Technical Report

IRI-TR-10-03

Path Planning with Pose SLAM
IRI Technical Report

Rafael Valencia
Juan Andrade-Cetto
Josep M. Porta

Institut de Robòtica i Informàtica Industrial

Abstract

The probabilistic belief networks that result from standard feature-based simultaneous localization and map building (SLAM) approaches 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 (BRMs). The original BRM algorithm assumes a known model of the environment from which probabilistic sampling generates a roadmap. In our work, the roadmap is built on-line by the Pose SLAM algorithm. The result is a hybrid BRM-Pose SLAM method that devises optimal navigation strategies on-line by searching for the path with lowest accumulated uncertainty for the robot pose. The method is validated over synthetic data and standard SLAM datasets.

Institut de Robòtica i Informàtica Industrial (IRI)
Consejo Superior de Investigaciones Científicas (CSIC)
Universitat Politècnica de Catalunya (UPC)
Llorens i Artigas 4-6, 08028, Barcelona, Spain

Tel (fax): +34 93 401 5750 (5751)
`http://www.iri.upc.edu`

Corresponding author:

Rafael Valencia
tel: +34 93 4015780
`rvalencia@iri.upc.edu`
`http://www.iri.upc.edu/people/rvalencia`

1 Introduction

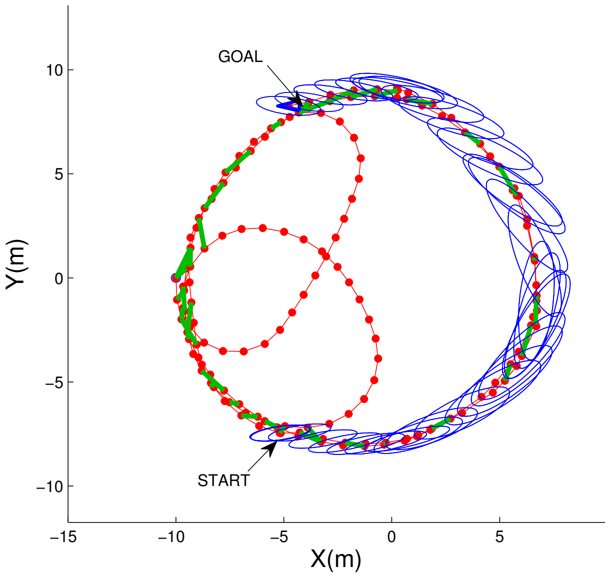
Aside from applications such as the reconstruction of archaeological sites [9] or the inspection of dangerous areas [48], the final objective for an autonomous robot is not to build a map of the environment, but to use this map as a pre-requisite for navigation, i.e., to reach distant locations in the environment efficiently and safely. In recent years, we have witnessed an amazing advance in the field of simultaneous localization and map building (SLAM). Whereas in the seminal approaches to SLAM [41] only few tens of landmarks could be managed, state of the art approaches can now efficiently manage thousands of landmarks [10, 21, 47]. 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 the obstacles in the environment. Those sparse models could be somehow enriched with obstacle-related information [33], but then the complexity of the system increases considerably.

The motion planning community has achieved remarkable success in path planning even in high dimensionality configuration spaces. The most successful path planning methods are those based on randomized sampling [19, 26]. In those approaches, 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. All paths in the configuration space are considered equally valid and, thus, the focus in this field is to determine the shortest path between the given start and goal configurations. 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 perfectly known too. Despite this, in the last years many extensions have been introduced to deal with different sources of uncertainty, either in the model of the environment [31], in the configuration of the robot [34], or in the effect of robot actions [1]. The extension that best matches the stochastic nature of the SLAM methods is the Belief Roadmap (BRM) [37]. In this approach, the edges defining the roadmap include information about the uncertainty change when traversing the corresponding edge. This uncertainty variation is encoded using a factorization that allows to reuse the roadmap for any possible query. 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 [15] is perfectly suited to be used as a belief roadmap.

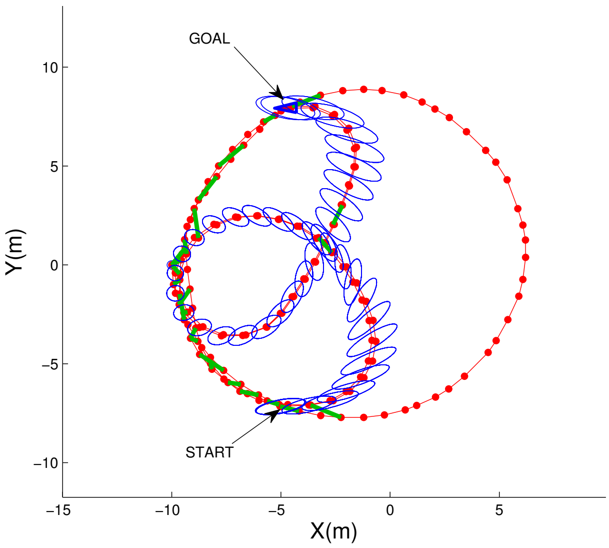
Pose SLAM is the variant of SLAM where only the robot trajectory is estimated and where landmarks are only used to produce relative constraints between robot poses [8, 23]. Thus, the map in Pose SLAM only contains the trajectory of the robot formed by collision free configurations. In this paper we show that, using this map, we can plan in the belief space to obtain paths to remote locations that take into account not only the travelled distance but also the uncertainty along the path (see Fig. 1). The main motivation behind our method is that, in Pose SLAM, poses in areas with less reliable sensor registration have larger uncertainty. Therefore, the uncertainty in the poses provides information on the expected capability of the robot to reliably localize in a given area, and thus, on the paths to prefer or avoid during navigation.

The particular nature of Pose SLAM forces some modifications to the original objectives and methods of the BRM approach. First, path planning over a Pose SLAM map is basically a single query path planner and, thus, it is not necessary to factorize the covariance with respect to the initial belief conditions. Moreover, in Pose SLAM different paths that reach a single node are fused to provide an accurate estimate of the robot pose at that node (i.e., loop closure). This makes all paths to a particular node end up with the same uncertainty, despite having varying uncertainty in intermediate poses for different paths. Thus, a cost function that considers the overall uncertainty along the path is to be devised. Besides this, map construction in Pose SLAM allows for a distribution of samples that takes into account the sensor characteristics giving an approximately uniform distribution of samples in the information space [16] and not in the configuration space as it is usually done in motion planning.

From the point of view of Pose SLAM, this paper constitute a step forward to actually use the out-



(a) Planning in configuration space we obtain the shortest path to the goal.



(b) Planing in information space we obtain the minimum uncertainty path to the goal.

Figure 1: Path planning using the map generated by a Pose SLAM algorithm.

put 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 model of the environment.

After a brief review of the state of the art in path planning under uncertainty (Section 2), the rest of the paper is devoted to detail the extension of Pose SLAM to perform path planning. In Section 3 we briefly summarize Pose SLAM and reinterpret its map as a set of samples approximately equidistributed in the information space and in Section 4 we describe how to plan using a roadmap defined on these samples. In Section 5, the new planning approach is tested with simulated and with real data sets and, finally, Section 6 gives some concluding remarks and points to aspects of the new approach deserving further attention.

2 Related Work

Partial Observable Markov Decision Processes (POMDP) provide the most general framework for planning under uncertainty. In a POMDP, the knowledge about the agent's state is encoded as a *belief* (a probability distribution over all possible states) and the objective is to determine a *policy* giving the best action for each belief. Robot navigation is naturally modeled as a continuous problem but, unfortunately there are very few approaches able to deal with POMDPs in continuous spaces [36, 45]. Thus, the usual approach discretizes the problem and applies standard value iteration algorithms [17, 42]. Unfortunately, those algorithms can only deal with problems of low dimensionality. Point-value iteration algorithms [35, 43] somehow alleviate these problems focusing the planning to the reachable belief space. However, they are not efficient enough to be applied to large-scale navigation problems. Another inconvenient of standard POMDP algorithms is that they assume a known model of the environment.

In the context of robot navigation, model learning at a large scale is addressed by the SLAM community. The first solutions to SLAM formulate the problem as the probabilistic estimation of the robot pose and the location of static landmarks, jointly modeled as a multivariate Gaussian and updated with the Extended Kalman Filter (EKF) [6, 41]. However, maintaining the fully correlated covariance matrix of the robot pose and the map of features has quadratic computational complexity, limiting the approach to relatively small environments. This computational cost can be alleviated using the Extended Information Filter (EIF) and its alternative parametrization of Gaussian distributions based on the information vector and the information matrix. The information matrix in landmark-based SLAM is approximately sparse with very small matrix entries for distant landmarks [46]. These entries can be removed, compacting the map and speeding up the filter. If instead of only estimating the last robot pose, the whole robot trajectory is included in the state together with the landmarks (an approach typically referred to as full SLAM [18, 32]) a sparse information matrix is obtained without using approximations. Going one step further, in Pose SLAM [8, 14, 22, 28] only the trajectory of the robot is included in the state and the landmarks are only used to derive relative motion constraints between poses. The result is an exactly sparse information matrix which grows with the number of poses and that only has non-null entries for those poses directly related by an observation. Therefore, Pose SLAM only requires moderate memory resources even when mapping large areas. In any case, SLAM is a passive framework where the robot only estimates the model of the environment, but without taking any decisions on the trajectory of the robot.

Motion planning [25] deals with the problem of finding adequate trajectories to reach distant locations. For low dimensional spaces approximate cell decomposition provides solutions to the motion planning process by discretizing the environment in cells, selecting obstacle-free ones and finding a shortest path to the goal using a standard shortest path algorithm. However, for higher dimensional spaces the approach becomes unfeasible due to the curse of dimensionality. The solution is to resort to a roadmap-based method. A roadmap is a collection of one-dimensional curves that capture the topology of the configuration space. Paths between two given configurations are obtained traveling over the roadmap. The silhouette method [2] defines roadmaps with a guarantee of completeness, but can only be applied to small problems. Stochastic variants such as Probabilistic Roadmap (PRMs) or Rapidly Expanding RandomTrees (RRTs) only offer probabilistic completeness, but can be successfully applied to problems with high dimensionality [19, 26]. The main issues with classical motion planning algorithms when applied to navigation is that they assume a known model of the environment and that they do not take into account the inherent noise in robot motion.

Several combinations of planning under uncertainty, SLAM, and motion planning exist. Planning under uncertainty and mapping are combined in approaches that attempt to simultaneously capture the environment model and optimize the policy [4, 38]. Up to now, those approaches are only valid for relatively small problems. Motion planning taking into account uncertainty is also present in recent contributions that consider the noise in the environment model [31] or in the robot pose due to the uncertain effects of actions [1, 11, 34, 37, 39]. SLAM and motion planning are combined in approaches that select the best actions in order to efficiently explore a given environment [20, 29, 30, 40] or in approaches where a map learned using a SLAM algorithm is used in a standard motion planning algorithm [24, 49] sometimes after extracting the environment structure in a sort of roadmap [7]. Another way to combine those two approaches is by interfacing RRTs with particle-based SLAM [13]. In this case, particles can be seen as known configurations and a standard RRT can be grown for each one of them. This simple integration of the two approaches comes at the computational cost of managing several RRTs separately.

In our approach, we combine the three disciplines: We use a Pose SLAM algorithm to acquire a model of the environment, and an information-based planner to select trajectories taking into account the uncertainty both in the acquired model and in the robot's pose.

3 Environment Sampling with Pose SLAM

Belief roadmaps were originally conceived as a belief-state variant of probabilistic roadmaps (PRMs). As with PRMs, BRMs are constructed by probabilistic sampling in configuration space of a given environment model. In this work, we argue that the set of poses defining the map in Pose SLAM [15] (or in any other delayed-state SLAM algorithm) can be used as the starting point for a BRM. This choice overcomes the implicit assumption when using BRMs that a model of the environment exists to produce collision-free samples and trajectories between them. The poses stored in the map by Pose SLAM during exploration 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 such robot controllers, such as the existence of restricted traversable regions (grass or sidewalks), or the right of way along paths. Another advantage of using the Pose SLAM map in planning, is that it can be constructed on the fly, giving the possibility to perform path planning right from the start of the mapping process, and to perform re-planning as needed, should the map change.

The result of the Pose SLAM algorithm is a directed graph, in which the nodes are poses or waypoints, and the edges are established through odometry or sensor registration of the environment. Assuming Gaussian distributions, a probabilistic estimate of the poses defining the nodes in the graph, $\mathbf{x} = \{x_1, \ldots, x_k\}$, 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 [8].

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

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

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 a naturally sparse information matrix, $\boldsymbol{\Lambda}$, 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 produces relative motions, but now between non-consecutive poses. These relative constrains can also be modeled with a compounded operation

$$z_{ik} = h(x_i, x_k) = \ominus x_i \oplus x_k, \tag{2}$$

where $h(x_t, x_i)$ gives the relative displacement from x_i to x_k in the frame of reference of x_i. When establishing such a link, the update operation only modifies the diagonal blocks i and k of the sparse 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 sampling, 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 information space.

The distance in the information space of a 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 none of the links is informative enough, there is no need to include the new pose in the map since it is too close to other poses in the information space. By keeping only the poses with a minimum information distance to previous poses, the set of nodes maintained by Pose SLAM will tend to be approximately equidistributed in the information space.

Formally, the information gain of a link can be evaluated as [15]

$$\mathcal{I} = \frac{1}{2} \ln \frac{|\mathbf{S}|}{|\mathbf{\Sigma}_y|} \tag{3}$$

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

$$\mathbf{S} = \mathbf{\Sigma}_y + [\mathbf{H}_i \, \mathbf{H}_k] \begin{bmatrix} \mathbf{\Sigma}_{ii} & \mathbf{\Sigma}_{ik} \\ \mathbf{\Sigma}_{ik}^\top & \mathbf{\Sigma}_{kk} \end{bmatrix} [\mathbf{H}_i \, \mathbf{H}_k]^\top, \tag{4}$$

and $\mathbf{H}_i$, $\mathbf{H}_k$ are the Jacobians of h with respect to the i-th and k-th components of the state, evaluated at the state means $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_k$, $\mathbf{\Sigma}_{ii}$ is the marginal covariance of pose i, and $\mathbf{\Sigma}_{ik}$ is the cross correlation between poses i and k. If $\mathcal{I}$ is below a given threshold g, for all possible links, the pose is too close to other poses in the information space and can be safely discarded. Note that the constant g is only a lower bound on the information content on any link but there is not any upper bound. The consequence is that the determinants of the marginal covariances of the poses in the map may vary significantly depending on the information content leading to such pose.

One can say that in Pose SLAM, the sampling methodology is not probabilistic, but instead it 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.

4 Path Planning in Pose SLAM

In the same spirit as with BRMs, we assume during path planning, that measurements are maximum likelihood observations, i.e., $z_{ik} = h(\mu_i, \mu_k)$, which implies that, the mean estimate after a sequence of controls will lie at the mean of a node in the Pose SLAM graph. Given this 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.

In the original BRM algorithm, one step covariance transfer functions were used to implement composition of EKF updates. For multiple pair shortest path searches, belief variation along path segments could then be reused, independent of the marginal covariances at the start and end nodes. Even when Pose SLAM also allows for a similar covariance factorization [15], it is not needed, since belief path segments are not reused for multiple queries. The initial location for navigation is the current pose, and we need only a single query path planner.

4.1 Increasing Graph Connectivity during Path Search

The initial graph for path planning is the Pose SLAM graph without sensor-derived links. The Pose SLAM graph cannot be directly used as loop-closure edges are not necessarily traversable. Only odometry-based links ensure the existence of collision-free transitions.

This graph is sparse with nodes approximately equidistributed in the information space. Loosely connected graphs however 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.

Neighbor node search is computed during path planning 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

$$\mu_d = h(\mu_k, \mu_i), \tag{5}$$

$$\mathbf{\Sigma}_d = [\mathbf{H}_i \; \mathbf{H}_k] \begin{bmatrix} \mathbf{\Sigma}_{ii} & \mathbf{\Sigma}_{ik} \\ \mathbf{\Sigma}_{ik}^\top & \mathbf{\Sigma}_{kk} \end{bmatrix} [\mathbf{H}_i \; \mathbf{H}_k]^\top \; . \tag{6}$$

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

$$\begin{aligned} p_r &= \int_{-v_r}^{+v_r} \mathcal{N}(\mu_r, \sigma_r^2) \\ &= \frac{1}{2}\left(\text{erf}\left(\frac{v_r - \mu_r}{\sigma_r\,\sqrt{2}}\right) - \text{erf}\left(\frac{-v_r - \mu_r}{\sigma_r\,\sqrt{2}}\right)\right) . \end{aligned} \tag{7}$$

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, and the node is added to the path search queue.

In many cases there will not exist a collision free path between neighboring poses. These cases, however, can be easily detected, the poses be removed from the list of neighbors, and a re-plan process be triggered. Moreover, odometry-based links ensure the existence of collision-free way-outs for every pose, thus guaranteeing reachability.

4.2 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.

Some regions of the environment may have better features for localization than others [39] [Takeda pami94]. This fact is reflected in the estimated graph with Pose SLAM. In Pose SLAM, poses in areas with less reliable sensor registration have larger uncertainty. Therefore, the uncertainty in the poses provides information on the expected capability of the robot to reliably localize in a given area, and thus, on the paths to prefer or avoid during navigation. This is the motivation behind our path planner. By minimizing belief uncertainty at any given configuration we are reducing the likelihood of the robot becomes lost.

Belief uncertainty at any given configuration, x in the information space is proportional to the entropy at that point, which for Gaussians is a scalar of $|\mathbf{\Sigma}|$. For a parametrized continuous trajectory $\tau(t)$, $t \in [0, 1]$ [5], the cost of the trajectory is

$$J = \int_0^1 |\mathbf{\Sigma}(t)| \, dt. \tag{8}$$

For a discrete trajectory, $\mathbf{u}_{1:T}$, we approximate this continuous cost as

$$J \approx \sum_{i=1}^{T} \rho(u_i) \, |\mathbf{\Sigma}_{ii}|, \tag{9}$$

where ρ is a distance metric in the space where the path is defined (the information space in this case) giving the cost of moving from one belief point to the next. Since nodes in the belief network are approximately equidistributed in the information space, the distance can be assumed constant, and it suffices to find the path that accumulates the least uncertainty along the path

$$J \propto \sum_{i=1}^{T} |\mathbf{\Sigma}_{ii}| \tag{10}$$

To find the optimal path, Dijkstra's algorithm is implemented.

Another possibility would be to search for the path with min-max marginal uncertainty along the path. This strategy is aimed to avoid single unreliable belief nodes. The strategy however, as in the case of minimum final uncertainty, does not take into account the cost of actions and could return as optimal a very large path along a valley of low uncertainty.

4.3 The Pose SLAM Roadmap Algorithm

The path planning algorithm introduced in this paper is formally described in Algorithm 1. This algorithm implements a minimum uncertainty path search on a graph where the graph is implicitly defined by the neighboring relations between the poses stored in the map built by Pose SLAM. 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 a set with all the nodes in the graph (Lines 1-2) and establishes and initial cost for the path to each node (Line 4) and a fake predecessor for each node (Line 5). Then, the cost to reach the starting configuration is set to 0 (Lines 7-8) and we enter in a loop until we reach the goal (Lines 10-25). At each iteration of the loop, we extrat the node n with minimum cost from $\mathbf{Q}$ (Line 11). If this is not the goal, we consider all the nodes n' close to n (Line 15). The neighboring nodes are determined using the procedure given in Section 4.1 that takes into account the uncertainty in the pose estimates. Line 18 computes the cost to each neighbor n' passing by n using the path cost criterion described in Section 4.2. If this path is cheaper than the best known until this moment, the cost to reach n' is updated (Line 20) and we set n as the predecessor of n' (Line 21). When the goal is reached (Line 12), the search finishes and the minimum uncertainty path to the goal is reconstructed using the chains to predecessor nodes stored in v (Lines 26-33). If the parameters to select neighboring poses are set too restrictively, we could end up with a disconnected graph where the goal can not be reached from the start pose. In this pathological case the algorithm will return an empty path.

The asymptotic cost of the shortest path algorithm is $O(e \log^2 m)$ with e the number of edges in the graph (i.e., the number of neighboring pose pairs) and m the number of nodes in the graph. This cost assumes that the nodes in $\mathbf{Q}$ 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 $O(e\, m \log m)$.

Note that, when planning in the information space we need to marginalize parts of the covariance matrix (Line 15). The most efficient way to do that is by inverting the whole information matrix before starting to plan since, marginalization from the covariance matrix is straightforward. Despite the size of the information matrix, we can efficiently invert it taking advantage of its sparsity using, for instance, sparse supernodal Cholesky decomposition [3].

Computing the inverse of the information matrix has largest asymptotic cost of all planning steps. However, in practice this inverse is computed in near linear time for sparse systems such as ours. Moreover, for large maps the problem is not execution time but memory usage. In these cases we suggest to use approximation techniques to recover marginals, such as for instance, Markov blankets.

Finally, should a map change significantly during path execution (i.e., a new highly informative loop closure), the Pose SLAM algorithm performs a full state update and replanning should be enforced. Note

Pose SLAM Path Planning(M,g)

Inputs:

M: The map computed by Pose SLAM.

g: The goal pose.

Outputs:

p: Minimum uncertainty path to the goal pose.

```
m ← NumPoses(M)
Q ← {1, ..., m}
for all n ∈ Q do
    d[n] ← ∞
    v[n] ← 0
end for
s ← CurrentPose(M)
d[s] ← 0
f ← False
while Not(f) And Q ≠ ∅ do
    n ← ExtractMin(Q, d)
    if n = g then
        f ← True
    else
        N ← Neighbors(M, n)
        for all n' ∈ N do
            Σ ← GetMarginalCovariance(M, n')
            d' = d[n] + |Σ|
            if d[n'] < d' then
                d[n'] ← d'
                v[n'] ← n
            end if
        end for
    end if
end while
p ← ∅
if f then
    c ← g
    while c ≠ 0 do
        p ← {c} ∪ p
        c ← v[c]
    end while
end if
return p
```

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

that this is seldom the case since the optimal path is traversing 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. Thus, re-traversing "bad" paths will still leave them "bad". The map can only be extended or improved when exploring new paths to cover a larger area or by joining different paths closing a loop. However, exploration is out of the scope of the paper.

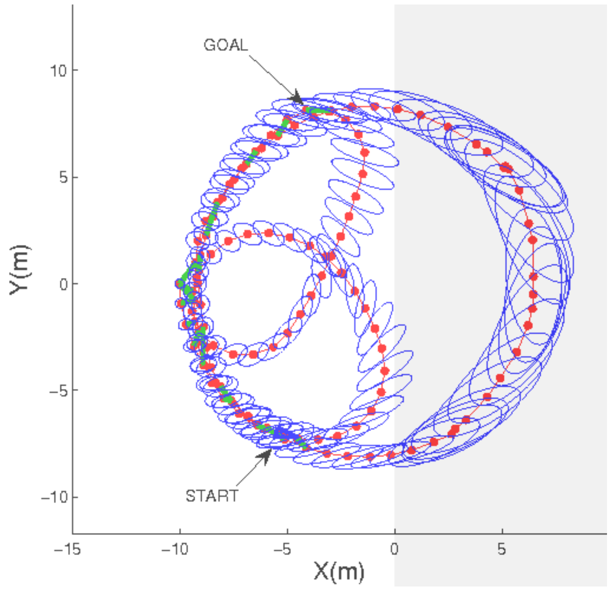


Figure 2: Map obtained with Pose SLAM in a simulated environment.

5 Experimental Results

In order to evaluate the planning approach presented in this paper we addressed two problems. The first one is a simulated environment used to illustrate the basic principles of the paradigm. The second one is a test with a public available dataset to show the performance of the approach in realistic conditions.

In the first experiment, we simulate a robot moving over a given 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 rad in orientation. A second sensor is able to establish a link between any two poses closer than ± 3 m in x and y, and ± 0.26 rad in orientation. This sensor is simulated with noise covariance $\mathbf{\Sigma}_y = \mathrm{diag}(0.2\,\mathrm{m}, 0.2\,\mathrm{m}, 0.009\,\mathrm{rad})^2$. Fig. 2 shows the final map as estimated by the Pose SLAM algorithm. In the shadowed area of Fig. 2 the odometry and loop closure errors are increased by a factor of 4. This more noisy area simulates a part of the environment with less features and where constraints between poses are harder to be defined.

After acquiring 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(a) shows the trajectory to the goal using a shortest path criterion. Fig. 1(b) shows the trajectory when using the minimum uncertainty criteria introduced in Section 4.2. We can see that the planned trajectory departs from the shortest path in order to avoid the noisy area in the environment.

Fig. 3 shows a plot of the determinant of the marginal covariance for the robot poses at each step in the two trajectories. We see that the two trajectories start and end with the same uncertainty, but the accumulated uncertainty of the shortest path along the trajectory is much larger than that of the minimum uncertainty trajectory. 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.

To test the performance of the planning technique on a realistic problem, we used the data set collected at the Intel Research Lab building (Seattle), which is available at [12]. 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 [28]. 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. The robot

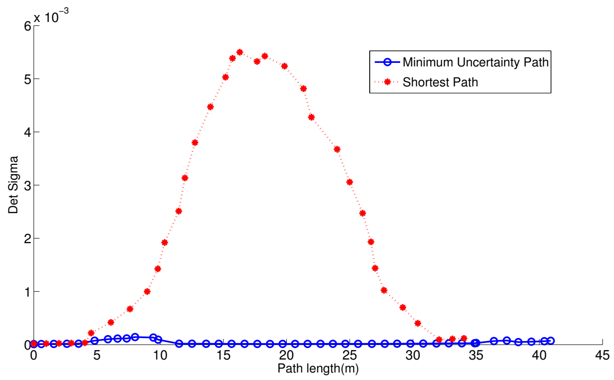
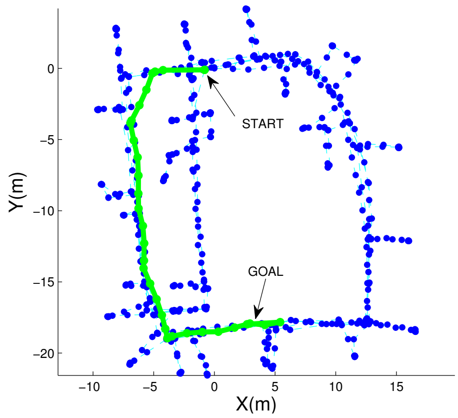
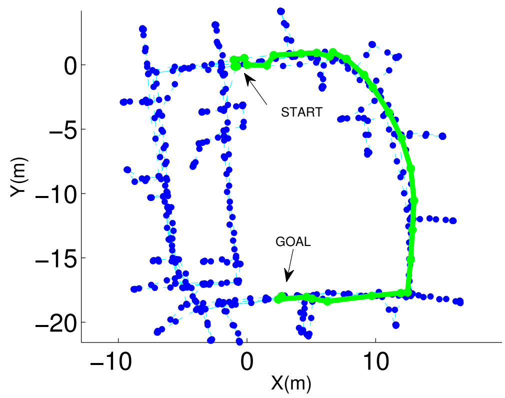


Figure 3: Marginal cost of the paths obtained in the simulated experiment.

Figure 4: Filtered trajectory using encoder odometry and laser scans of the Intel dataset.



(a) Planning in configuration space we obtain the shortest path to the goal



(b) Planing in information space we obtain the minimum uncertainty path to the goal.

Figure 5: Path planning in the Intel dataset

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 shows the trajectory estimated by Pose SLAM together with the laser scan associated with each of the stored poses in light grey.

Due to its large size, this data set is often pre-processed and reduced to about 1000 poses with about 3500 loop closure links [18]. The Pose SLAM system we use automatically selects a uniformly distributed optimal subset of poses in the sense of information gain and not with respect to Euclidean distance, allowing for a more principled selection of loop closure links and nodes rendering only 1218 poses approximately equidistant in the information space.

This map of poses determined by the Pose SLAM 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. Fig. 5 shows the minimum 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 in different orientations accumulate.

Fig. 6 shows the determinant of the marginal covariance for the robot pose along the two trajectories. As in the simulated case, the uncertainty of the shortest path along the trajectory is higher than that for the

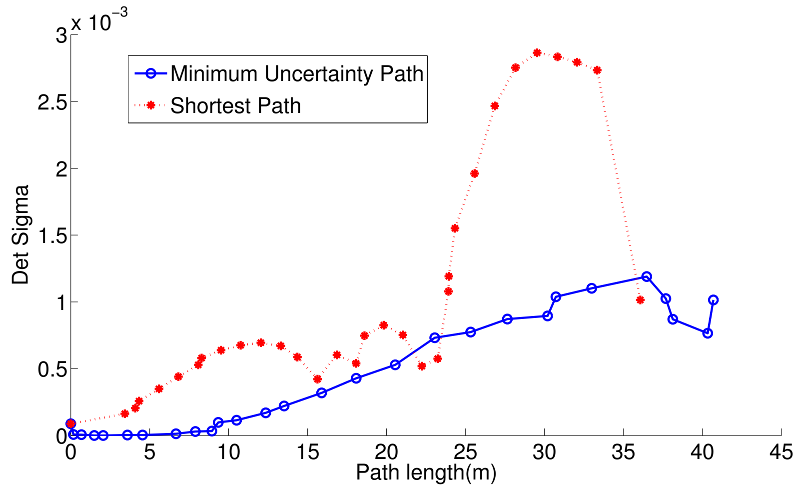


Figure 6: Marginal cost of the paths obtained in the Intel experiment.

minimum uncertainty trajectory. Therefore, following this second trajectory the robot is better localized all along the path. The decrease in uncertainty in the last step of the shortest path trajectory is due to the fact that, at the end of the trajectory, a loop is closed with the goal pose cancelling out the accumulated odometric error.

It might be the case for larger trajectories with poor connectivity (and consequently larger marginal uncertainty) that sensor registration might fail during path execution at a poorly localized configuration. At that point, the robot might get lost and fail to execute the path. The minimum accumulated uncertainty method provides optimal conditions to guarantee the best possible sensor registration at any instance during path execution.

6 Conclusion

This work constitutes a step towards an integrated framework for exploration, mapping, and planning for autonomous robots. The departing point of this work is an existing planning algorithm, the BRM, that can deal with noisy actions and can integrate the effect of extereoceptive observations in the planning, but assumes the knowledge of a model of the environment. In this paper, we integrated this planning framework with Pose SLAM. We show how the poses in the Pose SLAM map can be taken as samples for a belief roadmap and used for planning. An advantage of using the Pose SLAM graph versus any other delayed-state SLAM graph is that Pose SLAM has its nodes approximately equidistributed in the information space. This allows us to propose a principled way to evaluate the cost of a path simultaneously taking into account the length of the path and the uncertainty in the poses. This is in contrast with existing approaches that either take into account the path length or the path uncertainty, and heuristic combinations of the two is always problematic since they are defined in different spaces and with different units.

Aspects that are beyond of the scope of this work are path execution and exploration. During path execution the map can substantially change due to loop closure or some of the planned actions can actually be unfeasible due to changes in the environment. If this occurs, the plan must be recomputed from the pose of the robot at that point. This is not yet included in the current algorithm. Moreover, when the goal pose is not included in the map, the robot must autonomously explore the environment to find it. To deal with exploration the presented algorithm has to be either extended to include obstacle related information, in which case an algorithm such as the D* could be used [44], or rely on minimalistic path planning approaches, such as the bug algorithms [27].

References

[1] 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.

[2] J. F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, 1988.

[3] 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*, 35(3), 2008.

[4] P. Dallaire, C. Besse, S. Ross, and B. Chaib-draa. Bayessian reinforcement learning in continuous POMDPs with Gaussian processes. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 2604–2609, Saint Louis, Oct. 2009.

[5] R. Diankov and J. Kuffner. Randomized statistical path planning. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 1–6, San Diego, Nov. 2007.

[6] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Automat.*, 17(3):229–241, Jun. 2001.

[7] D. Dolgov and S. Thrun. Autonomous driving in semi-structured environments: Mapping and planning. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 3407–3414, Kobe, May 2009.

[8] R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Trans. Robot.*, 22(6):1100–1114, Dec. 2006.

[9] R. M. Eustice, H. Singh, J. J. Leonard, and M. R. Walter. Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information flters. *Int. J. Robot. Res.*, 25(12):1223–1242, 2006.

[10] U. Frese and L. Schröder. Closing a million-landmarks loop. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 5032–5039, Beijing, Oct. 2006.

[11] J. P. Gonzalez and A. Stentz. Planning with uncertainty in position using high-resolution maps. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 1015–1022, Rome, Apr. 2007.

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

[13] Y. Huang and K. Gupta. RRT-SLAM for motin planning with motion and map uncertainty for robot exploration. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 1077–1082, Nice, Sep. 2008.

[14] V. Ila, J. Andrade-Cetto, R. Valencia, and A. Sanfeliu. Vision-based loop closing for delayed state robot mapping. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 3892–3897, San Diego, Nov. 2007.

[15] V. Ila, J. M. Porta, and J. Andrade-Cetto. Information-based compact Pose SLAM. *IEEE Trans. Robot.*, 26(1):78–93, Feb. 2010.

[16] V. Ila, J.M. Porta, and J. Andrade-Cetto. Reduced state representation in delayed state SLAM. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 4919–4924, Saint Louis, Oct. 2009.

[17] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101:99–134, 1995.

[18] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robot.*, 24(6):1365–1378, 2008.

[19] L. Kavraki, Latombe P. Svestkaand J.-C, and M. Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Trans. Robot.*, 12(4):566–580, 1996.

[20] T. Kollar and N. Roy. Trajectory optimization using reinforcement learning for map exploration. *Int. J. Robot. Res.*, 27(2):175–197, 2008.

[21] K. Konolige. Large-scale map-making. In *Proc. 19th AAAI Conf. on Artificial Intelligence*, pages 457–463, San Jose, California, Jul. 2004.

[22] K. Konolige and M. Agrawal. FrameSLAM: from bundle adjustment to realtime visual mapping. *IEEE Trans. Robot.*, 24(5):1066–1077, 2008.

[23] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. In *Robotics: Science and Systems V*, Seattle, USA, Jun. 2009.

[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. Automat.*, pages 3395–3400, Kobe, May 2009.

[25] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic, London, 1991.

[26] S. LaValle and J. Kuffner. Randomized kinodinamid plannig. *Int. J. Robot. Res.*, 20(5):378–400, 2001.

[27] Steven M LaValle. *Planning Algorithms*. Cambridge University Press, 2004.

[28] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Auton. Robot.*, 4(4):333–349, 1997.

[29] A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte. An experiment in integrated exploration. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 534–539, Lausanne, Oct. 2002.

[30] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet. A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Auton. Robot.*, 27(2):93–103, 2009.

[31] P. Missiuro and N. Roy. Adapting probabilistic roadmaps to handle uncertain maps. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 1261–1267, Orlando, May 2006.

[32] M. Montemerlo and S. Thrun. *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*, volume 27 of *Springer Tracts in Advanced Robotics*. Springer, 2007.

[33] J. Nieto, J. Guivant, and E. Nebot. DenseSLAM: Simultaneous localization and dense mapping. *Int. J. Robot. Res.*, 25(8):711–744, 2006.

[34] R. Pepy, M. Kieffer, and E. Walter. Reliable robutst path planner. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 1655–1660, Nice, Sep. 2008.

[35] J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proc. Int. Joint Conf. Artificial Intell.*, Acapulco, Aug. 2003.

[36] J. M. Porta, N. Vlassis, M. T. J. Spaan, and P. Poupart. Point-based value iteration for continuous pomdps. *J. Mach. Learning Res.*, 7:2329–2367, 2006.

[37] S. Prentice and N. Roy. The Belief Roadmap: Efficient planning in belief space by factoring the covariance. *Int. J. Robot. Res.*, 29(11-12):1448–1465, 2009.

[38] S. Ross, B. Chaib-Draa, and J. Pineau. Bayesian reinforcement learning in continuous POMDPs with application to robot navigation. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2845–2851, Pasadena, May 2008.

[39] N. Roy and S. Thrun. Coastal navigation with mobile robots. In *Advances in Neural Information Processing Systems 12*, pages 1043–1049. The MIT Press, Apr. 1999.

[40] R. Sim and N. Roy. Global A-optimal robot exploration in SLAM. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 673–678, Barcelona, Apr. 2005.

[41] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Proc. 4th Int. Sym. Robot. Res.*, pages 467–474, Santa Clara, 1988.

[42] E.J. Sondik. The optimal control of partially observable Markov processes over the infinite horizon: discounted cost. *Op. Res.*, 26:282–304, 1978.

[43] M. T. J. Spaan and N. A. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *J. Artif. Intell. Res.*, 24:195–220, 2005.

[44] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 3310–3317, San Diego, May 1994.

[45] S. Thrun. Monte carlo POMDPs. In *Advances in Neural Information Processing Systems 12*, pages 1064–1070. The MIT Press, Apr. 1999.

[46] 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.*, 23(7-8):693–716, Jul. 2004.

[47] S. Thrun and M. Montemerlo. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *Int. J. Robot. Res.*, 25(5-6):403–429, 2006.

[48] 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. *IEEE Robot. Automat. Mag.*, 11(4):79–91, Dec. 2004.

[49] M. A. Whitty and J. Guivant. Efficient path planning in deformable maps. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 5401–5406, Saint Louis, Oct. 2009.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Innovation under projects DPI-2007-60858, DPI-2008-06022, and MIPRCV Consolider-Ingenio 2010; and by a PhD scholarship to R. Valencia from the Mexican Council of Science and Technology.

IRI reports

This report is in the series of IRI technical reports.
All IRI technical reports are available for download at the IRI website
`http://www.iri.upc.edu`.