Active Pose SLAM

Rafael Valencia, Jaime Valls Miró, Gamini Dissanayake and Juan Andrade-Cetto

Abstract— **We present an active exploration strategy that complements Pose SLAM [1] and optimal navigation in Pose SLAM [2]. The method evaluates the utility of exploratory and place revisiting sequences and chooses the one that minimizes overall map and path entropies. The technique considers trajectories of similar path length taking marginal pose uncertainties into account. An advantage of the proposed strategy with respect to competing approaches is that to evaluate information gain over the map, only a very coarse prior map estimate needs to be computed. Its coarseness is independent and does not jeopardize the Pose SLAM estimate. Moreover, a replanning scheme is devised to detect significant localization improvement during path execution. The approach is tested in simulations in a common publicly available dataset comparing favorably against frontier based exploration.**

I. INTRODUCTION

State of the art approaches to Simultaneous Localization and Mapping (SLAM) can now efficiently manage thousands of landmarks [3]–[6]. Most of these techniques are passive in the sense that the robot only estimates the model of the environment, but without taking any decisions on its trajectory. An active technique on the contrary, would also compute the appropriate robot actions to reduce the uncertainty about its own localization and the map, while at the same time optimizing coverage [7, 8].

A straightforward solution is to combine a classical exploration method with a SLAM technique. However, classical exploration methods focus on reducing the amount of unseen area disregarding the cumulative effect of localization drift, leading the robot to accumulate more and more uncertainty. Thus, a solution to this problem should revisit known areas from time to time, trading off coverage with accuracy.

Although action selection is the central issue in exploration for SLAM, there are also other issues that need to be considered. For instance, we need to choose a SLAM method, an environment representation, a coverage strategy, and an objective function. Each one imposes different challenges. Besides the well-known challenges in the SLAM

R. Valencia and J. Andrade-Cetto are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens Artigas 4-6, Barcelona 08028, Spain `rvalencia,cetto@iri.upc.edu`

J. Valls Miró and G. Dissanayake are with the Faculty of Engineering and IT, University of Technology Sydney (UTS), NSW 2007, Australia `jaime.vallsmiro,gamini.dissanayake@cas.edu.au`

problem (e.g. scalability, consistency, data association, robot perception), the selection of the adequate objective function is determinant to define the quality of the map as well as the strategy to cover efficiently the environment. Regarding the action selection, a key challenge is to trade off between optimality and efficiency.

We tackle the exploration problem for the case of Pose SLAM [1], a delayed-state SLAM algorithm where only the robot trajectory is estimated and where landmarks are only used to produce relative constraints between robot poses. The map in Pose SLAM only contains the trajectory of the robot formed by collision free configurations. This map is also perfectly suited to be used as a belief roadmap [9] and can be directly employed to plan optimal paths that take into account the localization uncertainty along the trajectory [2]. In this work, we automate this roadmap construction from scratch by selecting the appropriate actions to drive the robot so as to maximize coverage and at the same time minimize localization and map uncertainties.

To guarantee coverage, an occupancy grid of the environment is maintained. A significant advantage of the approach is that this grid is only computed to hypothesize entropy reduction of candidate map posteriors, and that it can be computed at a very coarse resolution since it is not used to maintain neither the robot localization estimate, nor the structure of the environment. In a similar way to [10], the technique evaluates two types of actions: exploratory actions and place revisiting actions. Action decisions are made based on entropy reduction estimates. By maintaining a Pose SLAM estimate at run time, the technique allows to replan trajectories online should significant change in the Pose SLAM estimate be detected, something that would make the computed entropy reduction estimates obsolete.

In Section II we relate the method to the state of the art in exploration for SLAM. Section III briefly summarizes Pose SLAM. Next, the set of actions is described in Section IV, and the computation of their utility is described in Section V. Replanning is covered in Section VI, and Section VII describes a set of experiments that validate the strategy. Finally, Section VIII gives some concluding remarks.

II. RELATED WORK

Exploration strategies driven by uncertainty reduction date back to the seminal work of Whaite [11] for the acquisition of 3-D models of objects from range data. Within the context of SLAM, it is the work of Feder et al. [12], who first proposed a metric to evaluate uncertainty reduction as the sum of the independent robot and landmark entropies with an exploration horizon of one step to autonomously produce

occupancy maps. Bourgault et al. [13] alternatively proposed a utility function for exploration that trades off the potential reduction of vehicle localization uncertainty, measured as entropy over a feature-based map, and the information gained over an occupancy grid. In contrast to these approaches, which independently consider the reduction of vehicle and map entropies, Vidal et al., [8] tackled the issue of joint robot and map entropy reduction, taking into account robot and map cross correlations for the Visual SLAM EKF case.

Action selection in SLAM can also be approached as an optimization problem using receding horizon strategies [7, 14, 15]. Multi-step look ahead exploration in the context of SLAM makes sense only for situations in which the concatenation of prior estimates without measurement evidence remain accurate for large motion sequences. For highly unstructured scenarios and poor odometry models, this is hardly the case. In this work we opt for a one step look ahead action selection strategy, but that considers trajectories of variable sizes.

One technique that tackles the problem of exploration in SLAM as a one step look ahead entropy minimization problem makes use of Rao-Blackwellized particle filters [10]. The technique extends the classical frontier-based exploration method [16] to the full SLAM case. When using particle filters for exploration, only a very narrow action space can be evaluated due to the complexity in computing the expected information gain. The main bottleneck is the generation of the expected measurements that each action sequence would produce, which is generated by a ray-casting operation in the map of each particle. In contrast, measurement predictions in a Pose SLAM implementation, such as ours, can be computed much faster, having only one map posterior per action to evaluate, instead of the many that a particle filter requires. Moreover, in [10], the cost of choosing a given action is subtracted from the expected information gain with a user selected weighting factor. In our approach, the cost of long action sequences is taken into consideration during the selection of goal candidates, using the same information metrics that help us keep the robot localized during path execution.

III. POSE SLAM

In Pose SLAM [1] a probabilistic estimate of the robot pose history is maintained as a sparse graph, 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 [17]. Graph links indicate geometric relative constraints between robot poses, and the density of the graph is rigorously controlled using information measures.

In Pose SLAM, state transitions result from the composition of motion commands u_k to previous poses,

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

Registration of sensory data also produces relative motion constraints, but now between non-consecutive poses.

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

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. These links enforce graph connectivity, or loop closure in SLAM parlance, and revise the entire path state estimate, reducing overall uncertainty.

To enforce sparseness in Pose SLAM, only the non redundant poses and the highly informative links are added to the graph. A new pose is considered redundant when it is too close to another pose already in the trajectory and not much information is gained by linking this new pose to the map. However, if the new pose allows to establish an informative link, both the link and the pose are added to the map.

The information gain of a link, i.e., the difference in entropies on the entire map before and after the link is established, takes the form

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

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

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

$\mathbf{H}_i$, $\mathbf{H}_k$ are the Jacobians of h with respect to poses i and k evaluated at the state means μ_i and μ_k, $\boldsymbol{\Sigma}_{ii}$ is the marginal covariance of pose i, and $\boldsymbol{\Sigma}_{ik}$ is the cross correlation between poses i and k. Links that provide information above a threshold γ are added to the graph, either to link previous states, or to connect a new pose with the map prior.

Thus, all decisions to update the graph, either by adding more nodes or by closing loops, are taken in terms of the overall information gain.

IV. ACTION SET

Evaluating the effect of potential actions in the context of SLAM is expensive since the posterior must be evaluated for each candidate. Thus, to allow scalability, the exploration strategy should consider only a limited set of actions. In [10] for instance, actions are limited to long trajectories of two types: exploratory sequences and place re-visiting sequences. We take a similar approach and show how to compute these actions in the context of Pose SLAM.

A. Exploratory actions

Exploratory actions are computed by classical frontier-based exploration [16] and are intended to maximize coverage, that is, to drive the robot towards unknown regions. As in our case we do not have access to an explicit metric representation of the obstacles we have to generate it. However, the necessary data is implicitly encoded in the Pose SLAM posterior. Moreover, as this metric representation is not needed for the estimation process (i.e. the SLAM
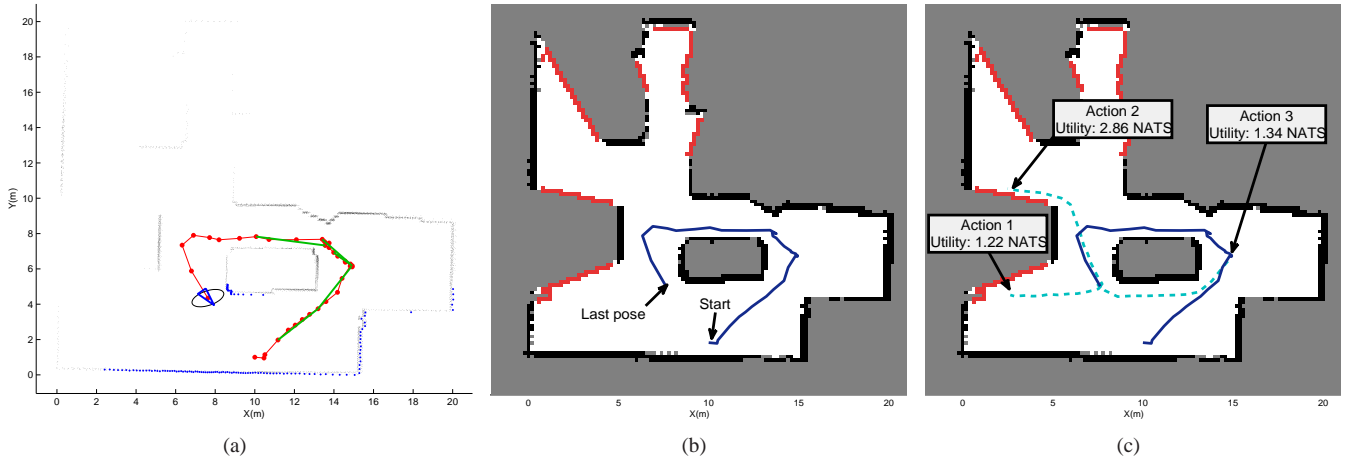
Fig. 1. The Pose SLAM posterior is used to render an occupancy map, which is used to generate candidate paths. (a) Pose SLAM map. (b) Gridmap and frontiers (red cells). (c) Candidate paths and their utilities.

process), as long as we can find frontiers and plan a path to those frontiers we are free to build this map as coarse as possible.

In the implementation reported here, the Pose SLAM algorithm stores the laser scans corresponding to each of the nodes in the pose graph. It is possible to use these scans to render an occupancy grid [18] for goal selection. Our premise of maximum likelihood navigation suggests that the map will not significantly change during path traversal, but only upon completing a path or should replanning be triggered due to large map shift at loop closure. This situation prevents us from online map rendering, since its computation is only needed at those events in time. Fig. 1(a) shows a Pose SLAM map. Frame (b) and (c) in the same figure show the rendered occupancy probability $p(m)$ for the grid map m at the mean prior, with a resolution of 20×20 cm.

Once we have computed the occupancy grid map, we extract frontiers and plan a path from the last robot pose to reach them. Following [16], frontiers are the set of free cells that have at least one neighboring unknown cell. Once we have identified the set of frontier cells we apply connected component labeling to detect connected regions of frontier cells. Then, for each frontier region of size larger than a threshold, we obtain its center of mass and compute a path from the last robot pose to that point. Path planning was implemented by looking for the shortest path within a probabilistic roadmap (PRM) [19].

Moreover, the marginal pose posterior was hypothesized through the Pose SLAM engine for each of the simulated trajectories, and only those trajectories with an entropy measure below a given threshold were chosen as safe exploratory routes. This effectively limits the length of exploratory paths to a cumulative open loop uncertainty threshold. Fig. 1(b) shows in red all frontier regions. Of these, actions 1 and 2, frame (c), were considered safe to reach.

B. Place re-visiting actions

In contrast to exploratory actions, place re-visiting actions are intended to improve localization of the robot, which translates into a decrease in entropy. In [10] a topological map is built to search for loop closures. In our case, the Pose SLAM map readily provides this topological structure. The search for loop closure candidates in our case uses the very same mechanisms of data association introduced in Pose SLAM [1], and hence, takes explicitly into account the uncertainty in localization when computing the distance between poses. The objective is to select the set of poses close to a distance d from the current pose, and to choose from these the one that maximizes information gain for the entire network.

First, we compute a distribution of the squared distance in belief space from the current pose x_k to each other pose x_i in the map[1]

$$\mu_d = \|\mu_k - \mu_i\|^2, \tag{5}$$

$$\sigma_d^2 = \mathbf{H}_d \begin{bmatrix} \mathbf{\Sigma}_{ii} & \mathbf{\Sigma}_{ik} \\ \mathbf{\Sigma}_{ik}^\top & \mathbf{\Sigma}_{kk} \end{bmatrix} \mathbf{H}_d^\top. \tag{6}$$

We do not want to consider neither loop closures that make the robot return large paths nor those that connect only to nearby neighbors. The probability of pose x_i being at a squared distance d with a threshold v to pose x_k is

$$p_d = \int_{d-v}^{d+v} \mathcal{N}(\mu_d, \sigma_d^2) \,. \tag{7}$$

The parameter d sets the mean squared distance to consider and the parameter v sets the window search size. Small values indicate that we want to consider loops strictly at a squared distance d from the current location, whereas large values would be more permissive. This probability, for a Gaussian distribution is easily evaluated with the error

---

[1]With a slight abuse in notation, μ_i refers here only to the x and y components of μ_i, and $\mathbf{\Sigma}_{ii}$ to the marginal elements of $\mathbf{\Sigma}_{ii}$, leaving the orientation terms out. The Jacobian $\mathbf{H}_d$ is simply $2[(\mu_i - \mu_k)^\top, (\mu_k - \mu_i)^\top]$.

function (`errf`). If the probability of being within the range $(d - v, d + v)$ (in belief space) is above a given threshold, the pose is added to the set of loop closure candidates.

Next, from this set of loop closure candidates we select the one that provides the largest information gain, computed with Eq. 3. Continuing with the example in Fig. 1 (c), it is action 3 the loop closure sequence that fulfills this condition. In contrast to the active loop closing technique used in [10], the technique discussed here accounts for the uncertainty in the trajectory, and is therefore more robust to localization errors. Finally, a path to the loop closure candidate is computed using the same approach as with the exploration candidates.

V. UTILITY OF ACTIONS

Once we have computed a set of candidate paths, we need to calculate their utility and select the one with largest reward. Our utility function is the expected information gain, i.e, the decrease in entropy for the posterior.

Just as in [10], we can approximate the full entropy of the trajectory and map as the sum of the individual entropies. That is, the joint entropy of a trajectory $x = x_{1:N}$ and a map m, given a set of motion commands $u = u_{0:N-1}$ and a set of observations $z = z_{1:N}$ is

$$
\begin{aligned}
H(x, m | u, z) &= H(x | u, z) + \int_x p(x | u, z) H(m | x, u, z) dx \\
&\approx H(x | u, z) + H(m | u, z).
\end{aligned}
\tag{8}
$$

The entropy of the path in Pose SLAM, being a multivariate Gaussian, is given by

$$
H(x | u, z) = \ln((2\pi e)^{(n/2)} |\mathbf{\Sigma}|),
\tag{9}
$$

where n is the dimension of the whole state vector.

Unfortunately, the evaluation of Eq. 9 has a drawback. As noted in [20], the covariance might easily become ill defined, with full correlated confidence and one or more eigenvalues near 0. This happens for instance when two poses become fully correlated, shrinking the probability distribution along a linear combination of states, while no information is gained in other dimensions.

To overcome this situation, we approximate the entropy of the trajectory without taking into account the correlation between poses, but averaging instead over the individual marginals [10]

$$
H(x | u, z) \approx \frac{1}{N} \sum_{i=1}^{N} \ln((2\pi e)^{(n'/2)} |\mathbf{\Sigma}_{ii}|),
\tag{10}
$$

where n' is the dimension of the individual pose vector. Another option would be to use an a-optimal measure of information for the path such as the trace of $\mathbf{\Sigma}$ [20]. In our experiments we have experienced better results with the average form than the aggregated measure (trace) when used in combination with the map entropy as in Eq 8. The main reason is that the effect of path length is averaged in the first case. This is a reasonable choice since we have already settle for a range of path lengths as discussed in Sec. IV-B.

For a map m with cell size l, the entropy is computed with

$$
H(m | u, z) = -l^2 \sum_{c \in m} (p(c) \ln p(c) + (1 - p(c)) \ln(1 - p(c))).
\tag{11}
$$

To compute this posterior, we must hypothesize about unknown ray casting measurements. We take the same approach as in [10], where statistics about the change in entropy are computed as a function of the number of unclassified cells covered by a hypothetical laser beam. When an unknown cell is hit, its probability contribution to Eq. 11 is taken from this statistic.

Fortunately, and unlike with particle filters, we only have one map prior in which to simulate observations, instead of doing so for each map particle. Moreover, given that state estimation is not dependent on this map, it can be computed at a very coarse resolution, with the consequent advantages in computational cost. Another advantage of our approach in contrast to the particle filter implementation is that we do not need to arbitrarily weight the cost to reach the goal as this might bias the exploration behavior. Instead, the two techniques discussed in Section IV guarantee that all paths in the set are of similar length (either by thresholding on open loop uncertainty during exploration, or by searching for loop closures close to a distance d from the current pose). Nonetheless, high costs in path execution mean large probabilities of becoming lost. For this reason, we enforce a replanning strategy should unforeseen loop closure occur during path execution.

Given that all actions are evaluated departing from the same prior, selecting the action or path a that maximizes information gain is exactly the same as selecting the path that minimizes the entropy of the joint posterior (x', m') given the path u and upon traversing the hypothetical path a, and observing z and hypothesizing about ray casted unexplored cells z'

$$
a^* = \arg \min H(x', m' | u + a, z + z').
\tag{12}
$$

For the candidate paths and utilities shown in Figure 1(c), actions 1 and 2 are exploratory, whereas action 3 closes a loop. Action 1 only reduces uncertainty about the environment as it drives the vehicle to an unknown region. Action 3 only reduces path uncertainty bringing the robot back to a known location. Our action selection mechanism chooses path 2, which reduces uncertainty about the environment while keeping the robot well localized.

VI. REPLANNING

When planning long paths we might need to predict many observations ahead of time and, most likely, these predictions will differ substantially from the actual observations obtained when the action is executed. The most evident case is when the robot closes a large loop during path execution. The path and map estimates will change considerably and the predicted gain at the end of the path might not be relevant anymore, or even worse, the rest of the path candidate might be willing to drive the robot to a collision.
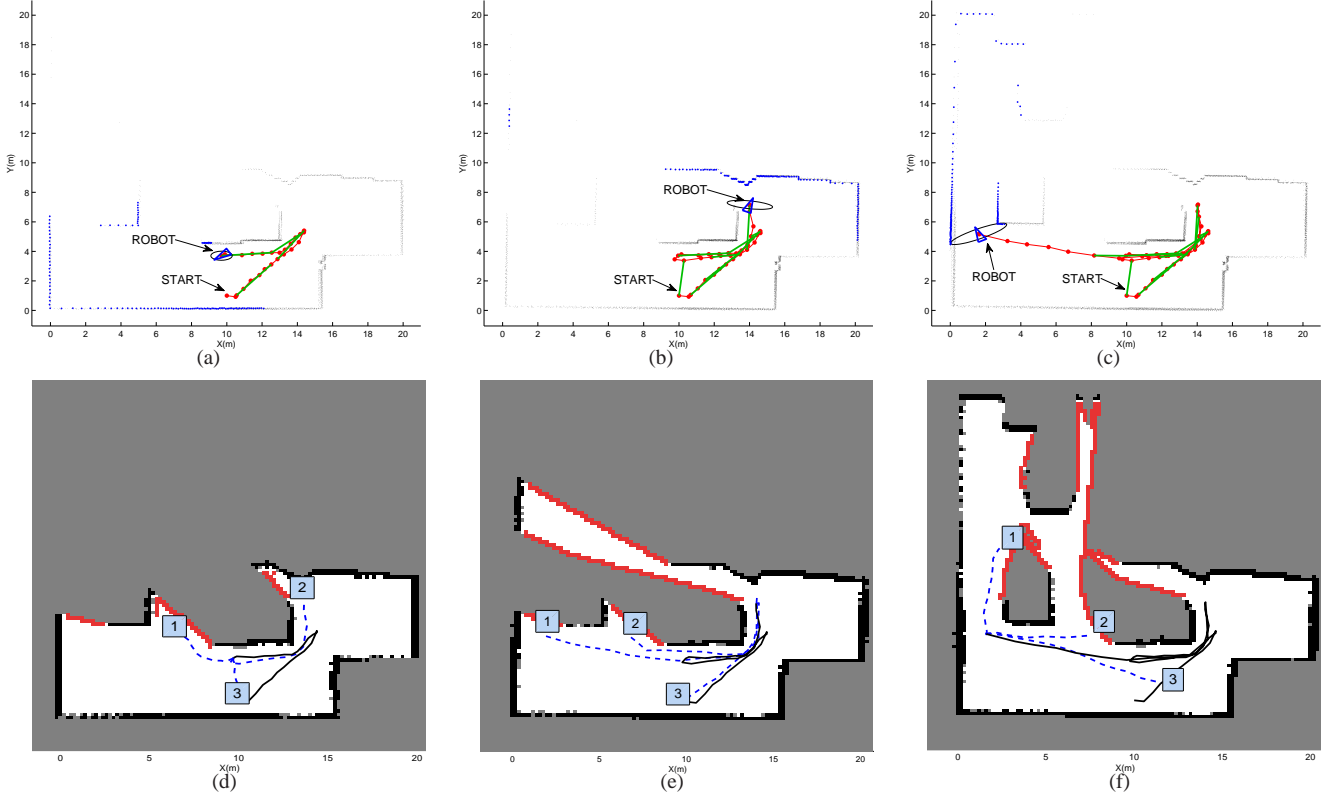
Fig. 2. Three points in time during the exploration process. At time step 26 (frames a and d), the robot has the following reduction in entropy: Action 1 = 1.1121 nats, Action 2 = 1.2378 nats, and Action 3 = 0.7111 nats. At time step 39 (frames b and e) Action 1 = 1.7534 nats, Action 2 = 1.4252 nats, and Action 3 = 1.1171 nats. Finally, at time step 52 (frames c and f), Action 1 = 1.8482 nats, Action 2 = 2.0334 nats, and Action 3 = 1.7042 nats. The actions chosen are 2, 1, and 2, respectively.

One clear alternative is to use a receding horizon to plan, but such continuous replanning is prohibitively expensive in computational terms, especially for large or finely grained maps. We opt to re-plan only when it is worth doing so. One way to know when it is wise to replan is by anticipating large deformations in the map. This occurs only if large information gains are fed to the pose network. Fortunately, these can be anticipated with the information gain in Eq. 3. That is, we replan if during path execution this value becomes large for any loop closure, making our predictions obsolete.

VII. Experiments

In order to evaluate the exploration strategy presented in this paper we simulated a robot exploring the widely used cave-like two-dimensional environment available from [21], scaled to a resolution of $20\,\text{m} \times 20\,\text{m}$. We present results of the evolution of the exploration method, the effects of replanning, and a comparison with frontier based exploration [16].

In the reported setting, robot motion was simulated with an odometric sensor with noise covariance $\Sigma_u = \text{diag}(0.1\,\text{m},0.1\,\text{m},0.0026\,\text{rad})^2$. Moreover, a laser range finder sensor was simulated to establish links between any two poses closer than $\pm 3\,\text{m}$ in x and y, and $\pm 0.52\,\text{rad}$ in orientation. Relative motion constraints were measured using the iterative closest point algorithm. Measurement noise covariance was fixed at $\Sigma_y =$ $\text{diag}(0.05\,\text{m},0.05\,\text{m},0.0017\,\text{rad})^2$. Laser scans were simulated by ray casting over a ground truth gridmap of the environment using the true robot path. The initial uncertainty of the robot pose was set to $\Sigma_0 = \text{diag}(0.1\,\text{m},\ 0.1\,\text{m},\ 0.09\,\text{rad})^2$. Nearby poses were detected with γ at $2.5\,\text{nats}$.

A. Exploration

The algorithm was executed with the aforementioned conditions and the effects of the exploration strategy were recorded. Fig. 2 shows the obtained maps at three points in time. At each iteration, using the Pose SLAM prior (top row), a gridmap is rendered (bottom row) and used to compute the next exploration path. For instance, at time step 26 (frames a and d), the algorithm chooses Action 2, leading the robot to explore a region to reduce map entropy. Then, at time step 39, the PRM planner does not find a path to the nearest frontier. The free cells to reach it form a narrow hallway which cannot be safely traversed. Instead, the path planner selects another frontier. Eventually, the algorithm chooses Action 1 because along this path the robot observes more unknown cells with the consequent larger reduction in map entropy. Finally, at time step 52, the more conservative Action 2 is selected this time since it reduces both the path and map entropies. Fig. 3 shows the path and map entropy evolution for the execution of the entire exploration sequence.
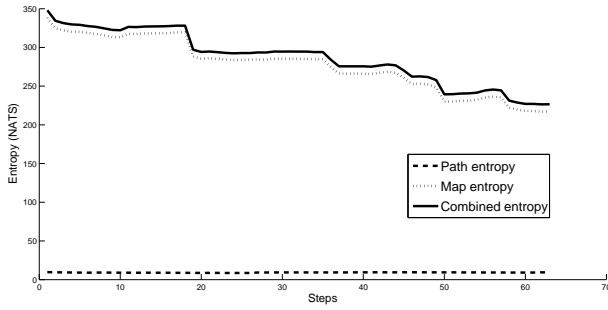
Fig. 3. Entropy evolution.

B. Replanning

The exploration strategy can be improved with a re-planning scheme. Replanning is triggered when we detect significant change between the entire Pose SLAM prior and posterior upon loop closure. It is an indicator of significant shift in the map estimate and the scheme is devised to antici-pate those changes. In the reported experiments, replanning is triggered upon loop closure with information content greater than 4 nats.

Figure 4 shows a comparison of the exploration results with and without replanning. A slight drop in map entropy is observed when replanning is considered, from 147.89 nats to 146.23 nats for experimental runs of the same duration of 180 time steps. While the changes in the final map are subtle (some different regions covered, and slight improvement of entropy reduction), the changes in the localization estimates are more evident. Fig. 5 shows the overall path entropy evolution during the entire duration of the experiment. We have noticed that the replanning strategy not only helps reduce overall map uncertainty, but also enforces better robot localization, maintaining full path entropy bounded to about 9.5 nats. The figure also shows how without replanning, the exploration strategy eagerly seeks path uncertainty reduction by finalizing loop closure paths to their end even when a loop closure has already been asserted prior to their completion (first 20 time steps) paying this greed in localization soon after.

C. Comparison with frontier-based exploration

Next, we compare our method against pure frontier-based exploration using the same environment and specifications employed in the aforementioned experiments. Frontier-based exploration always drives the robot to the closest frontier disregarding uncertainty in the map and its localization. In our implementation analyzed frontiers are limited to a size larger than 9 cells. See Fig. 6. One can note that, although this greedy scheme eventually covers all the environment, the resulting map and path contain severe localization errors, as the robot barely closes three loops, which are not enough to correct the drift, causing it to end up with a final map entropy of 152.62 nats. In contrast, the Active Pose SLAM approach presented in this paper also covers the whole environment in the same number of time steps, yielding a slightly lower
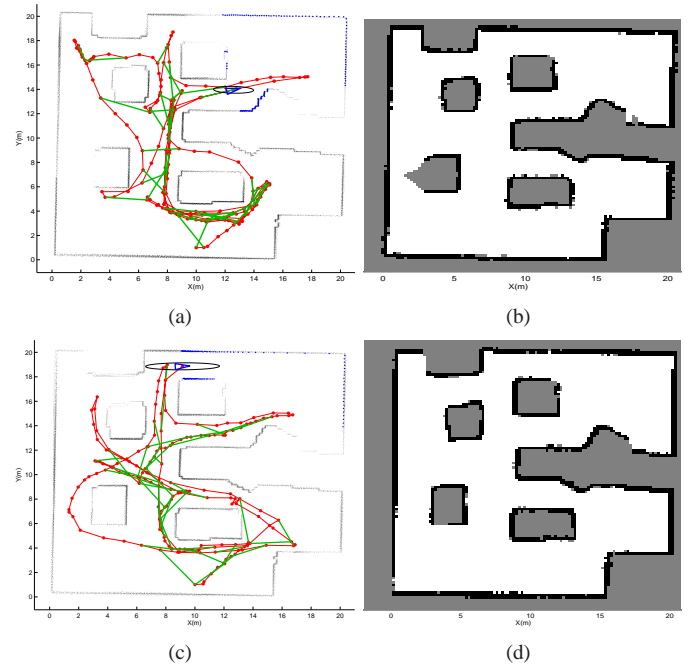

Fig. 4. Exploration with and without replanning. (a) Pose SLAM map and (b) gridmap made without replanning, with a final map entropy of 147.89 nats. (c) Pose SLAM map and (d) gridmap made with replanning, with a final map entropy of 146.23 nats.
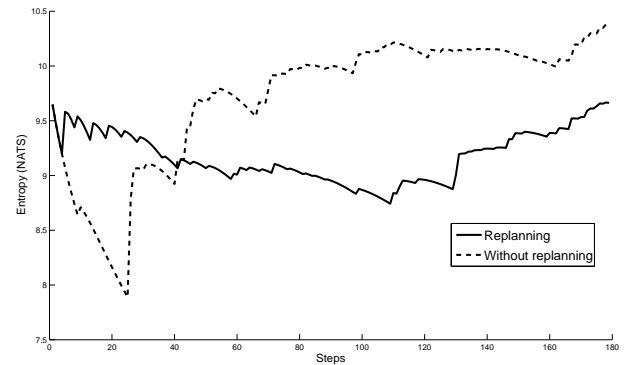

Fig. 5. Path entropy evolution with replanning (continuous line) and without replanning (dashed line).

final map entropy of 146.23 nats for the same experimental setting (see Fig. 4c-d), thus better satisfying the competing objectives of coverage and accuracy.

VIII. CONCLUSION

In this paper we presented an active exploration strategy tightly integrated with Pose SLAM. The work is inspired in the action selection mechanisms reported in [10]. The approach needs only to compute map entropy at the Pose SLAM mean instead of at each map particle. Furthermore, the resolution of the gridmap used is independent of the Pose SLAM estimate, and it can be as coarse as needed. These two issues allow efficient computation of the information gain objective function used to evaluate candidate exploration paths, with the end result of a scalable solution to the
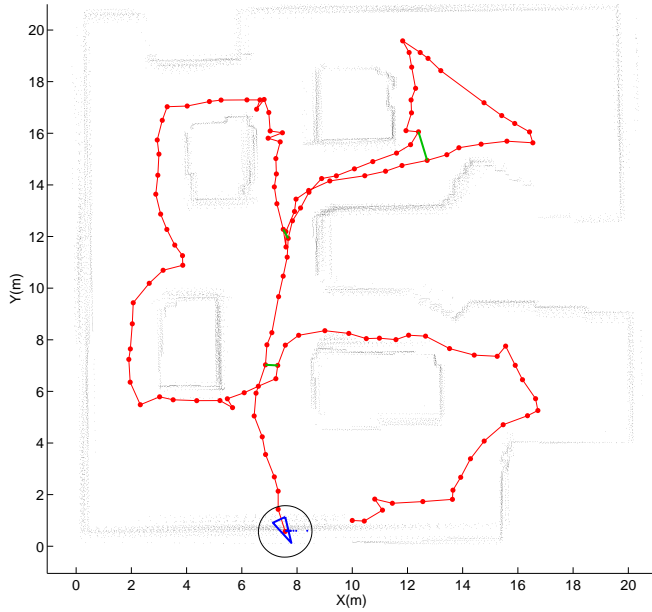
Fig. 6. Frontier-based exploration. The final map entropy is only reduced to 152.62 nats. Contrary to the proposed approach, this technique does not evaluate the need for loop closure and as a consequence, severe localization errors are evident at the end of the trajectory.

problem.

The mechanism to evaluate place revisiting actions is simple and is tightly integrated within Pose SLAM. The selection of loop closure candidates takes into account path uncertainty and benefits from the marginal estimates maintained within Pose SLAM. The exploration strategy detects significant changes in the state estimate to interrupt the execution of large loop closure trajectories and triggers replanning. The end result is improved map and localization entropy reduction.

In the same way that replanning can be triggered upon unexpected loop closing with the consequent reduction of path uncertainty above 4 nats, it could be possible to trigger replanning upon unexpected significant improvement of map entropy before completing an exploratory trajectory. Take for instance frame e in Fig 2. Passing near point 2 in the path to point 1 might increase map coverage significantly above than the current map prior, and hence, continuing exploration towards point 1 might not be the right thing to do, especially since odometric error accumulates during open loop traverse. To account for this, we need a way to evaluate overall map entropy at a higher frame rate, perhaps by only measuring information gain over the cells that become covered during path execution. The fact that entropy reduction in the path can be computed online and at high frame rate is thanks to the use of Pose SLAM as the estimation workhorse, but unfortunately, in Pose SLAM, the map posterior is marginalized out and needs to be computed to evaluate exploration candidates. Doing so at a high frame rate is left as future work.

REFERENCES

[1] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact Pose SLAM," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 78–93, 2010.

[2] R. Valencia, J. Andrade-Cetto, and J. M. Porta, "Path planning in belief space with Pose SLAM," in *ICRA*, Shanghai, 2011, pp. 78–83.

[3] K. Konolige, "Large-scale map-making," in *AAAI*, San Jose, 2004, pp. 457–463.

[4] 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, 2004.

[5] U. Frese, "Treemap: An O(log n) algorithm for indoor simultaneous localization and mapping," *Auton. Robot.*, vol. 21, no. 2, pp. 103–122, 2006.

[6] R. Valencia, E. H. Teniente, E. Trulls, and J. Andrade-Cetto, "3D mapping for urban service robots," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009.

[7] C. Leung, S. Huang, N. Kwok, and G. Dissanayake, "Planning under uncertainty using model predictive control for information gathering," *Robot. Auton. Syst.*, vol. 54, no. 11, pp. 898 – 910, 2006.

[8] T. Vidal-Calleja, A. Sanfeliu, and J. Andrade-Cetto, "Action selection for single camera SLAM," *IEEE Trans. Syst., Man, Cybern. B*, vol. 40, no. 6, pp. 1567–1581, 2010.

[9] 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.

[10] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters." in *RSS*, Cambridge, 2005.

[11] P. Whaite and F. P. Frank, "Autonomous exploration: Driven by uncertainty," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 3, pp. 193–205, 1997.

[12] H. Feder, J. J. Leonard, and C. Smith, "Adaptive mobile robot navigation and mapping," *Int. J. Robot. Res.*, vol. 18, pp. 650–668, 1999.

[13] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *IROS*, Lausanne, 2002, pp. 540–545.

[14] S. Huang, N. M. Kwok, G. Dissanayake, Q. P. Ha, and G. Fang;, "Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity," in *ICRA*, Barcelona, 2005, pp. 1091 – 1096.

[15] C. Leung, S. Huang, and G. Dissanayake, "Active SLAM for structured environments," in *ICRA*, Pasadena, 2008, pp. 1898–1903.

[16] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *CIRA*, Washington, 1997, pp. 146–151.

[17] 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, 2006.

[18] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *ICRA*, St. Louis, 1985, pp. 116–121.

[19] L. Kavraki, 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.

[20] R. Sim and N. Roy, "Global A-optimal robot exploration in SLAM," in *ICRA*, Barcelona, 2005, pp. 673–678.

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