minima. We found that using the NICP variant of ICP avoids such shortcomings in most practical situations, particularly when dealing with small view point changes, as is the case for our homing strategy. Refer to the work of Serafin and Grisetti [35] for a robustness analysis.

6. Experiments

The experiments are designed to illustrate (i) the advantages of our predictive exploration approach, if it is safe, and (ii) that the robot can rewind trajectories in case of failure of the mapping system.

For evaluating the next view point selection approach, we use a standard frontier-based exploration approach as a baseline and show that our exploration approach selects frontiers that lead to loop closures which in turn result in improved maps of the environment. The underlying mapping framework for all exploration experiments is a state-of-the-art graph-based SLAM system, which uses g2o [37] and FLIRT features to speed up the search for possible data associations [38], uses scan matching for incremental alignments, and applies single cluster graph partitioning to resolve ambiguities as proposed by Olson [39]. The exploration and homing systems have been implemented in C++ as ROS modules.

6.1. Map Comparisons

First, we compare the quality of the maps obtained with frontier-based exploration vs. our predictive exploration. The environments considered here are parts of the Roman catacomb Priscilla, a difficult to traverse and large-scale underground environment in Rome. The robot is equipped with tracks and thus its odometry is in general worse than the one of a wheeled robot and it sometimes reveals a (temporarily) bias to one side.

Fig. 7 illustrates the obtained results for two environments using exactly the same mapping system and identical parameters for the comparison. The map database consists of maps constructed from other catacomb sites representing a similar type of environment but not the same one. The images on the left are the “ground truth” maps obtained from manual surveys. The images in the second column correspond to the results of the frontier-based exploration, while the images on the right show our approach. As can be seen already visually, our approach yielded a consistent model of the environment, while the frontier-based approach failed. Using the frontier-based approach the robot was unable to continue its exploration task due

to an inconsistent map that prevented the computation of further exploration actions. We performed similar experiments in different nested tunnel environments and obtained comparable results.

The exploration task strongly benefits from achieving loop closures as early as possible, avoiding a high uncertainty in the pose-graph. Our approach improves the amount of loop closures whenever the current environment resembles previously seen maps, either in previous or current explorations runs. Thus, a new environment with recurrent structures also benefits from this approach. In case there is no similarity between the current environment and the maps stored in the database, no map should be retrieved and thus the system falls back to frontier-based exploration.

The execution time of our approach depends on the number of unexplored frontiers, as well as on the map size and resolution. On a standard computer and a map size of 150 m by 100 m with a grid resolution of 5 cm, next view point selection time ranged from 131 ms up to 4.8 s in the most complex situation. In practice, time consumed by the robot reaching the next view point usually dominates the time consumed by the selection task.

6.2. Exploration Path Length

The advantages of the prediction-based approach come at a cost—the cost of traversing exploration paths that are longer than the ones generated by the frontier-based approach. This experiment is designed to evaluate the increase in path length. As we are not able to obtain consistent maps for the frontier-based approach under a realistic noise model for the task under consideration, we executed this evaluation under zero noise in the simulator. Using a zero noise odometry, also the frontier-based system is able to build consistent maps. In this setting there is no advantage in using our predictive approach as the pose uncertainty is zero and no uncertainty reduction is gained from closing loops. We compared the distances traveled for the frontier-based and our approach. The distances traveled are summarized in Fig. 8. In the worst case scenario, the path generated by our approach was 1.85 times longer than the one of the frontier-based approach. The minimum increase was a factor of 1.5. Generating on average a 1.7 times longer trajectory is clearly an overhead—for actively closing loops and in this way reducing uncertainty, however, this price must be paid.

6.3. Statistical Map Consistency Check and Robust Homing

After the robot finishes exploring the environment, it needs to find its way home. The evaluation of our framework is designed to illustrate the performance of the statistical map consistency check in conjunction with an approach to safely and robustly rewind the trajectory to return the robot to the starting position should the consistency check report the map as inconsistent.

First, Fig. 5 illustrates an example of the statistical map consistency check performed on range data from the Priscilla catacombs in Rome. The partial maps computed around the shortest path are usually substantially smaller than the map of

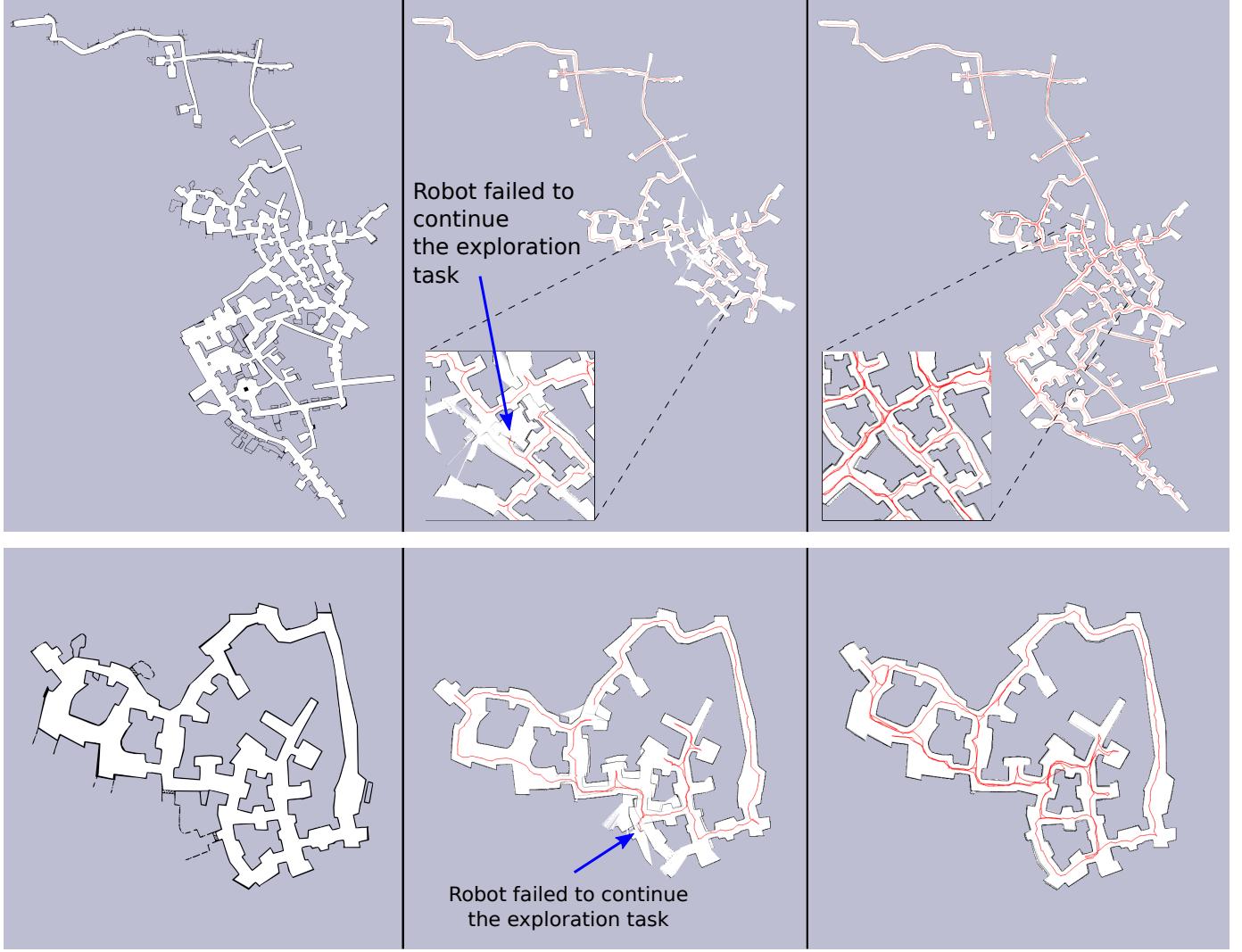


Figure 7: Two performance comparisons in constant odometry bias scenario. On the left, the original map. In the middle, the closer frontier approach. On the right, our prediction-based approach. Note that the nearest frontier approach produces a map that is non consistent with the original one, so that the robot can not continue the exploration task. The map produced by the prediction-based approach is instead consistent with the original one.

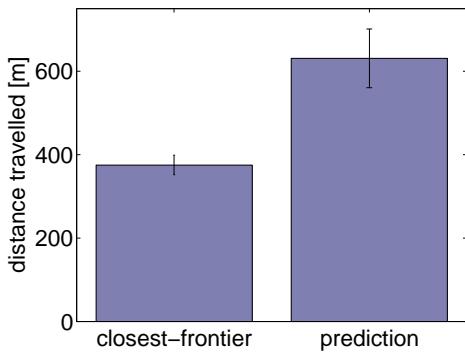


Figure 8: Mean and standard deviation of the distances traveled in the frontier-based approach and in the proposed approach.

the whole environment, especially if the environment has multiple alternative branches and forms a complicated network of corridors or rooms as we experience it often in catacombs or underground mines. Testing smaller maps results in speed-up of the statistical consistency evaluation procedure. The timings for the maps presented in Fig. 5 are as follows: full map shown on top—2,930 ms; middle—140 ms; bottom—170 ms. The computational time depends on the number of scans to analyze and the gain in speed grows with the difference between the sizes of the full and the reduced maps and the overlapping scans. We performed the map consistency test on five different datasets recorded in the Priscilla catacomb and the consistency check always generated correct results. In sum, testing a map along the planned path for consistency takes less than 200 ms and thus can be executed on the fly on the robot. Additionally, most of the computations could be cached when dealing with huge maps (although this was not done here). In this case, the test would only require a recomputation if the SLAM back-end

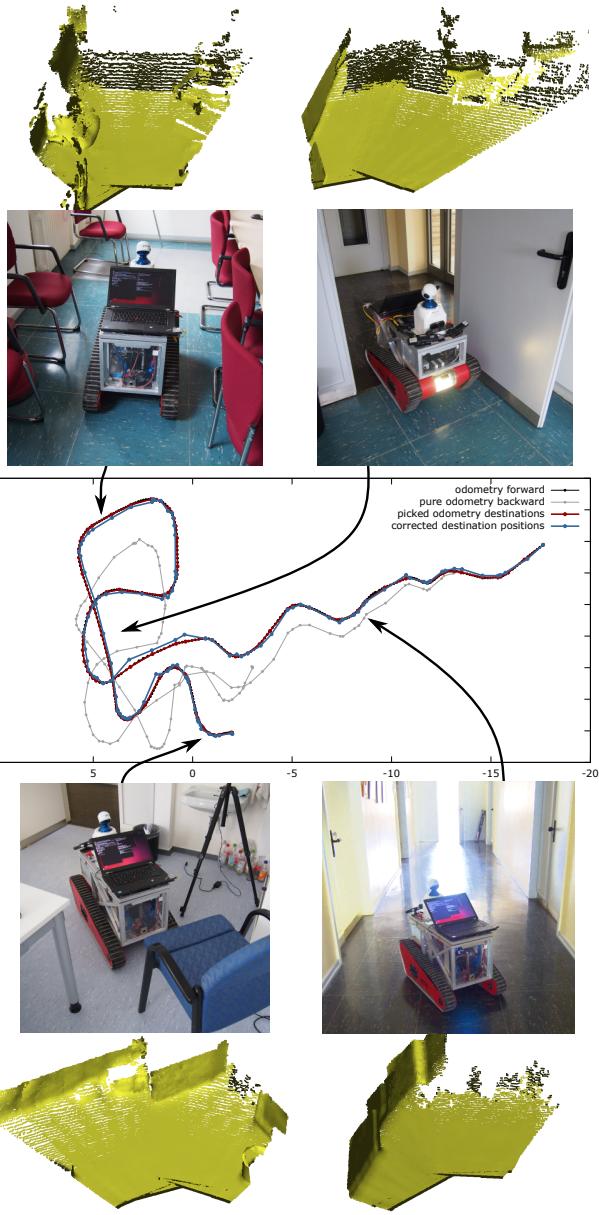


Figure 9: Illustration of rewinding the trajectory through the office environment. The robot is steered from the bottom “tail” of the depicted trajectory to the upper-right one. Black line denotes the odometry poses saved while the robot is steered, gray denotes the odometry on the way back, red shows the temporary destination poses picked from the odometry and blue shows the same poses after the ICP correction. The pictures show several example locations visited by the robot. These feature tight doors to rooms as well as feature-scarce corridors.

changes the configuration of the pose graph substantially.

Second, if the proposed statistical consistency check evaluates the map as inconsistent we need a robust way to return the robot home to the starting location. We evaluate the ability of our approach to rewind the trajectory by carrying out 20 experiments in our lab environment. One of these experiments is illustrated in Fig. 9. We steered the robot on a rather complicated trajectory through an obstacle parcour containing narrow

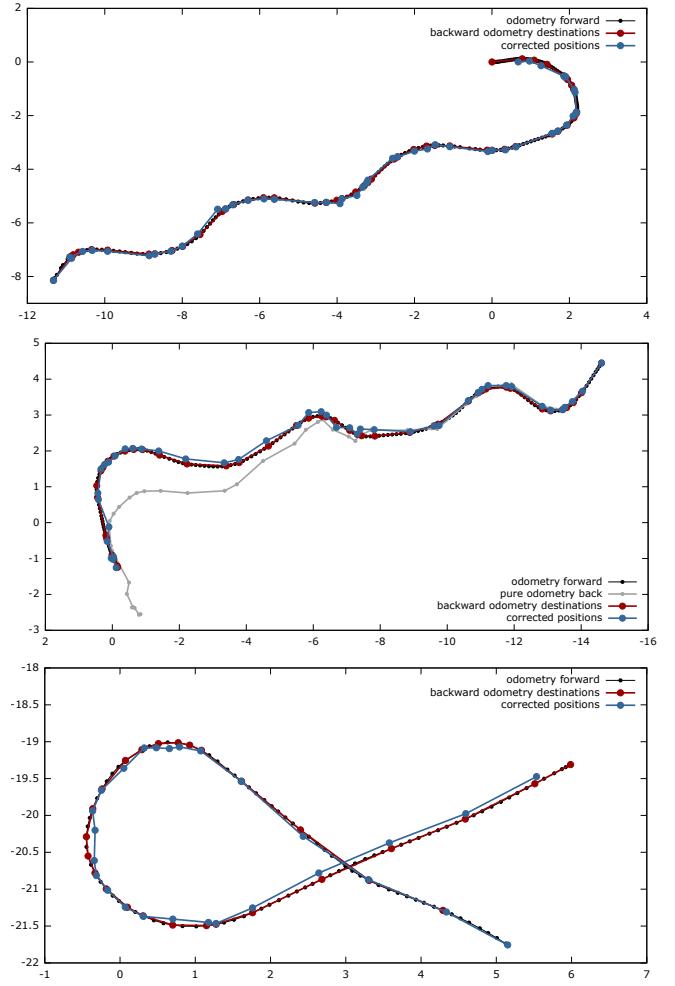


Figure 10: Three experiments performed in different settings. The meaning of the lines is the same as in Fig. 9 with the difference that the top and the bottom graphs do not show the pure odometry measurement on the return path. The average deviation from the original trajectory is between 4 cm (top dataset) and 6 cm (bottom dataset).

passages as well as areas with lots of flat wall, which represents a challenge for the matcher. The robot activated the “rewind the trajectory” behavior after we (manually) broke the SLAM system so that it followed the way in reverse order using the NICP-based pose correction.

In Fig. 9, the original odometry measurements from the *forward path* are drawn in black (hardly visible as the red trajectory perfectly overlays it). The red line illustrates the subsampled trajectory that the robot has selected as its sequence $\{X_0, \dots, X_n\}$ for rewinding the trajectory. Both trajectories overlay because the robot does not use any global map and relies solely on the poses he recorded in the odometry frame (to navigate back).

The gray line depicts the pure odometry measurements recorded while performing rewinding while the blue line shows the poses of the robot after the alignment by NICP. As can be seen, the robot accurately follows the previous trajectory with our approach as the blue and the red trajectories are similar. We compute the deviation from the original odometry path by

searching the closest pose of the robot from the forward traversal (black) for each pose from the backwards traversal (blue) and averaging over all such distances. In this experiment, the average deviation of the rewinding trajectory is approximately 5 cm. From the gray trajectory, we can furthermore see that the odometry error must be taken into account—otherwise, the robot would deviate substantially from the reference path (and would collide with walls and obstacles).

We executed similar experiments in 20 different settings with trajectory lengths ranging from 10 m to nearly 100 m and the robot was always able to robustly drive back to the start location. The trajectory in Fig. 9 is approximately 52 m long while three smaller examples are illustrated in Fig. 10. Overall, this evaluation suggest that our robot is able to rewind different trajectories through the environment, robustly handling corridor-like environments with multiple narrow passages such as the doorways. Note that the robot cannot observe doorways before it fully passed through them. Only by following the reference trajectory precisely, the robot can return.

7. Conclusion

The ability to robustly operate without user intervention is an important capability for exploration robots in real-world settings. In this paper, we proposed a novel approach for autonomous exploration of unknown environments with robust homing. The key contributions of this work are two-fold. First, we presented a technique to predict possible environment structures in the unseen parts of the robot’s surroundings based on previously explored environments. We exploit this belief to predict possible loop closures that the robot may experience when exploring an unknown part of the scene. This allows the robot to actively reduce the uncertainty in its belief through its exploration actions. Secondly, we presented a homing system that addresses the problem of returning a robot operating in an unknown environment to its starting position even if the underlying SLAM system fails. We combined a statistical map consistency test with an NICP-based approach to precisely rewind a previously taken trajectory.

We implemented our approach and executed it both, in simulation and on a real autonomous robot. Our experiments illustrate that our technique allows for an effective exploration of difficult to map environments. By actively closing loops, we are able to obtain consistent maps of the environment. In contrast to that, a traditional frontier-based exploration approach is not able to successfully explore the scene if the SLAM system fails. In the case of a mapping failure leading to an inconsistent map, the proposed robust homing system can accurately rewind trajectories guiding the robot through narrow passages such as doorways, even when the robot could not see these narrow spaces while navigating through them.

8. Acknowledgments

This work has partly been supported by the European Commission under the grant number FP7-ICT-600890-ROVINA, and

by the Agencia Canaria de Investigación, Innovación y Sociedad de la Información (ACIISI), co-funded by the European Social Fund (ESF).

References

- [1] M. Mazuran, G. Tipaldi, L. Spinello, W. Burgard, C. Stachniss, A statistical measure for map consistency in slam, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), Hong Kong, China, 2014.
- [2] S. Koenig, C. Tovey, Improved analysis of greedy mapping, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2003.
- [3] B. Yamauchi, Frontier-based exploration using multiple robots, in: Proc. of the Int. Conf. on Autonomous Agents, 1998, pp. 47–53.
- [4] P. Whaite, F. P. Ferrie, Autonomous exploration: Driven by uncertainty, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (3) (1997) 193–205.
- [5] C. Stachniss, W. Burgard, Mapping and exploration with mobile robots using coverage maps, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2003, pp. 476–481.
- [6] R. Rocha, J. Dias, A. Carvalho, Exploring information theory for vision-based volumetric mapping, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Edmonton, Canada, 2005, pp. 2409–2414.
- [7] T. Duckett, S. Marsland, J. Shapiro, Fast, on-line learning of globally consistent maps, Journal of Autonomous Robots 12 (3) (2002) 287–300.
- [8] J. Ko, B. Stewart, D. Fox, K. Konolige, B. Limketkai, A practical, decision-theoretic approach to multi-robot mapping and exploration, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2003, pp. 3232–3238.
- [9] S. Shen, N. Michael, V. Kumar, 3d indoor exploration with a computationally constrained mav, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2012.
- [10] R. Shade, P. Newman, Choosing where to go: Complete 3d exploration with stereo, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2011.
- [11] C. Stachniss, G. Grisetti, W. Burgard, Information gain-based exploration using rao-blackwellized particle filters, in: Proc. of Robotics: Science and Systems (RSS), Cambridge, MA, USA, 2005, pp. 65–72.
- [12] A. Makarenko, S. Williams, F. Bourgoult, F. Durrant-Whyte, An experiment in integrated exploration, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2002.
- [13] F. Bourgoult, A. Makarenko, S. Williams, B. Grocholsky, F. Durrant-Whyte, Information based adaptive robotic exploration, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Lausanne, Switzerland, 2002.
- [14] R. Sim, G. Dudek, N. Roy, Online control policy optimization for minimizing map uncertainty during exploration, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2004.
- [15] H. Kretzschmar, C. Stachniss, Information-theoretic pose graph compression for laser-based SLAM, Int. Journal of Robotics Research 31 (2012) 1219–1230.
- [16] A. Krause, C. Guestrin, Near-optimal nonmyopic value of information in graphical models, in: Proc. of Uncertainty in Artificial Intelligence (UAI), 2005.
- [17] G. Hitz, A. Gotovos, F. Pomerleau, M.-E. Garneau, C. Pradalier, A. Krause, R. Siegwart, Fully autonomous focused exploration for robotic environmental monitoring, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2014.
- [18] S. Sadat, K. Chutskoff, D. Jungic, J. Wawerla, R. Vaughan, Feature-rich path planning for robust navigation of mavs with mono-slam, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), Hong Kong, China, 2014.
- [19] C. Mostegel, A. Wendel, H. Bischof, Active monocular localization: Towards autonomous monocular exploration for multirotor mavs, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), Hong Kong, China, 2014.
- [20] D. Fox, J. Ko, K. Konolige, B. Stewart, A hierarchical bayesian approach to the revisiting problem in mobile robot map building, in: Proc. of the Int. Symposium of Robotics Research (ISRR), 2003.

- [21] H. Chang, C. Lee, Y. Lu, Y. Hu, P-slam: Simultaneous localization and mapping with environmental-structure prediction, *IEEE Transactions on Robotics* 23 (2) (2007) 281–293.
- [22] C. Stachniss, O. Martínez-Mozos, A. Rottmann, W. Burgard, Semantic labeling of places, in: Proc. of the Int. Symposium of Robotics Research (ISRR), San Francisco, CA, USA, 2005.
- [23] K. Wurm, C. Stachniss, W. Burgard, Coordinated multi-robot exploration using a segmentation of the environment, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2008.
- [24] C. Stachniss, O. Martinez Mozos, W. Burgard, Efficient exploration of unknown indoor environments using a team of mobile robots, *Annals of Mathematics and Artificial Intelligence* 52 (2009) 205ff.
- [25] D. Holz, N. Basilico, F. Amigoni, S. Behnke, A comparative evaluation of exploration strategies and heuristics to improve them, in: Proc. of the European Conference on Mobile Robots (ECMR), Örebro, Sweden, 2011, pp. 25–30.
- [26] C. Sprunk, G. Tipaldi, A. Cherubini, W. Burgard, Lidar-based teach-and-repeat of mobile robot trajectories, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Tokyo, Japan, 2013.
- [27] P. Furgale, P. Krüsi, F. Pomerleau, U. Schwesinger, F. Colas, R. Siegwart, There and back again—dealing with highly-dynamic scenes and long-term change during topological/metric route following, in: ICRA14 Workshop on Modelling, Estimation, Perception, and Control of All Terrain Mobile Robots, 2014.
- [28] P. Furgale, T. Barfoot, Visual teach and repeat for long-range rover autonomy, *Journal on Field Robotics* 27 (5) (2010) 534–560.
- [29] M. Nitsche, T. Pire, T. Krajnik, M. Kulich, M. Mejail, Monte carlo localization for teach-and-repeat feature-based navigation, in: M. Misra, A. Leonardis, M. Witkowski, C. Melhuish (Eds.), *Advances in Autonomous Robotics Systems*, Vol. 8717 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 13–24.
- [30] E. Battesti, S. Bazeille, D. Filliat, Qualitative localization using vision and odometry for path following in topo-metric maps, in: Proc. of the European Conference on Mobile Robots (ECMR), 2011, pp. 303–308.
- [31] D. P. Ström, F. Nenci, C. Stachniss, Predictive exploration considering previously mapped environments, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2015.
- [32] I. Bogoslavskyi, O. Vysotska, J. Serafin, G. Grisetti, C. Stachniss, Efficient traversability analysis for mobile robots using the kinect sensor, in: Proc. of the European Conference on Mobile Robots (ECMR), 2013.
- [33] M. Cummins, P. Newman, Highly scalable appearance-only slam fab-map 2.0, in: Proc. of Robotics: Science and Systems (RSS), 2009.
- [34] P. Beeson, N. Jong, B. Kuipers, Towards autonomous topological place detection using the extended voronoi graph, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2005.
- [35] J. Serafin, G. Grisetti, NICP: Dense normal based point cloud registration and mapping, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2015.
- [36] A. Segal, D. Haehnel, S. Thrun, Generalized-icp., in: Proc. of Robotics: Science and Systems (RSS), Vol. 2, 2009.
- [37] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, g2o: A general framework for graph optimization, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2011, pp. 3607–3613.
- [38] G. Tipaldi, K. Arras, Flirt—interest regions for 2d range data, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2010, pp. 3616–3622.
- [39] E. Olson, Recognizing places using spectrally clustered local matches, *Robotics and Autonomous Systems*.