

Planning Reliable Paths with Pose SLAM

Rafael Valencia, Martí Morta,
Juan Andrade-Cetto, and Josep M. Porta

Abstract—The maps built by standard feature-based SLAM methods cannot be directly used to compute paths for navigation, unless enriched with obstacle or traversability information with the consequent increase in complexity. Here, we propose a method that directly uses the Pose SLAM graph of constraints to determine the path between two robot configurations with lowest accumulated pose uncertainty, i.e., the most reliable path to the goal. The method shows improved navigation results when compared to standard path planning strategies, both over datasets and real world experiments.

Index Terms—SLAM, Path Planning, Autonomous Navigation

I. INTRODUCTION

Aside from applications such as the reconstruction of archaeological sites [1] or the inspection of dangerous areas [2], the final objective for an autonomous robot is not to build a map of the environment, but to use this map for navigation, i.e., to reach distant locations in the environment efficiently and safely. In recent years, we have witnessed an amazing advance in the field of simultaneous localization and map building (SLAM), and state of the art approaches can now build maps over several kilometers [3]. For efficiency reasons, most SLAM algorithms represent the environment using a sparse set of features. Unfortunately, this representation cannot be directly used for collision-free path planning since it does not provide much information about which routes in the map have been previously traversed safely, or about the nature of the obstacles it represents. Those sparse models could be somehow enriched with obstacle or traversability related information [4]–[6], but at the expense of an increase in complexity.

The problem of finding paths to reach distant locations is addressed in the motion planning literature, and the most successful methods are based on randomized sampling [7, 8], in which collision-free configurations are stochastically drawn and where, if possible, neighbor samples are connected forming a roadmap. This roadmap is later used to find a path between any two given configurations. Some approaches have addressed the problem of optimizing the quality of this path, mainly focusing on reducing the path length [9, 10]. In any case, the research in motion planning typically assumes deterministic setups where a perfect model of the environment is available and where the configuration of the robot is perfectly known too.

Some extensions have been introduced recently to deal with uncertainties in the model of the environment [11], in the robot configuration [12], in the effect of robot actions [13], or in the effect of actions and measurements [14]. The extension that best matches the stochastic nature of SLAM is the Belief Roadmap (BRM) [15, 16]. In this approach, the edges defining the roadmap include information about the uncertainty change when traversing

such an edge. However, the main drawback of the BRM is that it still assumes a known model of the environment, which is in general not available in real applications. In this paper, we aim to overcome the limitation of BRMs noting that the map generated by Pose SLAM [17], or any other delayed-state SLAM method [18]–[20], can be directly used as a belief roadmap.

Pose SLAM is the variant of SLAM where only the robot path is estimated and where landmarks are only used to produce relative constraints between robot poses. In a semi-autonomous scenario where a human initially drives the robot through a set of interest way points, the outcome of Pose SLAM is a graph of obstacle-free paths in the area where the robot has been operated. Using this graph for navigation allows to have an infrastructure-free automated guided vehicle, as those widely used for material distribution in factories, or for drug delivery in hospitals [21, 22]. An added advantage is that Pose SLAM is agnostic with respect to the sensor modalities used, which facilitates its application in different environments and robots. Moreover, the paths stored in the map satisfy constraints not easy to model in the robot controller, such as the existence of restricted regions, or the right of way along paths. Deviations from these paths might result in an inconvenience for other operations carried out in the factory or hospital. Thus, a robot that can adequately choose the correct path from a set of previously traversed ones, or their combination, is desirable for such applications. However, in those scenarios, the use of a robot is only practical if it is able to autonomously navigate without becoming lost.

In this paper, we show that, using the information stored in the Pose SLAM graph, we can plan in the belief space to obtain paths with the least probability of becoming lost. The key idea behind our method is that, in Pose SLAM, highly informative areas of the environment result in poses in the graph with low uncertainty. In these areas, sensor matching is more reliable, and there is less risk of deviating from the path during execution. We first introduced this idea in [23], and here, we a) simplify the computation of the path cost showing that the uncertainty of the localization estimate can be evaluated without explicitly maintaining a localization filter, b) propose alternatives to efficiently deal with large-scale problems by approximating marginal covariances, and c) thoroughly evaluate the approach in simulated and real datasets and with a real robot in outdoor experiments.

From the point of view of SLAM, this paper constitutes a step forward to actually use the output of the mapping process for path planning. From the point of view of motion planning, this paper contributes with a method to generate belief roadmaps without resorting to stochastic sampling on a pre-defined environment model. Note however that the proposed method limits the planning only to those areas already covered by the Pose SLAM graph. The exploration of novel paths is out of the scope of this work. We address the issue in [34].

After a review of the state of the art in Section II, the rest of the paper details the proposed extensions to Pose SLAM to determine reliable paths. Section III summarizes Pose SLAM and reinterpret its map as a set of samples in belief space, and Section IV describes how to plan using a roadmap defined on these samples. In Section V, this new planning approach is tested with datasets and real world experiments and, finally, Section VI gives some concluding remarks.

II. RELATED WORK

Initial work in SLAM represented the environment using a sparse set of features. However, this representation needs to be enriched with obstacles or traversability related information before it can be used for collision-free path planning. For instance, in [4], the map of the environment is enriched with virtual free-space markers connected defining a graph of traversable regions. A robot would navigate first

This work has been partially supported by a PhD scholarship to R. Valencia from the Mexican Council of Science and Technology and by the Spanish Ministry of Economy and Competitiveness under projects DPI-2008-06022, DPI-2011-27510, DPI-2010-18449, MIPRCV Consolider-Ingenio 2010, and by the EU Project ARCAS FP7-287617.

The authors are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens Artigas 4-6, Barcelona 08028, Spain (email: rvalencia@iri.upc.edu; mmorta@iri.upc.edu; cetto@iri.upc.edu; porta@iri.upc.edu)

The paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors. The material includes the Matlab code of the introduced planner and a video of the real robot navigation experiment presented in the paper. Contact rvalencia@iri.upc.edu for further questions about this material.

to the nearest free-space marker and then follow the free-space graph to the goal. Alternatively, the hybrid metric maps (HYMMs) in [5] split the environment in local triangular regions (LTR) whose corners are features in the map. Each LTR has local paths that traverse it and an associated cost for each of these local paths so that a planner can find the lowest cost path from any point to the goal through a sequence of LTRs. HYMMs were improved to include a scalar evaluation of the potential information that the robot can obtain from the environment at each LTR [6]. This is relevant information for path planning, but comes at the expense of a significant increase in complexity and in memory use.

Instead of enriching feature-based maps, other approaches build grid maps out of volumetric representations of 3D point clouds [24, 25]. These techniques, typically use the 3D map to extract a 2D map of traversable regions from which a graph-like roadmap is derived. Such graph is then used for path planning relying on standard graph search algorithms. Those approaches, however, also come at the expense of increased complexity and high memory cost. Moreover, traversability is typically computed over the mean estimate of the map, disregarding map uncertainty.

An alternative is to compute a lighter representation of the environment such as a topological map [26]–[28] and to use it for path planning [29]. However, since topological maps are not accurate enough to localize the robot in all cases, they are sometimes combined with local grid maps [30, 32]. In these approaches, the topological map is used as a roadmap to devise a path to the goal using graph search techniques, and the local grid map associated with each node in the topological map is used to compute a path from one node to the next, considering local obstacle avoidance and path smoothing. Although the use of hybrid metric-topological maps improves the efficiency of path planning when compared to the use of a global grid map, these approaches still require a considerable effort to maintain the coherence between both representations, especially at loop closure. As with the traversability maps, the computed routes on topological maps also ignore the sources of uncertainty included in the map.

In this paper we observe that the maps computed with Pose SLAM can be directly used as belief roadmaps and, thus, used for planning reliable paths without further processing the map nor enriching it with additional information. Moreover, since we rely on Pose SLAM which marginalizes out the sensor readings, the approach can be used with any type of sensors. In [3] it is suggested to use the graph of poses built with bundle adjustment for path planning, but the uncertainty information in the map is not exploited in the computation of the optimal path. In contrast, we use the maps computed with Pose SLAM to plan in the belief space obtaining paths to remote locations that take into account the uncertainty balance along them.

The approach is devised to autonomously guide the robot in scenarios where the robot had already built a map. This mapping session need not be exhaustive as long as it traverses all areas the robot is intended to visit in normal operations. Note however that the technique could be extended to plan routes in a partially built scenario during autonomous exploration [34], but this extension is out of the scope of this paper.

III. ENVIRONMENT SAMPLING WITH POSE SLAM

Pose SLAM produces a directed graph, in which the nodes are poses or waypoints, and the edges are established from odometry or sensor registration. Assuming Gaussian distributions, a probabilistic estimate of the pose nodes, $\mathbf{x} = \{x_1, \dots, x_k\}$, is maintained using an information filter with a canonical parametrization $p(\mathbf{x}) = \mathcal{N}^{-1}(\boldsymbol{\eta}, \boldsymbol{\Lambda})$, with information vector $\boldsymbol{\eta}$, and information matrix $\boldsymbol{\Lambda}$. This parametrization, compared to the traditional Kalman

form, with mean $\boldsymbol{\mu} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$ and covariance $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1}$, has the advantage of being exactly sparse [17].

In Pose SLAM, state transitions result from the composition of a motion command u_k to the previous pose,

$$x_k = f(x_{k-1}, u_k) = x_{k-1} \oplus u_k, \quad (1)$$

where \oplus is the operator used to add the relative displacement u_k to pose x_{k-1} , as described in [35].

Augmenting the state in information form introduces shared information only between the new robot pose x_k and the previous one x_{k-1} , resulting in an information matrix with a tridiagonal block structure. If the state mean is available, this operation can be performed in constant time.

Registration of sensory data also introduces shared information, but now between non-consecutive poses. These relative constraints can be modeled as

$$z_{ki} = h(x_k, x_i) = \ominus x_k \oplus x_i, \quad (2)$$

that is, the tail-to-tail operation defined in [35], which computes the relative displacement from x_k to x_i in the frame of reference of x_k . When establishing such a link, the update operation only modifies the diagonal blocks i and k of the information matrix $\boldsymbol{\Lambda}$ and introduces new off-diagonal blocks at locations ik , and ki . This operation is also executed in constant time, assuming the state mean to be available. These links enforce graph connectivity, or loop closure in SLAM parlance, and revise the entire path state estimate, reducing the overall uncertainty. The result is that the marginal uncertainty at each node in the graph results from the fusion of the uncertainties for all possible paths from the origin of the map to that node.

From the point of view of planning, it seems reasonable to distribute poses uniformly in the space where the plan is to be defined. In classical motion planning algorithms, the plan is built in the configuration space, but when taking into account the uncertainty, the plan must be defined in the belief space.

During map building, the distance in belief space from a new pose to any pose already in the map can be measured by the amount of information encoded in the link that connects those poses. Formally, the information gain of a link, i.e., the difference in entropy on the entire map before and after the link is established, can be evaluated as [17]

$$\mathcal{I}_{ki} = \frac{1}{2} \ln \frac{|\boldsymbol{\Lambda} + \Delta\boldsymbol{\Lambda}|}{|\boldsymbol{\Lambda}|} = \frac{1}{2} \ln \frac{|\mathbf{S}_{ki}|}{|\boldsymbol{\Sigma}_y|}, \quad (3)$$

where $\Delta\boldsymbol{\Lambda}$ is the information provided by the new link, $\boldsymbol{\Sigma}_y$ is the sensor registration error, \mathbf{S}_{ki} is the innovation covariance

$$\mathbf{S}_{ki} = \boldsymbol{\Sigma}_y + [\mathbf{H}_k \mathbf{H}_i] \begin{bmatrix} \boldsymbol{\Sigma}_{kk} & \boldsymbol{\Sigma}_{ki} \\ \boldsymbol{\Sigma}_{ki}^\top & \boldsymbol{\Sigma}_{ii} \end{bmatrix} [\mathbf{H}_k \mathbf{H}_i]^\top, \quad (4)$$

\mathbf{H}_k , \mathbf{H}_i are the Jacobians of h with respect to poses k and i evaluated at the state means μ_k and μ_i , $\boldsymbol{\Sigma}_{kk}$ and $\boldsymbol{\Sigma}_{ii}$ are the marginal covariances of poses k and i , respectively, and $\boldsymbol{\Sigma}_{ki}$ is the cross correlation between these two poses. If none of the links with neighboring poses is informative enough (above a threshold γ), there is no need to include such new pose in the map as it is too close to other poses in belief space. Note that the Pose SLAM system maintains the marginal covariances and cross correlations in amortized constant time during mapping [36], but when using a general delayed-state SLAM algorithm their computation requires to invert the information matrix, which is computationally expensive.

In Pose SLAM, nodes are added to the graph as a function of the information content in their connecting links, as measured by Eq. (3). Thus, we can say that the sampling methodology is aware of the uncertainty in both the motion and sensor models. Note, however, that the information content separating two nodes is only lower

bounded by γ , but there is no upper bound. Actually, information content between neighbor nodes varies depending on the quality of sensor registration, the speed of the robot, and the density of loop closures in that region. For this reason, different paths from a given start configuration to a goal node would entail different uncertainty profiles.

IV. PATH PLANNING WITH POSE SLAM

We are in the quest for a path $p = r_{1:T}$ that would drive the robot from its current configuration $s = r_1$ to a goal configuration $g = r_T$, that, for the applications considered in this paper, is always included in the Pose SLAM graph. In this quest, we assume that the robot is equipped with a local planner able to drive the robot to nearby configurations. Moreover, we assume maximum likelihood actions and measurements, as it is usual when planning in belief space [37]. With this, when tracking a path on the graph, the uncertainty estimate would evolve according to the motion uncertainty and sensor registration, but the mean estimate will lie at the mean of a node in the graph, and the observation previously obtained at that position will be repeated. In consequence, the proposed planning approach only needs to consider poses in the graph, which are guaranteed to be collision free.

Given that candidate paths lie on top of this graph, after path execution the final robot uncertainty will be close to the original marginal at that node. Thus, a cost function that only evaluates the belief state at the goal is unsuitable and we are interested instead in determining reliable paths, i.e., paths where the robot has low probability of becoming lost. Herein, we assume that the probability of becoming lost is directly related with the increments in the uncertainty in robot positioning since uncertainty decrements can only result in a better track of the path.

To identify the most reliable path we follow these steps. First, we increase the connectivity of the Pose SLAM graph so that paths combining different exploration sequences can be considered. Next, we propose a principled way to evaluate the changes in the uncertainty of the transitions between nodes, and use this uncertainty measure to define the cost of a path as its mechanical work in the uncertainty surface. Finally, this cost function is used to search the minimum uncertainty path to the goal.

A. Increasing Graph Connectivity

The graph used for path planning is initialized with the odometry edges of the Pose SLAM graph. However, we allow the local planner to attempt connections to other neighboring poses. In this way, the global planner can switch among different exploration sequences in the quest for an optimal path.

Extra edges are included for nodes with high probability of being close to each other and, thus, likely to be reachable using the local planner. To determine such poses, we estimate the relative displacement, d , from any robot pose x_k to any other pose x_i as a Gaussian with parameters

$$\mu_d = h(\mu_k, \mu_i), \quad \text{and} \quad (5)$$

$$\Sigma_d = [\mathbf{H}_i \ \mathbf{H}_k] \begin{bmatrix} \Sigma_{ii} & \Sigma_{ik} \\ \Sigma_{ik}^\top & \Sigma_{kk} \end{bmatrix} [\mathbf{H}_i \ \mathbf{H}_k]^\top. \quad (6)$$

Marginalizing the distribution of the displacement, d , along each one of its dimensions, t , we get a one-dimensional Gaussian distribution $\mathcal{N}(\mu_t, \sigma_t^2)$, which can be used to compute the probability of pose x_i being closer than v_t to pose x_k along such dimension

$$\begin{aligned} p_t &= \int_{-v_t}^{+v_t} \mathcal{N}(\mu_t, \sigma_t^2) \\ &= \frac{1}{2} \left[\operatorname{erf} \left(\frac{v_t - \mu_t}{\sigma_t \sqrt{2}} \right) - \operatorname{erf} \left(\frac{-v_t - \mu_t}{\sigma_t \sqrt{2}} \right) \right]. \end{aligned} \quad (7)$$

If for all dimensions, p_t is above a given threshold s , then configuration x_i is considered close enough to configuration x_k . Pose SLAM [17] computes the set of neighbors for each pose during mapping in logarithmic time, organizing the poses in a tree. For other delayed-state systems, though, the computation of the set of neighbors requires to compute the marginal covariances and the cross correlations between all pairs of poses, which is a costly operation.

Observe that the proposed approach remains agnostic about the properties of the local planner and, thus, neighboring poses are searched in a rectangular box around the current pose in configuration space. If we had information about the kinematic constraints of the robot, or about the distribution of obstacles around the robot, we could reduce the search for neighboring poses to smaller areas. For instance, for a car-like robot, we could focus the search for neighbors into triangular areas on the xy plane, in front of and behind the robot, since these are the regions including the kinematically-feasible neighboring poses. Also, for a robot that can only safely move forward due to the arrangement of its obstacle detection sensors, only neighboring poses in front of the robot need to be detected. In all cases, the size of the area where to look for neighbors is limited by the accuracy of the local planner, which typically relies in odometry readings.

Independently of the location and size of the area used to search for neighbors, an edge is added to the path planning graph only if the kinematic constraints enforced by the local planner allow reaching the nearby node. Despite this verification, the local path might be still unfeasible during path execution, mainly due to the presence of obstacles. In this case, the problematic edge can be removed from the graph and a re-planning process can be triggered.

B. Uncertainty Change of a Path Step

Every pair of poses in a planned path, say r_{k-1} and r_k , would match two poses in the Pose SLAM graph, say x_i and x_j . The command u_k that drives the robot from r_{k-1} to r_k and the associated motion noise Σ_u are provided by the local planner. Since, during the planning process the actual sensor readings are not available, the update of this estimate is made assuming that r_k and x_j will be coincident and, thus, $z_{kj} = r_k - x_j \sim \mathcal{N}(\mathbf{0}, \Sigma_{jj})$ where Σ_{jj} is the marginal covariance of the pose x_j .

To evaluate the uncertainty change introduced by the motion from r_{k-1} to r_k , we must look at the dispersion of the conditional distribution $p(r_k | r_{k-1}, u_k, z_{kj})$ that is obtained from the estimate of the joint state (r_{k-1}, r_k) . This estimate can be computed using, for instance, an EKF as in [23]. We show next how this uncertainty change can be more elegantly evaluated using an EIF parametrization instead, i.e., $p(r_{k-1}, r_k) = \mathcal{N}^{-1}(\bar{\eta}, \bar{\Lambda})$, but without actually computing such pose estimate.

In the prediction step the joint information matrix on these two states would be updated with [19]

$$\begin{bmatrix} \bar{\Lambda}_{k-1k-1} + \mathbf{F} \mathbf{Q}^{-1} \mathbf{F}^\top & -\mathbf{F}^\top \mathbf{Q}^{-1} \\ -\mathbf{Q}^{-1} \mathbf{F} & \mathbf{Q}^{-1} \end{bmatrix}, \quad (8)$$

where $\mathbf{Q} = \mathbf{W} \Sigma_u \mathbf{W}^\top$, \mathbf{F} and \mathbf{W} are the Jacobians of f with respect to r_{k-1} and u_k evaluated at the means, and where $\bar{\Lambda}_{k-1k-1}$ is obtained marginalizing from the previous $\bar{\Lambda}$. In the correction step the measurement z_{kj} is used to update the information matrix as

$$\bar{\Lambda} = \begin{bmatrix} \bar{\Lambda}_{k-1k-1} + \mathbf{F} \mathbf{Q}^{-1} \mathbf{F}^\top & -\mathbf{F}^\top \mathbf{Q}^{-1} \\ -\mathbf{Q}^{-1} \mathbf{F} & \mathbf{Q}^{-1} + \Sigma_{jj}^{-1} \end{bmatrix}. \quad (9)$$

With this, the uncertainty change of r_k given full confidence about r_{k-1} can be evaluated as

$$U_k = \frac{1}{|\bar{\Lambda}_{k|k-1}|}, \quad (10)$$

where $\bar{\mathbf{A}}_{k|k-1}$ is the information matrix for the conditional $p(r_k|r_{k-1}, u_k, z_{k,j})$ obtained from $\bar{\mathbf{A}}$.

Fortunately enough, conditioning in information form is dual to marginalization in covariance form, thus Eq. (10) simply evaluates to

$$U_k = \frac{1}{|\mathbf{Q}^{-1} + \Sigma_{jj}^{-1}|}, \quad (11)$$

which saves us from explicitly maintaining a localization filter as done in [23]. Furthermore, we can safely assume \mathbf{Q} to be non-degenerate and, thus, the determinant in Eq. (11) would never be null and U_k will be always well-defined. Note that we use a measure of uncertainty change derived from the determinant of the covariance matrix which is related to the entropy of $p(r_k|r_{k-1}, u_k, z_{k,j})$, and ultimately, to the uncertainty hyperellipsoid defined by this matrix. A trace-based uncertainty measure [15] can be used as well, without affecting the overall planning algorithm.

As said, this measure of uncertainty change is computed independently of the estimation of the robot pose at each step and, thus, this formulation saves us from actually implementing the EIF to track the path. This does not imply that the robot must have an identical belief to a pre-existing node in the optimized Pose SLAM graph. What it implies is that the computation of the information gain is independent of the current belief, as long as maximum likelihood actions and measurements are considered. This is especially relevant to marginalize the effects of the initial belief in the planning. Moreover, as long as the graph does not change, the uncertainty measure for all transitions can be precomputed from the Pose SLAM graph and re-used to plan different paths. This is similar to what is done in [15], factorizing the covariance update, but simplified thanks to the use of the information form.

C. Minimum Uncertainty along a Path

Next, we propose a cost function that considers the cumulative probability of becoming lost for a given path. Assuming that this probability for a given path step is directly related with the increments in uncertainty, determining optimal paths can be seen as searching for a path of minimal mechanical work [38] in an uncertainty change surface [39] over the space of robot poses, where the uncertainty change of a path step is computed using the criterion described in Section IV-B.

Given a discrete path $p = r_{1:T}$, we define its mechanical work in the uncertainty surface as the sum of positive increments of individual step costs

$$W(r_{1:T}) = \sum_{k=2}^T \Delta U_k^+, \quad (12)$$

with

$$\Delta U_k^+ = \begin{cases} \Delta U_k & \Delta U_k > 0, \\ 0 & \Delta U_k \leq 0, \end{cases} \quad (13)$$

and

$$\Delta U_k = U_k - U_{k-1} \quad (14)$$

and where, by convention, $U_1 = 0$, to include the uncertainty of the first step of the path in W . Note that, the initial uncertainty of the robot is not included in W since it would result in a constant offset for the cost of all alternative paths. Moreover, since the costs are non-negative, there is always an acyclic minimum cost path to each reachable node in the map.

This strategy prefers short paths with possibly steep uncertainty changes over longer paths with gentle oscillations of uncertainty, thus avoiding the accumulation of small chances of becoming lost over large trajectories.

Algorithm 1: Path planning with Pose SLAM.

PoseSLAMPathPlanning(M, g)
input : M : The graph computed by Pose SLAM.
 g : The goal pose.
output: p : The most reliable path to g .

```

1  $Q \leftarrow \text{POSES}(M)$ 
2 forall the  $n \in Q$  do
3    $W[n] \leftarrow \infty$ 
4    $V[n] \leftarrow 0$ 
5  $s \leftarrow \text{CURRENTPOSE}(M)$ 
6  $W[s] \leftarrow 0$ 
7  $U[s] \leftarrow 0$ 
8 repeat
9    $i \leftarrow \text{EXTRACTMIN}(Q, W)$ 
10  if  $i \neq g$  and  $W[i] \neq \infty$  then
11     $N \leftarrow \text{NEIGHBORS}(M, i)$ 
12    forall the  $j \in N$  do
13       $(u, \mathbf{Q}) \leftarrow \text{LOCALPLANNER}(\mathbf{x}_i, \mathbf{x}_j)$ 
14      if  $u \neq \emptyset$  then
15         $\Sigma_{jj} \leftarrow \text{MARGINALCOVARIANCE}(M, j)$ 
16         $U = 1/|\mathbf{Q}^{-1} + \Sigma_{jj}^{-1}|$ 
17         $\Delta U = U - U[i]$ 
18        if  $\Delta U > 0$  then
19           $W' = W[i] + \Delta U$ 
20        else
21           $W' = W[i]$ 
22        if  $W' < W[j]$  then
23           $W[j] \leftarrow W'$ 
24           $V[j] \leftarrow i$ 
25           $U[j] \leftarrow U$ 
26 until  $i = g$  or  $W[i] = \infty$  or  $Q = \emptyset$ ;
27  $p \leftarrow \emptyset$ 
28 if  $i = g$  then
29    $c \leftarrow g$ 
30   while  $c \neq 0$  do
31      $p \leftarrow \{c\} \cup p$ 
32      $c \leftarrow V[c]$ 
33 RETURN  $p$ 
```

D. The Pose SLAM Path Planning Algorithm

The path planning algorithm introduced in this paper is formally described in Algorithm 1. It searches for the path to the goal with the least probability of becoming lost during path execution. The algorithm takes as inputs the Pose SLAM graph M and the goal pose, g , which is assumed in M . Should this not be the case, the closest pose in the graph to g (in configuration space) is used as a goal. We first initialize a set Q with all the nodes in the graph (Line 1) and establish an initial cost W for the path to each node (Line 3) and a fake predecessor V for each node (Line 4). Then, the cost to reach the starting configuration is set to 0 (Lines 5 to 7). At this point the algorithm enters in a loop until the goal is reached or the reachable region from the start configuration is fully explored (Lines 8 to 26). At each iteration of the loop, we extract the node i with minimum cost from Q (Line 9). If this is not the goal (Line 10), we perform breadth first search on the neighbor nodes to i (Line 11). The neighboring nodes are determined using the procedure given in Section IV-A that takes into account the uncertainty in the pose estimates. For each one of the possible transitions to neighbors, we use the local planner to determine if the transition is possible and to compute the expected motion uncertainty (Line 13). Using this uncertainty and the marginal covariance for the target pose (Line 15) we compute the step uncertainty as described in Section IV-B (Line 16). Then, Line 17 computes the uncertainty increment for a motion from node i

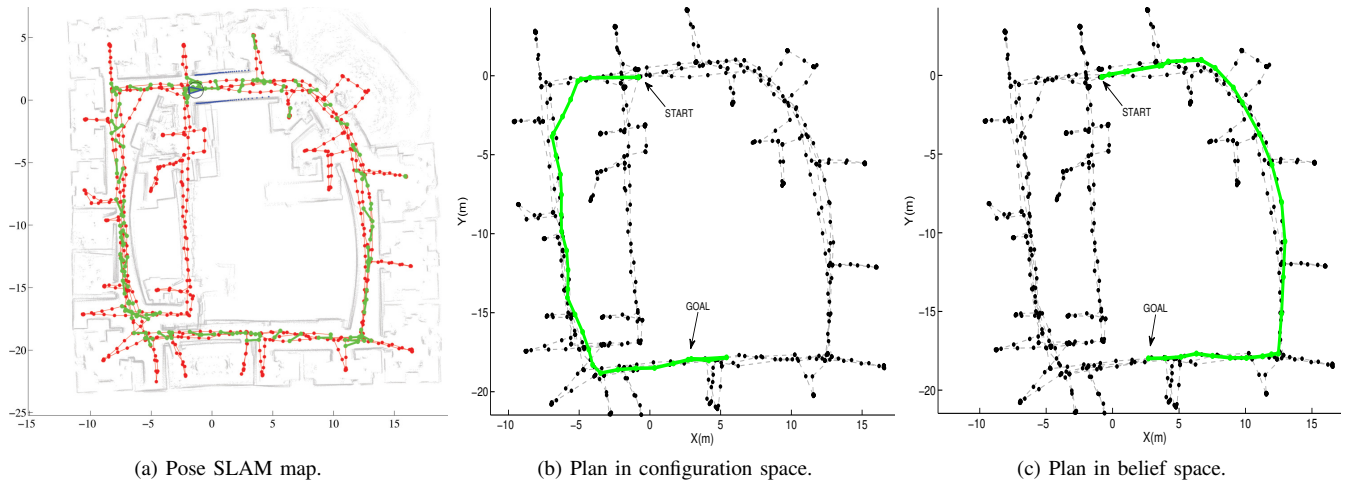


Fig. 1. Path planning over the Intel dataset. (a) Pose SLAM map built with encoder odometry and laser scans. The blue arrow indicates the final pose of the robot and the black ellipse the associated covariance at a 95% confidence level. (b) Planning in configuration space we obtain the shortest path to the goal on the underlying Pose SLAM graph. (c) Planning in belief space we obtain the most reliable path to the goal.

to node j . If this increment is positive, it is added to the path cost (Line 19). Otherwise, this step does not contribute to the overall path cost. If the cost of the new path to j is lower than the best known until that moment, the cost to reach j is updated (Line 23), we set i as the predecessor of j (Line 24), and we store the cost for the step reaching the node (Line 25). In the case of paths with equal cost, shorter ones are preferred and, since the costs of individual steps are non-negative, the considered paths never include cycles. If the goal is reached, the most reliable path to the goal is reconstructed using the chains to predecessor nodes stored in V (Lines 30 to 32). If the goal is determined to be non-reachable from the start configuration, an empty path is returned.

Without considering the cost of recovering the marginal covariances, the asymptotic cost of the algorithm is $O(e \log^2 n)$ with e the number of edges in the graph (i.e., the number of neighboring pose pairs) and n the number of nodes in the graph. This cost assumes that the nodes in Q are organized into a heap where the extraction of the minimum element is constant time and the update of the cost of an element is logarithmic. Moreover, it also assumes that poses are organized into a tree so that neighboring poses can be determined logarithmically [17]. If this search is performed linearly the cost increases to $O(en \log n)$.

Note that, when planning we do not need to maintain a localization estimate, but still we need to simulate registration with the map, for which the diagonal blocks of the covariance matrix are needed (Line 16). When using the Pose SLAM algorithm [17], these diagonal blocks are directly available [36], but this is not the case in other approaches [18]–[20]. In such cases, the most efficient way to compute the marginals is to invert the whole information matrix before starting to plan. One can efficiently invert it taking advantage of its sparsity using, for instance, sparse supernodal Cholesky decomposition [40]. For large-scale problems, however, this strategy becomes prohibitively expensive and we have to resort to approximations of the marginal covariances obtained using, for instance, Markov blankets [41]. Finally, should the map change significantly during path execution (i.e., a new highly informative loop closure is found), the Pose SLAM algorithm performs a full state update and re-planning is enforced.

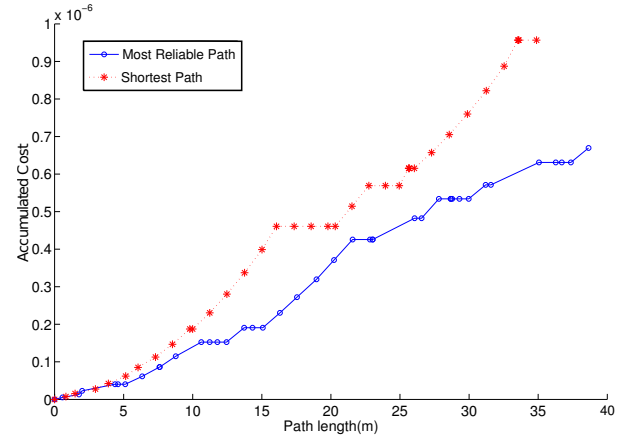


Fig. 2. Accumulated cost versus path length for the shortest path (red) and the most reliable path (blue) in the Intel experiment.

V. EXPERIMENTAL RESULTS

In order to evaluate the planning strategy introduced in this paper we show results with two data sets and in a real robot navigation experiment. The results for the first two tests were obtained with a Matlab implementation running on an Intel Core2 Quad system at 3 GHz with 4 GB of memory. For the third case, the system was implemented using the Robot Operating System (ROS) [42] on our 4-wheel robot Teo, a Segway RMP 400 platform.

A. Indoor Experiment

To test the performance of the algorithm on a widely used data set we choose data collected at the Intel Research Lab building in Seattle [43]. The dataset includes 26915 odometry readings and 13631 laser scans. The laser readings were used to generate scan-based odometry and to assert loop closures, by aligning them using an incremental closest point (ICP) scan matching algorithm [18]. In this case, only links between poses closer than ± 1 m in x and y , and ± 0.35 rad in orientation were considered reliable. These are also the thresholds used to determine neighboring poses when planning with $s = 0.1$. The robot odometry and the relative motion computed from laser scan matches were modeled with average noise covariances

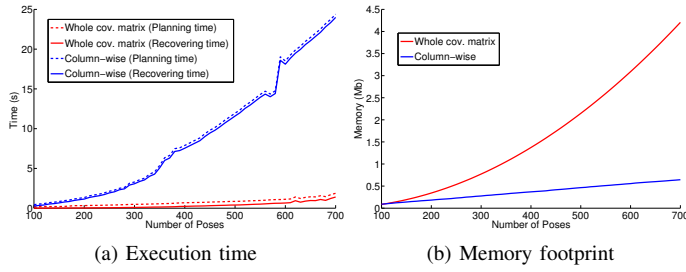


Fig. 3. Plots of execution time and memory footprint when planning with different subsets of the Intel map and employing two different strategies to recover marginals. (a) Execution time needed to recover only the marginals (continuous line) and for the whole planning algorithm (dashed line). (b) Memory footprint for marginal recovery.

$\Sigma_u = \text{diag}(0.05 \text{ m}, 0.05 \text{ m}, 0.03 \text{ rad})^2$ and $\Sigma_y = \text{diag}(0.05 \text{ m}, 0.05 \text{ m}, 0.009 \text{ rad})^2$, respectively and the minimum information gain was $\gamma = 4.5$ nats. Fig. 1(a) shows the path estimated by Pose SLAM together with the laser scans associated to each of the stored poses in light gray.

This map is the departing point of the planning algorithm and the goal is to connect two poses on opposite sides of the building. Frames (b) and (c) in Fig. 1 show the shortest and most reliable paths between the two poses. The apparent overshoot of the shortest path to the goal is due to the fact that the robot has to execute a 180 deg turn at the end of the path to align with the goal. Since sudden changes in orientation are not allowed by the kinematic constraints assumed for the robot, this rotation is only possible few meters away of the goal, in front of a door where many poses with the robot at different orientations accumulate.

Figure 2 shows the accumulated cost along the two paths. We can note that the accumulated increment in uncertainty of the shortest path is larger than that for the most reliable path. Therefore, following this second path the robot has lower probability of becoming lost at the cost of following a slightly larger path.

To test the efficiency of the method, Fig. 3 shows the execution time and memory footprint for planning, varying the number of poses in the Intel map. Since the most expensive step of the algorithm is the recovery of the marginal covariances, we applied two different strategies to recover them: recovering the whole Σ and recovering it column-wise as needed during planning. The continuous lines in Fig. 3(a) show the execution time needed to recover the marginals as a function of problem size, whereas the dashed lines show the execution time of the whole planning algorithm. The figure shows that recovering the whole matrix is computationally more efficient at the expense of increased memory space. On the contrary, on-the-fly computation of matrix columns results in repeated computations slowing down planner performance. The execution cost of re-planning when a graph edge is found to be non-traversable is reduced to the small difference between the continuous and the dashed lines in Fig. 3(a) since the map does not change and, thus, the marginal covariances do not need to be re-computed.

B. Large Scale Experiment

To demonstrate scalability, we tested our approach with a much larger map, for which memory space is a constraint. To this end, we planned paths using the simulated Manhattan data set [44] that includes over 10000 poses. In this experiment, average noise covariances for the robot odometry and the relative-pose measurements were set to $\Sigma_u = \Sigma_y = \text{diag}(0.05 \text{ m}, 0.05 \text{ m}, 0.03 \text{ rad})^2$, the threshold to detect neighboring poses was $s = 0.1$ searching in a

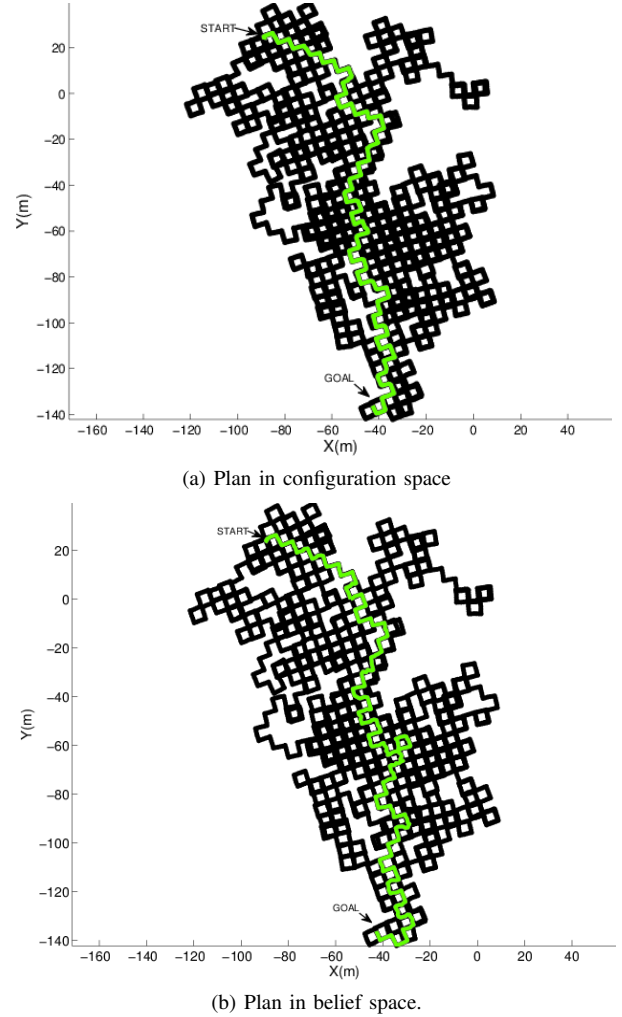


Fig. 4. Path planning over the Manhattan dataset. (a) Planning in configuration space we obtain the shortest path to the goal on the underlying Pose SLAM graph. (b) Planning in belief space we obtain the most reliable path to the goal.

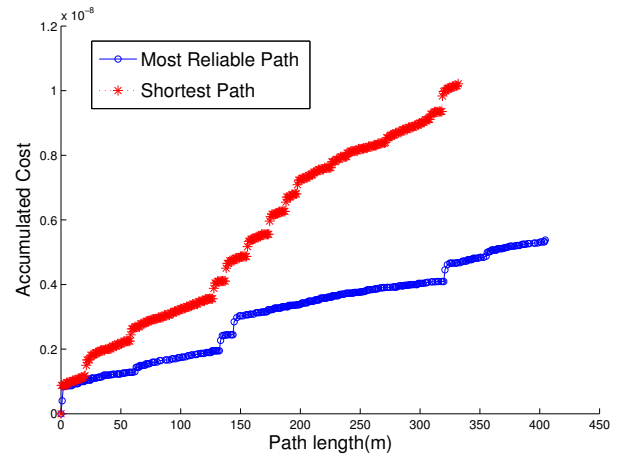


Fig. 5. Accumulated cost along the shortest (red) and the most reliable (blue) paths in the Manhattan experiment.

rectangle around the robot given by $\pm 8 \text{ m}$ in x , $\pm 8 \text{ m}$ in y , and $\pm 1 \text{ rad}$ in orientation. We only incorporated links between poses with an information gain above $\gamma = 9$ nats.

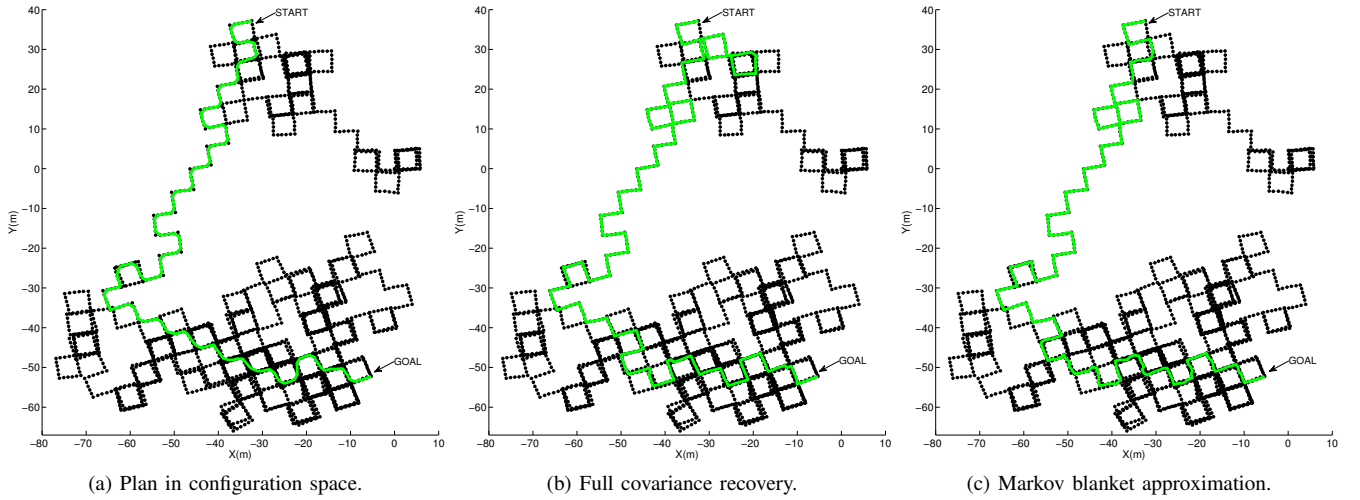


Fig. 6. Path planning over a section of the Manhattan dataset. (a) Planning in configuration space we obtain the shortest path to the goal on the underlying Pose SLAM graph. (b) Planning in belief space we obtain the most reliable path to the goal. (c) The most reliable path to the goal computed when the marginal covariances are recovered with Markov blankets.

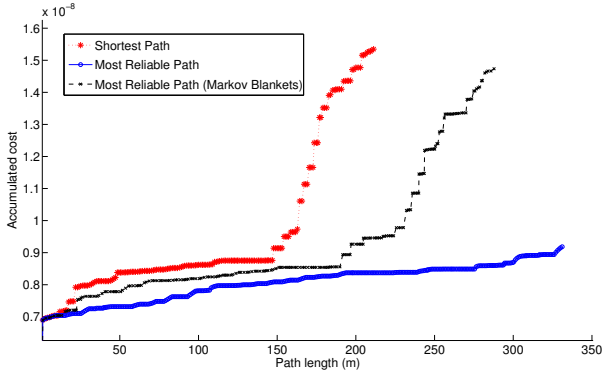


Fig. 7. Accumulated cost along the shortest (red) and along the most reliable path computed with exact marginal covariances (blue) and with Markov blankets (black).

Figure 4 shows the shortest and most reliable paths between the chosen start and goal poses, with the corresponding accumulated costs shown in Fig. 5.

With this dataset, full matrix recovery is not feasible with the computing resources used, and column-wise marginal computation is impractically slow. Therefore, marginal covariances are approximated using Markov blankets [41]. In this method the marginal covariance for a given pose is approximated considering only the subgraph of poses directly connected to it, which is typically small. As expected, the cost of the most reliable path obtained using Markov blankets is significantly better than that of shortest path, with only a marginal increase in path length. The planning time in this case is 122 s, which is reasonable considering that the planner was implemented in Matlab and that the problem includes more than 10000 poses. Thus, even when computing resources are a constraint, the presented method can still be used to plan a route for the robot to the best sensor registration regions at the expense of a possible degradation in the quality of the final path.

To analyze the effect of using approximated marginal covariances, the experiment with the Manhattan dataset was repeated, but this time using only a subset with the first 2700 poses, only to be able to compare the Markov blanket approximation with the computation of

exact covariances. Fig. 6 shows path planning results over this section of the Manhattan dataset. The use of the Markov blankets reduces the planning time by 50% but it hardly changes the obtained path, validating the approximation. Fig. 7 shows the accumulated cost in this experiment. As expected, the path length and cost when using the Markov blanket approximation are a compromise between the ones obtained with exact covariances and those of the shortest path.

C. Real Robot Navigation

To validate the planner in realistic conditions, we performed an experiment with a Segway RMP 400 robotic platform in an outdoor scenario with uneven and sandy terrain. We first acquired data to build a Pose SLAM map using dead-reckoning readings and laser scans over 350 m. The laser readings were used to assert loop closures by aligning them using an ICP algorithm. The Segway dead reckoning readings and the laser pose constraints were modeled with average noise covariances $\Sigma_u = \text{diag}(0.0316 \text{ m}, 0.0158 \text{ m}, 0.1104 \text{ rad})^2$, and $\Sigma_y = \text{diag}(0.2 \text{ m}, 0.2 \text{ m}, 0.03 \text{ rad})^2$, respectively, the uncertainty of the initial pose was set to $\Sigma_0 = \text{diag}(0.1 \text{ m}, 0.1 \text{ m}, 0.09 \text{ rad})^2$, and the minimum information gain was set to $\gamma = 1.5 \text{ nats}$. The local planner used was based on the dynamic window approach [45] available in ROS. Fig. 8 shows the path estimated by Pose SLAM. The red dots and lines represent the estimated path and the green lines indicate loop closure constraints established by registering scans at non-consecutive poses.

Using this map we computed the shortest and the most reliable paths that connect two robot configurations on opposite sides of the map, as shown in Fig. 9. For the planning, the thresholds to detect nearby poses are set to $\pm 4.5 \text{ m}$ in x , $\pm 4.5 \text{ m}$ in y or $\pm 1.04 \text{ rad}$ in orientation with $s = 0.1$.

The shortest path shown in Fig. 9(a), enters into an uneven and sandy region. The rugged terrain caused the laser to occasionally point to the soil which complicated the registration of the sensor readings. Moreover, the sand caused some slip that affected wheel odometry. Both effects contributed to produce a patch of the Pose SLAM map with higher uncertainty. In contrast, the path computed with our approach, shown in Fig. 9(b), avoids this region. This path traverses an even region of the environment in which the map has lower uncertainty, thanks to the better sensor registration and the more reliable odometry.

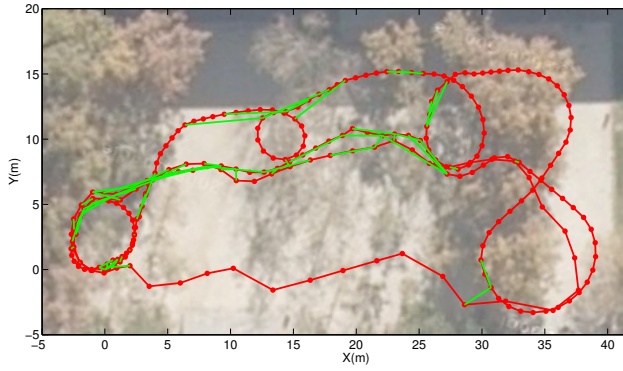


Fig. 8. Pose SLAM map built with encoder odometry and laser data in an outdoor scenario with a Segway RMP 400 robotic platform.

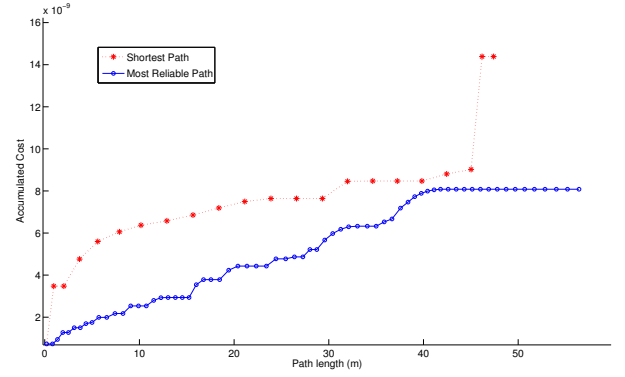
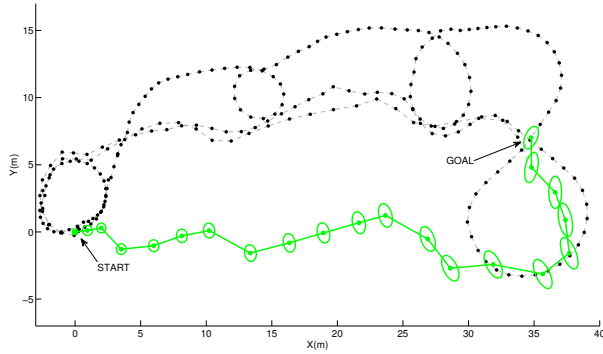
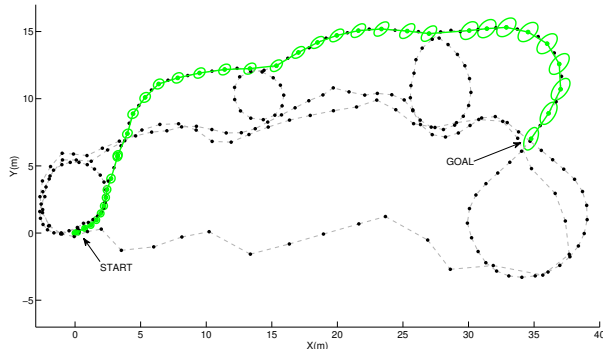


Fig. 10. Accumulated cost along the shortest (red) and the most reliable (blue) path in the real robot experiment.



(a) Shortest path to the goal.

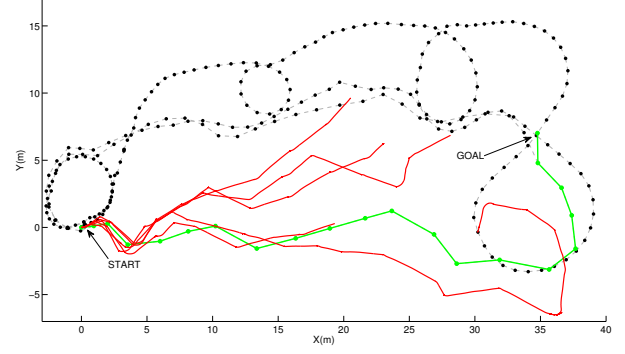


(b) Most reliable path to the goal.

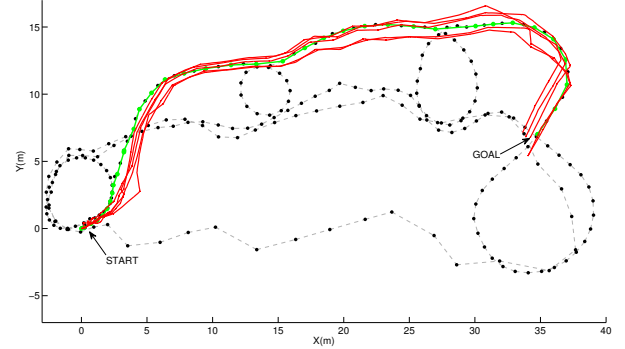
Fig. 9. Path planning over the map built with our mobile robot using encoder odometry and laser data.

Figure 10 shows the accumulated cost along the two paths. Note that the cost evaluation along the shortest path is actually underestimated since the motion uncertainty for the steps in this path is larger than the average noise used in the planning. In this case, the most reliable path is only 9 m longer than the shortest path and the total time to compute the plan was 6.5 s, which is significantly smaller than the 12 minutes required to execute it.

To verify that the assumptions taken in the planning hold in real conditions, we executed both paths with the robot for five times and the obtained trajectories are shown in Fig. 11. The supplementary material associated with this paper includes a video of this navigation experiment. The result of executing the shortest path are shown in Fig. 11(a). In this case, the robot was not able to reach the goal for any of the trials. On the contrary, the execution of the most reliable path, shown in Fig. 11(b), resulted in the robot safely arriving to the



(a) 5 attempts to execute the shortest path.



(b) 5 executions of the most reliable path.

Fig. 11. Real path execution of the shortest and most reliable paths to the goal with our mobile robot. The green line shows the planned paths computed with our method. The red lines represent the obtained trajectories when executing each path five times. The execution is interrupted when the deviation with respect to the intended plan is above a safety threshold.

goal in all trials, with an error in the interval of 0.5 m to 1.7 m.

VI. CONCLUSIONS

The work presented in this paper constitutes a step towards an integrated framework for mapping and planning for autonomous robots. We argue and show evidence that the poses of a Pose SLAM map can be readily used as nodes of a belief roadmap and thus, used for planning reliable routes. We also proposed a principled way to evaluate the cost of a path taking into account the uncertainty of traversing every edge in the map. The final path obtained is the most reliable among all the possible paths to the goal, increasing the chances to reach it. Three advantages of the proposed approach are that it is defined in the belief space, that it considers only the

uncertainty added when moving between poses, and that it scales to large environments using approximated marginal covariances. The presented approach is adequate for scenarios where a robot is initially guided during map construction, but autonomous during execution. For other scenarios in which more autonomy is required, the robot should be able to explore the environment without any supervision. We are currently working on this problem [34].

REFERENCES

- [1] R. M. Eustice, H. Singh, J. J. Leonard, and M. R. Walter, "Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1223–1242, 2006.
- [2] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hahnel, M. Montemerlo, A. Morris, Z. Omohundro, and C. Reverte, "Autonomous exploration and mapping of abandoned mines," *Robot. Automat. Mag.*, vol. 11, no. 4, pp. 79–91, Dec. 2004.
- [3] G. Sibley, C. Mei, I. Reid, and P. Newman, "Vast-scale outdoor navigation using adaptive relative bundle adjustment," *Int. J. Robot. Res.*, vol. 29, no. 8, pp. 958–980, 2010.
- [4] P. M. Newman, J. Leonard, J. Neira, and J. D. Tardós, "Explore and return: Experimental validation of real time concurrent mapping and localization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, May 2002, pp. 1802–1809.
- [5] J. Guivant, E. Nebot, J. Nieto, and F. Masson, "Navigation and mapping in large unstructured environments," *Int. J. Robot. Res.*, vol. 23, no. 4–5, pp. 449–472, Apr. 2004.
- [6] S. Rezaei, J. Guivant, J. Nieto, and E. Nebot, "Simultaneous information and global motion analysis ("SIGMA") for car-like robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, Apr. 2004, pp. 1939–1944.
- [7] L. Kavradi, P. Svestkaand, J. C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Trans. Robot.*, vol. 12, no. 4, pp. 566–580, 1996.
- [8] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.
- [9] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, May 2009, pp. 489–494.
- [10] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [11] P. Missiuro and N. Roy, "Adapting probabilistic roadmaps to handle uncertain maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, May 2006, pp. 1261–1267.
- [12] R. Pepp, M. Kieffer, and E. Walter, "Reliable robust path planner," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nice, Sep. 2008, pp. 1655–1660.
- [13] R. Alterovitz, T. Simeon, and K. Goldberg, "The Stochastic Motion Roadmap: A sampling framework for planning with Markov motion uncertainty," in *Robotics: Science and Systems III*, Atlanta, Jun. 2007.
- [14] A. Bry and N. Roy, "Rapidly-exploring Random Belief Trees for motion planning under uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, May 2011, pp. 723–730.
- [15] S. Prentice and N. Roy, "The Belief Roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. Robot. Res.*, vol. 29, no. 11–12, pp. 1448–1465, 2009.
- [16] R. He, S. Prentice, and N. Roy, "Planning in information space for a quadrotor helicopter in a GPS-denied environment," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, May 2008, pp. 1814–1820.
- [17] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact Pose SLAM," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 78–93, Feb. 2010.
- [18] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robot.*, vol. 4, no. 4, pp. 333–349, 1997.
- [19] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1100–1114, Dec. 2006.
- [20] K. Konolige and M. Agrawal, "FrameSLAM: from bundle adjustment to realtime visual mapping," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1066–1077, 2008.
- [21] N. Michael, E. Stump, and K. Mohta, "Persistent surveillance with a team of MAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, Sep. 2011, pp. 2708–2714.
- [22] A. Zini, "Robots expand delivery options," *Health Manag. Tech.*, pp. 10–12, Mar. 2011.
- [23] R. Valencia, J. Andrade-Cetto, and J. M. Porta, "Path planning in belief space with Pose SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, May 2011, pp. 78–83.
- [24] R. Kummerle, D. Hahnel, D. Dolgov, S. Thrun, and W. Burgard, "Autonomous driving in a multi-level parking structure," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, May 2009, pp. 3395–3400.
- [25] D. Dolgov and S. Thrun, "Autonomous driving in semi-structured environments: Mapping and planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, May 2009, pp. 3407–3414.
- [26] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Trans. Robot. Autom.*, vol. 17, no. 2, pp. 125–137, Apr. 2001.
- [27] F. Fraundorfer, C. Engels, and D. Nister, "Topological mapping, localization and navigation using image collections," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, Nov. 2007, pp. 3872–3877.
- [28] T. Goedeme, M. Nuttin, T. Tuytelaars, and L. V. Gool, "Omnidirectional vision based topological navigation," *Int. J. Comput. Vision*, vol. 74, no. 3, pp. 219–236, 2007.
- [29] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artif. Intell.*, vol. 99, no. 1, pp. 21–71, 1998.
- [30] M. A. Whitty and J. Guivant, "Efficient global path planning during dense map deformation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, May 2011, pp. 4943–4949.
- [31] —, "Efficient path planning in deformable maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Saint Louis, Oct. 2009, pp. 5401–5406.
- [32] K. Konolige, E. Marder-Eppstein, and B. Marthi, "Navigation in Hybrid Metric-Topological Maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, May 2011, pp. 3041–3047.
- [33] T. Kollar and N. Roy, "Efficient optimization of information-theoretic exploration in SLAM," in *Proc. 23th AAAI Conf. on Artificial Intelligence*, Chicago, Jul. 2008, pp. 1369–1375.
- [34] R. Valencia, J. Valls Miró, G. Dissanayake, and J. Andrade-Cetto, "Active Pose SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Algarve, Oct. 2012, pp. 1885–1891.
- [35] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, 1990, pp. 167–193.
- [36] V. Ila, J. Porta, and J. Andrade-Cetto, "Amortized constant time state estimation in Pose SLAM and hierarchical SLAM using a mixed Kalman-information filter," *Robot. Auton. Syst.*, vol. 59, no. 5, pp. 310–318, 2011.
- [37] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Robotics: Science and Systems VI*, Zaragoza, Spain, Jun. 2010.
- [38] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 635–645, Aug. 2010.
- [39] R. Sim and N. Roy, "Global A-optimal robot exploration in SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Apr. 2005, pp. 661–666.
- [40] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, "Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate," *ACM T. Math. Software*, vol. 35, no. 3, pp. 22:1–22:14, 2008.
- [41] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. J. Robot. Res.*, vol. 23, no. 7–8, pp. 693–716, Jul. 2004.
- [42] M. Quigley, B. Gerkey, K. Conley, T. F. J. Faust, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: An open-source robot operating system," in *Proc. IEEE ICRA Workshop Open Source Soft. Robot.*, Kobe, 2009.
- [43] A. Howard and N. Roy, "The robotics data set repository (Radish)," <http://radish.sourceforge.net>, 2003.
- [44] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, May 2006, pp. 2262–2269.
- [45] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robot. Automat. Mag.*, vol. 4, no. 1, pp. 23–33, 1997.