Path Planning in Belief Space with Pose SLAM

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

***Abstract*— The probabilistic belief networks that result from standard feature-based simultaneous localization and map building cannot be directly used to plan trajectories. The reason is that they produce a sparse graph of landmark estimates and their probabilistic relations, which is of little value to find collision free paths for navigation. In contrast, we argue in this paper that Pose SLAM graphs can be directly used as belief roadmaps. We present a method that devises optimal navigation strategies by searching for the path in the pose graph with lowest accumulated robot pose uncertainty, independently of the map reference frame. The method shows improved navigation results when compared to shortest paths both over synthetic data and real datasets.**

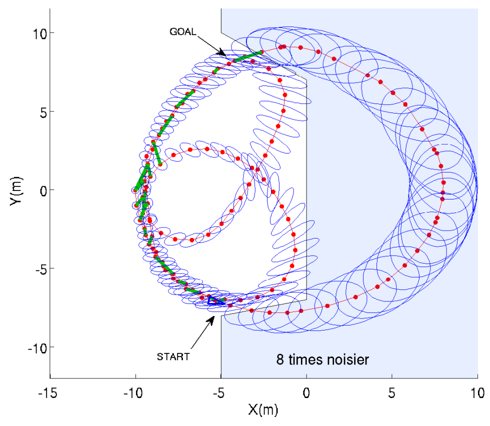
I. Introduction

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 as a prerequisite 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 manage thousands of features [3]. For efficiency reasons, most SLAM algorithms represent the environment using a sparse set of features. Unfortunately, this representation can not 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 they represent. Those sparse models could be somehow enriched with obstacle or traversability-related information [4], but at the expense of significant increased complexity.

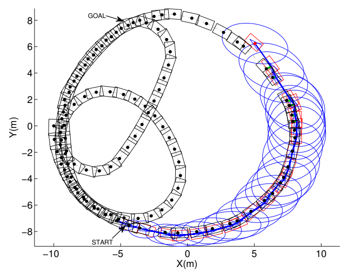
The problem of finding adequate trajectories to reach distant locations is addressed in the motion planning literature [5]. The most successful path planning methods are those based on randomized sampling such as the Probabilistic Roadmaps or the Rapidly-exploring Random Trees in which samples are stochastically drawn in the configuration space and, if possible, neighboring collision-free samples are connected via collision-free paths forming a roadmap. This roadmap is later used to connect any two given configurations. In these approaches, all collision free paths are considered valid and, thus, the focus is to determine the shortest path between the given start and goal configurations.

This work has been partially supported by the Spanish Ministry of Science and Innovation under projects DPI-2010-18449, DPI-2008-06022, and MIPRCV Consolider-Ingenio 2010; and by a PhD scholarship to R. Valencia from the Mexican Council of Science and Technology.

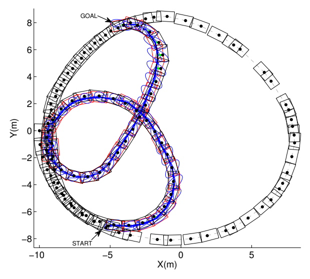
The authors are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens Artigas 4-6, Barcelona 08028, Spain {rvalencia,cetto,porta}@iri.upc.edu



(a)



(b)



(c)

Fig. 1. Path planning using the map generated by Pose SLAM. (a) The Pose SLAM graph. The red dots and lines represent the estimated trajectory, and the green lines indicate loop closure constraints established by registering sensor readings at different poses. (b) A plan in configuration space would produce the shortest path to the goal. At one point during path execution, sensor registration fails and the robot gets lost. (c) A plan in belief space produces the minimum uncertainty path to the goal. Plans with low uncertainty have larger probability of success.

Originally, the research in motion planning assumed deterministic setups where a perfect model of the environment was available and where the configuration of the robot was known too. Many extensions have been introduced recently to deal with different sources of uncertainty, either in the model of the environment [6], in the robot configuration [7], or in the effect of robot actions [8]. The extension that best matches the stochastic nature of the SLAM problem is the Belief Roadmap (BRM) [9]. In this approach, the edges defining the roadmap include information about the uncertainty change when traversing such edge. However, the main drawback of the BRM approach is that it still assumes a known model of the environment. In this paper, we aim to overcome this limitation noting that the map generated by Pose SLAM [10] (or in any other delayed-state SLAM algorithm [11, 12]) is perfectly suited to be used as a belief roadmap.

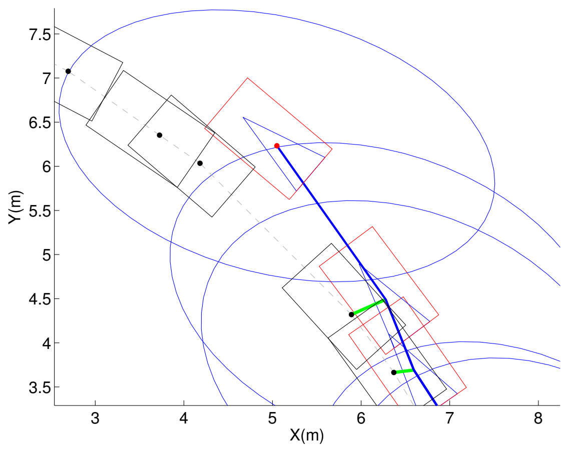


Fig. 2. Zoomed view of a region along the shortest path where the robot gets lost. The bad localization on this path leads the robot to deviate from the next way-point, producing failed sensor registration. The rectangles indicate the areas where sensor registration is reliable. The blue lines and ellipses represent the localization estimates.

Pose SLAM is the variant of SLAM where only the robot trajectory is estimated and where landmarks are only used to produce relative constraints between robot poses. Thus, the map in Pose SLAM only contains the trajectory of the robot. The poses stored in the map are, by construction, feasible and obstacle-free since they were already traversed by the robot when the map was originally built. Furthermore, since the robot trajectories are usually human-driven, they even satisfy mobility constraints not usually modeled in the robot controller, such as the existence of restricted traversable regions (grass or sidewalks), or the right of way along paths. In this paper, we show that, using this map, we can plan in the belief space to obtain paths to remote locations that take into account the uncertainty along the path (see Fig. 1). The main motivation behind our method is that, in Pose SLAM, poses with large uncertainty estimates lead to less reliable sensor registration. Therefore, a plan to navigate through these areas would suggest higher risk of becoming lost during path execution (see Fig. 2).

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. Other approaches that attempt to use SLAM for path planning either ignore the uncertainty in the robot pose [13, 14] or in the map [15] whereas our approach takes both of them into account. 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 model of the environment.

The rest of the paper is devoted to detail the extension of Pose SLAM to perform path planning. In Section II we summarize Pose SLAM and reinterpret its map as a set of samples in the belief space, and in Section III we describe how to plan using a roadmap defined on these samples. In Section IV, this new planning approach is tested with simulated and real data sets and, finally, Section V gives some concluding remarks.

II. Environment Sampling with Pose SLAM

Pose SLAM produces a directed graph in which the nodes are poses or way-points, and the edges are established through odometry or sensor registration of the environment. Assuming Gaussian distributions, a probabilistic estimate of the poses in the nodes, $\mathbf{x} = \{x_1, \ldots, x_k\}$, is maintained with a canonical parametrization $p(\mathbf{x}) = \mathcal{N}^{-1}(\boldsymbol{\eta}, \boldsymbol{\Lambda})$, using an information filter, with information vector $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}$, and information matrix $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$. This parametrization, compared to the traditional Kalman form (mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$) has the advantage of being exactly sparse [11].

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

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

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. Assuming the state mean to be 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 constrains can also be modeled with a compounded operation

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

where $h(x_k, x_i)$ gives 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 overall uncertainty. The result is that the marginal uncertainty for 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 uncertainty the plan is defined in the belief space.

During map building, the distance in the belief space of one pose with respect to the poses already in the map can be measured from the information carried by the links established between those poses. If links are low informative (i.e., if its information gain is below a threshold γ), there is no need to include the new pose in the map since it is too close to other poses in belief space.

Formally, the information gain of a link can be evaluated as [10]

$$\mathcal{I} = \frac{1}{2} \ln \frac{|\mathbf{S}|}{|\boldsymbol{\Sigma}_y|},$$

where $\mathbf{\Sigma}_y$ is the sensor registration error, $\mathbf{S}$ is the innovation covariance

$$\mathbf{S} = \mathbf{\Sigma}_y + [\mathbf{H}_i \, \mathbf{H}_k] \begin{bmatrix} \mathbf{\Sigma}_{ii} & \mathbf{\Sigma}_{ik} \\ \mathbf{\Sigma}_{ik}^\top & \mathbf{\Sigma}_{kk} \end{bmatrix} [\mathbf{H}_i \, \mathbf{H}_k]^\top,$$

$\mathbf{H}_i$, $\mathbf{H}_k$ are the Jacobians of h with respect poses i and k evaluated at the state means μ_i and μ_k, $\mathbf{\Sigma}_{ii}$ is the marginal covariance of pose i, and $\mathbf{\Sigma}_{ik}$ is the cross correlation between poses i and k.

One can say that in Pose SLAM, the sampling methodology is aware of both the motion and sensor models since nodes are added to the graph as a function of the information content in their connecting links.

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 and on the density of loop closures in that region. For this reason, different paths from the given start configuration to the goal node will not be uniformly distributed with respect to information content, and the accumulated relative uncertainty will vary between them.

III. Path Planning in Pose SLAM

During path planning we assume maximum likelihood actions and measurements, which implies that the mean estimate after a sequence of controls will lie at the mean of a node in the Pose SLAM graph and that the observation previously obtained at that position will be repeated. Given the Pose SLAM graph, and a goal destination, the objective of path planning is then to find an optimal collision-free path in the graph from the current robot pose to the goal.

A. Increasing Graph Connectivity with Guaranteed Reachability during Path Search

Note that only odometry-based links ensure the existence of collision-free transitions between poses. However, the graph with only odometry-based edges is sparse. Loosely connected graphs are not best suited for path planning and we need to increase the number of edges to allow the system to jump from one exploration sequence to another in the quest for an optimal path. Thus, beside odometry related poses, we consider the possible transition to all neighboring nodes during path planning.

Neighbor node search is computed by measuring the distance between query nodes and their candidate neighbors. The relative displacement, d, from the current robot pose x_k to any other previous pose in the trajectory x_i can be estimated as a Gaussian with parameters

$$\mu_d = h(\mu_k, \mu_i),$$
$$\mathbf{\Sigma}_d = [\mathbf{H}_i \, \mathbf{H}_k] \begin{bmatrix} \mathbf{\Sigma}_{ii} & \mathbf{\Sigma}_{ik} \\ \mathbf{\Sigma}_{ik}^\top & \mathbf{\Sigma}_{kk} \end{bmatrix} [\mathbf{H}_i \, \mathbf{H}_k]^\top .$$

Marginalizing the distribution of the displacement, d, for each one of its dimensions, r, we get a one-dimensional Gaussian distribution $\mathcal{N}(\mu_r, \sigma_r^2)$ that allows to compute the probability of pose x_i being closer than v_r to pose x_k along such dimension

$$p_r = \int_{-v_r}^{+v_r} \mathcal{N}(\mu_r, \sigma_r^2) = \frac{1}{2}\left(\text{erf}\left(\frac{v_r - \mu_r}{\sigma_r \sqrt{2}}\right) - \text{erf}\left(\frac{-v_r - \mu_r}{\sigma_r \sqrt{2}}\right)\right).$$

If for all dimensions, p_r is above a given threshold s, then configuration x_i is considered kinematically reachable from the current configuration, x_k.

In many cases there will not exist a collision free path between neighboring poses. These cases, however, can be easily detected during path execution, the poses be removed from the list of neighbors, and a re-plan process be triggered. One advantage of the method is that the original odometry-based links present in the Pose SLAM map ensure the existence of collision-free way-outs for every pose, thus guaranteeing reachability.

B. Minimum Uncertainty along a Path

Given that candidate paths lie on top of the graph, we can safely assume that, after path execution, sensor registration will close a loop and 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. We are interested instead in those paths that maintain the robot well localized throughout the whole trajectory.

We now propose a cost function that considers cumulative relative uncertainty during localization, independent of the map reference frame. Finding trajectories that accumulate the least uncertainty can be seen as searching for a path of minimal mechanical work in an information surface [16, 17] over the space of robot poses. In this case, the cost of traversing a link from node x_i to node x_j is proportional to the conditional entropy at node j given full confidence about node i, $H(x_j|x_i)$, which for Gaussians is proportional to

$$H(x_j|x_i) \propto |\bar{\mathbf{\Sigma}}_{jj} - \bar{\mathbf{\Sigma}}_{ji}\bar{\mathbf{\Sigma}}_{ii}^{-1}\bar{\mathbf{\Sigma}}_{ij}|, \qquad (1)$$

where the marginals and cross-correlations are extracted from $\bar{\mathbf{\Sigma}}$, the covariance of the compound localization estimate (x_i, x_j).

Given a discrete trajectory $\mathbf{u}_{1:T}$, we define its mechanical work in the information surface as the sum of relative entropy increments $\Delta H_i = H(x_{i+1}|x_i) - H(x_i|x_{i-1})$ along the path

$$W(T) = \sum_{i=1}^{T} \Delta H_i \quad \forall \Delta H_i > 0.$$

Thus, the minimal uncertainty path corresponds to the path that accumulates the least positive variations of uncertainty.

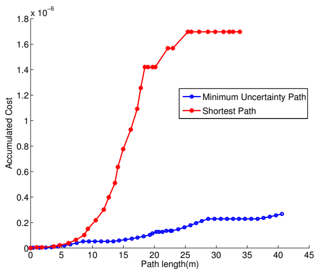
Equation 1 is a measure of the robot's ability to safely track its position during path execution. To compute both marginals and cross correlation terms in Eq. 1 we need to track localization estimates of the previous and current robot poses x_i and x_j. That is, we compute the compound localization estimate (x_i, x_j) using the Extended Kalman Filter (EKF), with the particularity that every EKF update is given by sensor registration with the Pose SLAM graph at node j, taking into account its marginal covariance Σ_{jj}.

```
PoseSLAMPathPlanning($M$,$g$)
  Inputs:
    $M$: The map computed by Pose SLAM.
    $g$:  The goal pose.
  Outputs:
    $p$:  Minimum uncertainty path to the goal pose.
1:  $m \leftarrow$ NumPoses($M$)
2:  $Q \leftarrow \{1, \ldots, m\}$
3:  $d[1, \ldots, m] \leftarrow \infty$
4:  $v[1, \ldots, m] \leftarrow 0$
5:  $s \leftarrow$ CurrentPose($M$)
6:  $d[s] \leftarrow 0$
7:  $H[s] \leftarrow 0$
8:  $\mathbf{\Sigma}[s] \leftarrow$ CurrentMarginalCovariance($M$)
9:  while $g \in Q$ do
10:     $i \leftarrow$ ExtractMin($Q, d$)
11:     if $i \neq g$ then
12:         for all $j \in$ (Neighbors($M, i$) $\cap Q$) do
13:             $\bar{\mathbf{\Sigma}} \leftarrow$ GetPosterior($M, i, j, \mathbf{\Sigma}[i]$)
14:             $H(j|i) = |\bar{\mathbf{\Sigma}}_{jj} - \bar{\mathbf{\Sigma}}_{ji}\bar{\mathbf{\Sigma}}_{ii}^{-1}\bar{\mathbf{\Sigma}}_{ij}|$
15:             $\Delta H = H(j|i) - H[i]$
16:             if $\Delta H > 0$ then
17:                 $d' = d[i] + \Delta H$
18:             else
19:                 $d' = d[i]$
20:             end if
21:             if $d[j] < d'$ then
22:                 $d[j] \leftarrow d'$
23:                 $v[j] \leftarrow i$
24:                 $\mathbf{\Sigma}[j] \leftarrow \bar{\mathbf{\Sigma}}_{jj}$
25:                 $H[j] \leftarrow H(j|i)$
26:             end if
27:         end for
28:     end if
29: end while
30: return  ReconstructPath($v, g$)
```

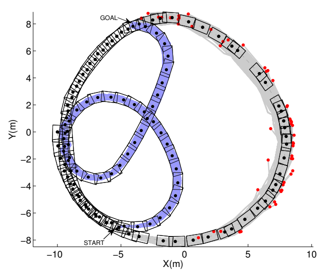
Algorithm 1: Path planning using the poses maintained in the Pose SLAM map and a minimum uncertainty criteria to select the optimal path.

C. The Pose SLAM Path Planning Algorithm

The path planning method introduced in this paper is formally described in Algorithm 1. This algorithm implements a minimum uncertainty path search on the graph implicitly defined by the neighboring relations between the poses stored in the map built by Pose SLAM. The distance between nodes is computed from relative entropy measures obtained simulating maximum likelihood localization estimates. The algorithm takes as inputs the Pose SLAM map M and the goal pose, g, which is assumed in M. Should this not be the case, the closest pose in the map to g (in configuration space) is used as a goal. The robot initializes the set of nodes not yet visited, Q, with all the nodes in the graph (Lines 1-2) and establishes an initial cost for the path to each node (Line 3) and a fake predecessor for each node (Line 4). Then, the cost to reach the starting configuration is set to zero (Lines 5-7), the marginal covariance at that node is read from the map (Line 8), and we enter in a loop until we reach the goal (Lines 9-29). At each iteration of the loop, we extract the node i with minimum cost from Q (Line 10). If this is not the goal, we perform breadth first search on the neighbor nodes to i already in Q. The neighboring nodes are determined using the procedure given in Section III-A that takes into account the uncertainty in the pose estimates. Line 15 computes the cost to reach each neighbor j from i



(a)



(b)

Fig. 3. (a) Accumulated cost along the shortest (red) and minimum uncertainty (blue) path in the simulated experiment. (b) Monte Carlo realization with 100 runs of the simulated experiment. The minimum uncertainty path guarantees path completion during localization. The red dots indicate points where the robot gets lost due to a missed sensor registration.

using the path cost criterion described in Section III-B. If this path is cheaper than the best known until this moment, the cost to reach j is updated, we set i as the predecessor of j, we update the marginal covariance for the best path to the node, and we store the marginal entropy for this node (Lines 22-25). When the goal is reached, the minimum uncertainty path to the goal is reconstructed using the chains to predecessor nodes stored in v (Line 30).

The asymptotic cost of the algorithm is $O(e \log^2 n)$ with e the number of edges in the graph and n the number of nodes. 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. If the search is performed linearly the cost increases to $O(e\, n\, \log\, n)$.

Note that, when planning in the belief space we need to simulate registration with the map during localization, for which marginals of the Pose SLAM covariance matrix are needed (Line 13). The most efficient way to compute these marginals is to invert the whole information matrix before starting to plan. Despite its presumably large size, one can efficiently invert it taking advantage of its sparsity using, for instance, sparse supernodal Cholesky decomposition [18]. As it will be shown in Section IV, the cost of searching for the optimal path in the graph is small compared to the cost of recovering the state marginals.

Finally, should a map change significantly during path execution (i.e., a new highly informative loop closure is found), replanning is enforced. Note that this is seldom the case since the optimal path traverses already visited regions in the environment as best localized as possible. Moreover, re-traversing a path on an already optimized map will seldom lead to map improvements as no new information is introduced. The map can only be improved or extended by joining different paths closing a loop or when exploring new paths to cover a larger area. However, exploration is out of the scope of the paper.

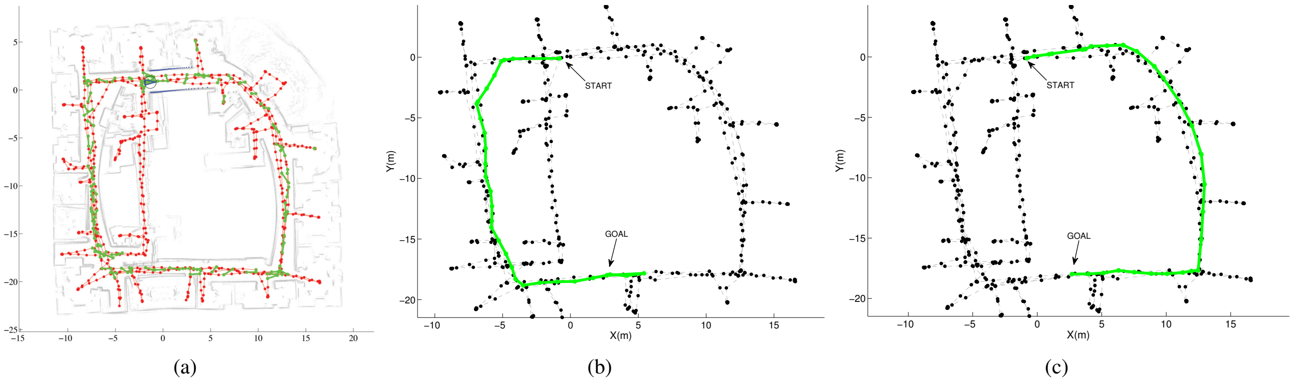


Fig. 4. Path planning over the Intel dataset. (a) Pose SLAM map built with encoder odometry and laser scans of the Intel dataset. The blue arrow indicates the final pose of the robot and the black ellipse the associated covariance at a 95% confidence level. (b) Planning in configuration space we obtain the shortest path to the goal. (c) Planning in belief space we obtain the minimum uncertainty path to the goal.

IV. Experimental Results

In order to evaluate the planning strategy presented in this paper we show results on two cases. The first one is a simulated environment used to illustrate the basic principles of the paradigm. The second one is a test with a publicly available dataset that shows its performance in real conditions.

In the first experiment, we simulate a robot moving over a trajectory with several loops. In the simulation, the motion of the robot is measured with an odometric sensor whose error is 5% of the displacement in x and y, and $0.0175\,\text{rad}$ in orientation. A second sensor is able to establish a link between any two poses closer than $\pm 1.25\,\text{m}$ in x, $\pm 0.75\,\text{m}$ in y, and $\pm 0.26\,\text{rad}$ in orientation. This sensor is simulated with noise covariance $\mathbf{\Sigma}_y = \text{diag}(0.2\,\text{m}, 0.2\,\text{m}, 0.009\,\text{rad})^2$. Fig. 1(a) shows the final map as estimated by the Pose SLAM algorithm. The shadowed area simulates harsher navigation conditions with odometry and loop closure errors increased by a factor of 8. This noisier area simulates a part of the environment with less features and where constraints between poses are harder to be established.

After building the map using Pose SLAM we planned the path from the last robot pose to a particular goal selected from the nodes in the map. Fig. 1(b) shows the trajectory to the goal using a shortest path criterion, and Fig. 1(c) shows the trajectory obtained when using the minimum uncertainty criterion introduced in Section III-B.

Fig. 3(a) shows a plot of the accumulated cost along the two trajectories. The accumulated uncertainty of the shortest path is significantly larger than that of the minimum uncertainty path. Therefore, following this second trajectory there is increased guarantee that the robot will be better localized all along the path and will less likely get into trouble, for instance, of getting lost. This is verified in Fig. 3(b) that shows a Monte Carlo realization of the this experiment with 100 runs. Navigation through the shortest path reached the goal only 45% of the times due to failed sensor registration along the path, whereas navigating over the minimum uncertainty path always reached the final destination since the trajectory avoids the noisier area in the environment.

To test the performance of the planning technique over real data we used the data set collected at the Intel Research Lab building (Seattle) [19]. The dataset includes 26915 odometry readings and 13631 laser scans. The laser scans are used to generate sensor-based odometry and to assert loop closures, by aligning them using an ICP scan matching algorithm [20]. In this case, only links between poses closer than $\pm 1\,\text{m}$ in x and y, and $\pm 0.35\,\text{rad}$ in orientation were considered reliable. These are also the thresholds used to determine neighboring poses when planning. The robot odometry and the relative motion computed from laser scan matches are modeled with noise covariances $\mathbf{\Sigma}_u = \text{diag}(0.05\,\text{m}, 0.05\,\text{m}, 0.03\,\text{rad})^2$ and $\mathbf{\Sigma}_y = \text{diag}(0.05\,\text{m}, 0.05\,\text{m}, 0.009\,\text{rad})^2$, respectively. Finally, the covariance of the initial pose is set to $\mathbf{\Sigma}_0 = \text{diag}(0.1\,\text{m}, 0.1\,\text{m}, 0.09\,\text{rad})^2$. Fig. 4(a) shows the trajectory 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 process starts from the last robot pose. The goal is selected at the opposite side of the building. Figures 4(b) and (c) show the shortest and minimum uncertainty paths between the two poses. The apparent overshoot of the shortest path trajectory at the goal is due to the fact that the robot has to execute a $180\,\text{deg}$ turn at the end of the trajectory to align with the goal. This rotation is only possible few meters away of the goal, in front of a door where many samples with the robot at different orientations accumulate.

Fig. 5(a) shows the accumulated cost along the two trajectories. As in the simulated case, the accumulated uncertainty of the shortest path along the trajectory is larger than that for the minimum uncertainty trajectory. Therefore, following this second trajectory the robot is better localized all along the path. Figure 5(b-c) shows the execution time and memory footprint for planning with different subsets of the Intel map, with varying number of poses, using a non-optimized Matlab code. It shows two different strategies for recovering the marginals: recovering the whole $\mathbf{\Sigma}$ and recovering them column-wise as needed during planning. The continuous line in Fig. 5(b) shows the execution time for recovering the marginals and the dashed line shows the execution time of the

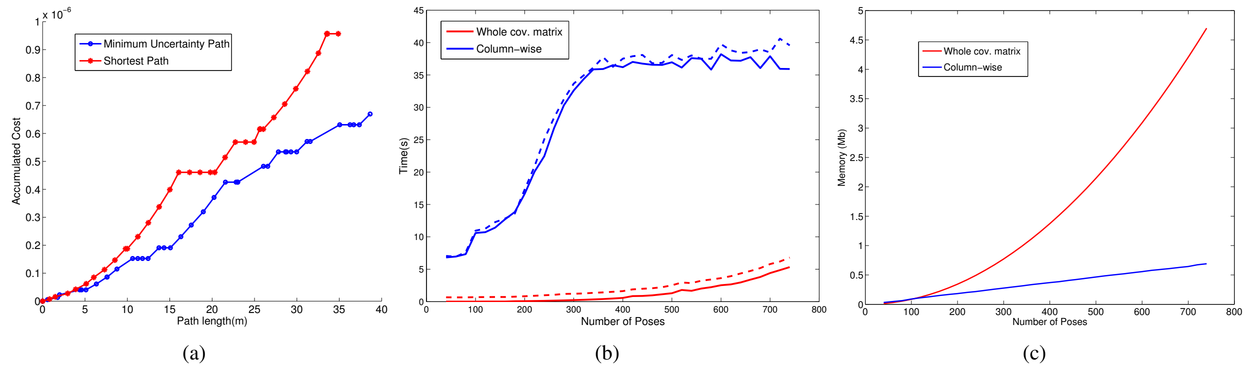


Fig. 5. (a) Accumulated cost along the shortest (red) and minimum uncertainty (blue) path in the Intel experiment. (b-c) Plots of execution time and memory footprint when planning with different subsets of the Intel map and employing two different strategies for recovering the marginals. (b) Execution time for recovering the marginals (continuous line) and for the whole planning algorithm (dashed line). (c) Memory footprint for recovering the marginals.

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 column-wise elements of the matrix results in repeated computations since these are not stored during plan search. A strategy of compromise would be to store the matrix columns computed during search. But, for searches that need to explore the entire graph without any pruning strategy, the space cost will be the same as that of full matrix inversion. In any case, full matrix inversion can be computed in reasonable time for sparse systems such as ours. If memory space is a constraint, we suggest instead to use approximation techniques to recover marginals, such as for instance, Markov blankets [21].

V. Conclusion

This work constitutes a step towards an integrated framework for exploration, mapping, and planning for autonomous robots. We presented a planning method showing how the poses of a Pose SLAM map can be readily used as nodes of a belief roadmap and, thus, used for planning minimum uncertainty routes. We also proposed a principled way to evaluate the cost of a path taking into account the uncertainty of traversing every edge. The final path obtained by the planner is the safer among all the possible paths to the goal, increasing the chances to reach it. Two advantages of the proposed metric are that it is defined in the belief space and that it encodes only the relative information between poses, independently of the map reference frame. Lastly, one aspect that is beyond the scope of this work is exploration. When the goal pose is not included in the map, the robot must autonomously explore the environment to find it. We leave this problem for future work.

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,” *IJRR*, vol. 25, no. 12, pp. 1223–1242, 2006.

[2] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hannel, M. Montemerlo, A. Morris, Z. Omohundro, and C. Reverte, “Autonomous exploration and mapping of abandoned mines,” *RAM*, vol. 11, no. 4, pp. 79–91, 2004.

[3] U. Frese and L. Schröder, “Closing a million-landmarks loop,” in *IROS*, Beijing, 2006, pp. 5032–5039.

[4] J. Nieto, J. Guivant, and E. Nebot, “DenseSLAM: Simultaneous localization and dense mapping,” *IJRR*, vol. 25, no. 8, pp. 711–744, 2006.

[5] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2004.

[6] P. Missiuro and N. Roy, “Adapting probabilistic roadmaps to handle uncertain maps,” in *ICRA*, Orlando, 2006, pp. 1261–1267.

[7] R. Pepy, M. Kieffer, and E. Walter, “Reliable robust path planner,” in *IROS*, Nice, 2008, pp. 1655–1660.

[8] R. Alterovitz, T. Simeon, and K. Goldberg, “The Stochastic Motion Roadmap: A sampling framework for planning with Markov motion uncertainty,” in *RSS*, Atlanta, 2007.

[9] S. Prentice and N. Roy, “The Belief Roadmap: Efficient planning in belief space by factoring the covariance,” *IJRR*, vol. 29, no. 11-12, pp. 1448–1465, 2009.

[10] V. Ila, J. M. Porta, and J. Andrade-Cetto, “Information-based compact Pose SLAM,” *T-RO*, vol. 26, no. 1, pp. 78–93, 2010.

[11] R. M. Eustice, H. Singh, and J. J. Leonard, “Exactly sparse delayed-state filters for view-based SLAM,” *T-RO*, vol. 22, no. 6, pp. 1100–1114, 2006.

[12] K. Konolige and M. Agrawal, “FrameSLAM: from bundle adjustment to realtime visual mapping,” *T-RO*, vol. 24, no. 5, pp. 1066–1077, 2008.

[13] R. Kummerle, D. Hahnel, D. Dolgov, S. Thrun, and W. Burgard, “Autonomous driving in a multi-level parking structure,” in *ICRA*, Kobe, 2009, pp. 3395–3400.

[14] G. Sibley, C. Mei, I. Reid, and P. Newman, “Vast-scale outdoor navigation using adaptive relative bundle adjustment,” *IJRR*, vol. 29, no. 8, pp. 958–980, 2010.

[15] D. Dolgov and S. Thrun, “Autonomous driving in semi-structured environments: Mapping and planning,” in *ICRA*, Kobe, 2009, pp. 3407–3414.

[16] R. Sim and N. Roy, “Global A-optimal robot exploration in SLAM,” in *ICRA*, Barcelona, 2005, pp. 661–666.

[17] L. Jaillet, J. Cortés, and T. Siméon, “Sampling-based path planning on configuration-space costmaps,” *T-RO*, vol. 26, no. 4, pp. 635–645, 2010.

[18] 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.

[19] A. Howard and N. Roy, “The robotics data set repository (Radish),” http://radish.sourceforge.net, 2003.

[20] F. Lu and E. Milios, “Globally consistent range scan alignment for environment mapping,” *Auton. Robot.*, vol. 4, no. 4, pp. 333–349, 1997.

[21] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” *IJRR*, vol. 23, no. 7-8, pp. 693–716, 2004.