Potential information fields for mobile robot exploration

Joan Vallvé, Juan Andrade-Cetto

Institut de Robòtica i Informàtica Industrial, CSIC-UPC
Llorens Artigas 4-6, 08028 Barcelona, Spain.

Abstract

We present a decision theoretic approach to mobile robot exploration. The method evaluates the reduction of joint path and map entropy and computes a potential information field in robot configuration space using these joint entropy reduction estimates. The exploration trajectory is computed descending on the gradient of these field. The technique uses Pose SLAM as its estimation backbone. Very efficient kernel convolution mechanisms are used to evaluate entropy reduction for each sensor ray, and for each possible robot orientation, taking frontiers and obstacles into account. In the end, the computation of this field on the entire configuration space is shown to be very efficient computationally. The approach is tested in simulations in a pair of publicly available datasets comparing favorably both in quality of estimates and in execution time against an RRT*-based search for the nearest frontier and also against a locally optimal exploration strategy.

Keywords: Mobile robotics, SLAM, Exploration

1. Introduction

We consider the problem of autonomous mobile robot exploration, and frame it as that of reducing both localization and map uncertainties. Exploration strategies driven by uncertainty reduction date back to the seminal work of Whaite [21] 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. [4], 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. [1] 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 consider independently the reduction of vehicle and map entropies, Vidal-Calleja et al., [20] 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 [6, 11, 10]. 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. So, we stick with the one step look ahead case.

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 [15]. The technique extends the classical frontier-based exploration method [22] 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 [15], 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.

In [17] our group proposed a solution to the exploration problem that maximizes information gain in both the map and path estimates. The method evaluates both exploratory and loop closure candidate trajectories, computing entropy reduction estimates from a coarse resolution realization of occupancy maps. The final trajectory is computed using A* in the occupancy grid, just as [9] does so over an initial reference trajectory. The computational bottleneck of [17] was in the estimation of the occupancy map. In this paper we present an alternative method, in which we compute directly the global entropy reduction estimate for each possible robot configuration. The use of very efficient kernel convolutions allow us to compute this estimate very fast and without the need to reduce the grid resolution. In [17], exploratory actions considered omnidirectional sensing and evaluated paths toward positions near frontiers, disregarding orientation. In a more general setting, a sensor, such as a laser range finder or a camera, would have a narrow field of view, and hence, we need to deal with full poses not just positions. In this paper we take this issue into account and compute instead entropy reduction estimates for the whole configuration space (C-space).

To find candidate exploration paths, the entropy reduction grid in C-space is transformed into a potential field, taking into account frontiers and obstacles. The path is obtained by gradient descent on this field. Potential field methods

have been previously used for exploration [2, 12], but different than our approach, these methods directly evaluate boundary conditions on deterministic maps of obstacles and frontiers, without taking uncertainty into account. Our method follows the idea of gradient descent to a desired exploratory or loop closing location, due to the minimization of joint map and path entropies.

In summary, the proposed method iteratively proceeds as follows. First, from the current Pose SLAM estimate (Sec. 2), a log odds occupancy map is synthesized from raw sensor data as shown in Sec. 3. We use this map to compute a potential information field (Sec. 4), and plan exploration trajectories as gradient descent along this field. Once the current exploration goal is reached, or a loop closure is obtained, a new exploration candidate is computed in the next iteration. The method is compared against frontier-based exploration and locally optimal planning in Sec. 5, and conclusions are drawn in Sec. 6.

2. Pose SLAM

The proposed exploration strategy uses Pose SLAM as its estimation backbone. In Pose SLAM [7], 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 has the advantage of being exactly sparse [3]. State transitions result from the composition of motion commands to previous poses,

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

and the registration of sensory data also produces relative motion constraints, but now between non-consecutive poses,

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

where the operators $\oplus$ and $\ominus$ are used to indicate forward and backward composition of one coordinate frame relative to another [14].

When establishing a link between the current robot pose, say k, and any other previous pose, say i, the update operation only modifies the corresponding diagonal blocks 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, hence entropy.

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

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

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

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

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

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

3. Log Odds Occupancy Grid

Pose SLAM does not maintain a grid representation of the environment. It only encodes relations about robot poses. The environment however, can be synthesized at any instance in time using the pose means in the graph and the raw sensor data. The resolution at which the map is synthesized depends on the foreseen use of this map. For instance, in [17] occupancy grid maps at very coarse resolution are produced to evaluate the effect of candidate trajectories in entropy reduction. In some cases one might not even need to render a map. Such is the case in [16], where only the graph is needed to plan optimal trajectories in a belief roadmap.

In this paper, we use the Pose SLAM estimate and raw sensor data to synthesize an occupancy map, and from this map, we then build an entropy reduction field in configuration space. The quality of the occupancy grid produced is a key element of our exploration strategy. The mapping of frontiers near obstacles in the presence of uncertainty might drive the robot to areas near collision, a situation we need to avoid. Moreover, there is a compromise between tractability and accuracy in choosing the resolution at which the occupancy cells are discretized.

To provide an accurate computation of the occupancy map, which is necessary for the proper computation of the potential information field, we render the map from all poses in the Pose SLAM graph, and not only a limited number of them. Moreover, the resolution at which the occupancy grid map is computed is finer than what we were able to compute in [17]. Instead of repeating the ray-casting operation at each iteration, we store local log odds occupancy maps at each robot pose, and aggregate them efficiently for the computation of a global log odds occupancy map.

At each robot pose x_k, the raw sensor data is ray-casted to accumulate evidence $p_k(c)$ for each cell c in a log odds occupancy grid in local coordinates

$$m_k(c) = \log \frac{p_k(c)}{1 - p_k(c)} \ . \tag{5}$$

4

| (a) m_1 | (b) m_2 | (c) m_3 | (d) m_4 |

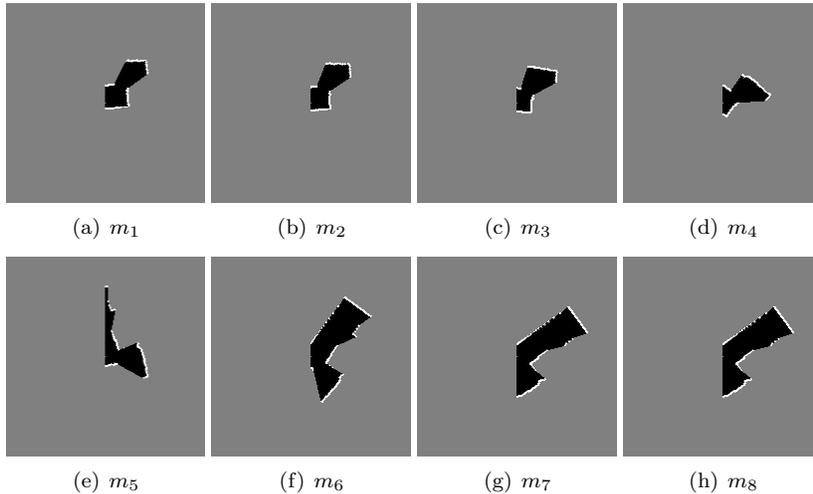| (e) m_5 | (f) m_6 | (g) m_7 | (h) m_8 |

Figure 1: A number of log odds occupancy maps in local coordinates.

These local log odds occupancy maps are shown in Fig. 1 for a small number of robot poses. Negative values (black) mean free space, and positive values (white) mean obstacles. A value of zero (gray) means unexplored. During open loop, each local map is aggregated into the global log odds occupancy map m. To relate them in a common reference frame, each local map is rotated via sheers and translated using very efficient image processing routines. Only at loop closure, the occupancy map is recomputed from scratch using all previously stored local log odds maps but oriented and translated according to the newly estimated robot poses. The result is shown in Fig. 2a.

Generalizing Eq. 5 to the global coordinate case, we can solve directly for aggregated cell classification probabilities,

$$p(c) = \frac{e^{m(c)}}{1 + e^{m(c)}} \; . \tag{6}$$

Note however that map aggregation was computed only at the mean pose estimates. To smooth out misclassified and unobserved cells and to classify free cells, obstacles and frontier cells (unobserved close to a free cell) morphological opening and closing operations on the global log odds map are used. The resulting detection of frontiers, obstacles and free cells is exemplified in Fig. 2b.

4. Exploration with Potential Information Fields

Our purpose at each exploratory step is to find a path that drives the robot to those locations in the map that reduce the uncertainty in classification of free and occupied cells. That is, as in [17], to drive the robot to minimize the joint robot path and map entropies.
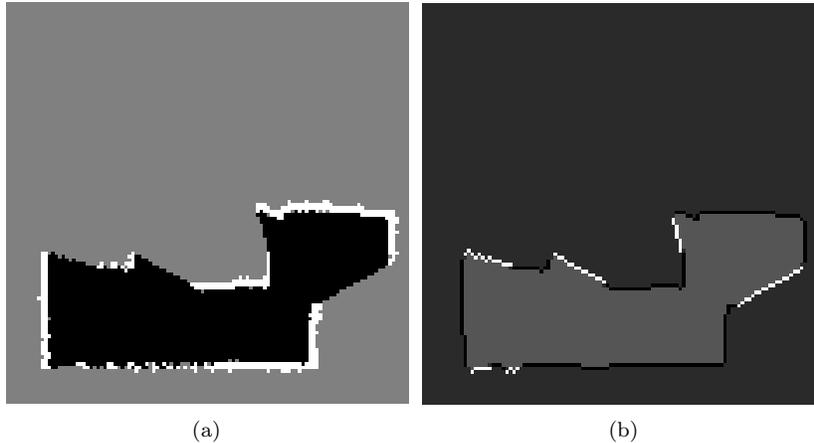
5

Figure 2: (a) Aggregated log odds occupancy map m. (b) Classified frontier cells (white), obstacle cells (black), free cells (light grey) and unobserved cells (dark grey).

The objective is to find a scalar function $\phi(c)$ defined over all C-space cells such that its gradient $\nabla\phi$ will consist of a path with largest joint path and map entropy decrease. Unfortunately, the entropy decrease when executing a trajectory from the current pose to any given C-space configuration is not independent of the path taken to arrive to such pose. Different routes induce different decrease values of path and map entropies. Take for instance two different routes to the same pose, one that goes close to previously visited locations or one that discovers unexplored areas. In the former, the robot would be able to close loops, and thus maintain bounded localization uncertainty. In the second, an exploratory route would reduce the map entropy instead.

Computing the optimal path to a goal taking into account the effect on the reduction of joint path and map entropies for each possible route is a computationally intractable process for anything else than very small academic scenarios. We are content with obtaining a suboptimal solution, assuming that the robot can reach each possible configuration in one single step and letting Pose SLAM take into account uncertainty reduction during path execution. That is, we do consider path and map entropy changes thru the path to each robot configuration but only the changes induced in entropy by 'appearing' at that configuration.

This is a strong assumption since we give up knowing about the change in entropy at intermediate steps for the ability to compute a dense entropy estimate over the whole C-space. This dense estimate ends up being conservative, but this is not a problem since it is only used for path planning and not to guarantee path execution. Any event triggering replaning (reaching a goal or a collision) triggers also a new evaluation of the entropy decrease estimate for the whole C-space. Contrary, in Active Pose SLAM [17], entropy reduction is indeed
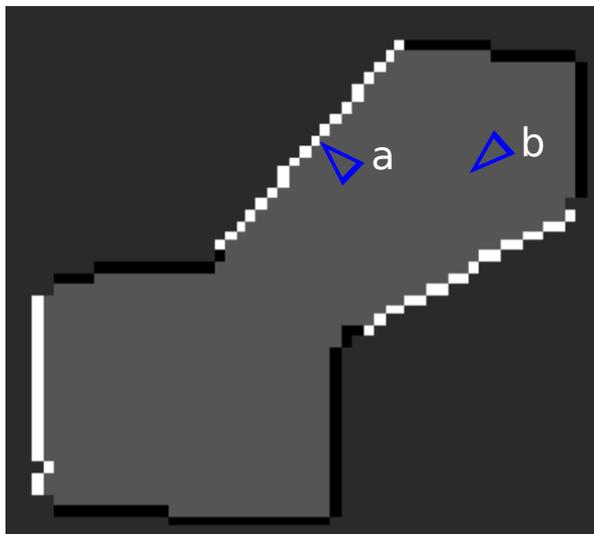
6

Figure 3: The choice of pose a, a configuration perpendicular to a frontier, as an exploratory goal is a suboptimal choice, compared to the selection of pose b, a configuration of optimal map entropy reduction.

simulated for every step along the path to the goal. This makes Active Pose SLAM computationally intractable for large planning horizons.

In contrast to our approach, [12] defines a potential scalar function using attraction and repulsion fields on frontiers and on the current robot pose, with some boundary conditions on obstacles. Choosing frontiers as attractors poses some challenges. Frontiers are unexplored areas next to free cells which have a significant probability of being yet unseen obstacles. The use of potential fields to reach frontiers produces perpendicular robot configurations at the arriving locations, thus making the robot face these new obstacles directly, with the consequent unavoidable collision. Other methods that select frontiers as goal locations during exploration that are not based on potential fields share the same inconvenience [23]. We instead set as attractors not the frontiers, but the robot configurations at which joint entropy reduction is maximized. These poses are not necessarily close to frontiers, but can be at any configuration in the free space. In addition, these attractors will also guarantee larger reductions in map entropy since more frontier cells can be observed from these locations than from the frontier itself. See Fig. 3.

The general formula of the joint map and path entropy given all motions and observations, would be the addition of the path entropy given all motions and observations and an average of the entropy of all infinite maps resulting from all infinite paths x of the probability distribution weighted by the probability

of each possible trajectory:

$$H(x, m|u, z) = H(x|u, z) + \int_x p(x|u, z)H(m|x, u, z)dx. \tag{7}$$

The joint state entropy is approximated, in a similar way as in [17] as

$$H(x, m|u, z) \approx H(x|u, z) + \alpha(p(x|u, z))H(m|\mu_x, u, z) \tag{8}$$

where instead of computing the map entropy averaging for all infinite possible maps, we compute it only for the mean pose estimates μ_x and differently to the approximation used in [17], we add a factor $\alpha(p(x|u, z))$ multiplying the map entropy depending on the probability distribution. Finally, we actually do not want the absolute value of the joint entropy but its change, so what we need to compute is

$$\Delta H(x, m|u, z) \approx \Delta H(x|u, z) + \alpha(p(x|u, z))\Delta H(m|\mu_x, u, z). \tag{9}$$

We evaluate joint entropy reduction on these two terms separately for each discretized robot configuration in C-space, treat this entropy reduction as an information field and smooth it to avoid discontinuities. We finally compute the exploration path as the gradient of this field.

4.1. Path entropy reduction

The first term in Eq. 8 accounts for the path entropy, which in Pose SLAM is given by

$$H(x|u, z) = \ln((2\pi e)^{(n/2)}|\mathbf{\Sigma}|). \tag{10}$$

The evaluation of Eq. 10 poses some drawbacks. As noted in [13], it can easily become ill defined. To overcome this situation one might approximate its value without taking into account the correlation between poses, and averaging over the individual pose marginals as proposed by Stachniss et al. [15] and also implemented in [17].

This is not necessary in our case, since we are not interested in computing the entropy itself, but its change –the information gain– (eq. 9) and not for just one posterior pose, but for the whole discretized C-space. To approximate it, we assume a noise free platform for the evaluation of the final leg in the path, and thus the jump from the current pose to each configuration will produce the same marginal posterior, with zero information gain, except at loop closure.

And, in closing a loop between any previous configuration i and the current one k, the decrease in path entropy is given precisely by the information gain $\mathcal{I}_{ik}$ encoded in a link connecting the two nodes as defined in Eq. 3,

$$\Delta H(x|u, z) = \begin{cases} \mathcal{I}_{ik} & \text{if a loop with configuration } i \text{ can be closed,} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

To establish such a link, the two configurations must be within the sensor range, i.e., inside the sensor match area. Instead of iterating over each cell in

the C-space grid and searching for its loop closure candidates in the Pose SLAM graph, the iteration proceeds the other way. For each pose in the Pose SLAM graph, we annotate the cells inside their match area in a C-space path entropy decrease grid with the corresponding information gain. The resulting C-space grid contains the amount of path entropy decrease when the robot is moved to that particular position and orientation.

4.2. Map entropy reduction

In contrast to [17], in which we compute the reduction in entropy for a limited set of final configurations, we now compute it for each configuration in the discretized C-space. For a map with size cell l, its entropy can be computed as a scalar value

$$H(m|u, z) = -l^2 \sum_{\forall c \in m} (p(c) \log p(c) + (1 - p(c)) \log(1 - p(c))). \qquad (12)$$

The reduction in entropy that is attained after moving to a new location and sensing new data depends basically on the number of cells that will change its status from unknown to discovered, either obstacle or free. Anticipating the number of discovered free cells before actually processing the new observations is impossible. We are content with approximating entropy reduction as the increase in the number of discovered frontier cells.

This map entropy reduction by observing frontier cells for each robot configuration could be found computing the frontier visibility of each sensor ray for each robot configurations but this wouldn't be efficient at all. We propose instead a novel method for computing it efficiently using convolutions. Taking into account that several robot orientations have common ray directions, and also that using a simple convolution, we can compute the frontier visibility for a specific direction from all positions, we invert the order of computations.

Hence, we are able to compute this entropy change very efficiently with the following three steps:

1. *Obstacle occlusion mask.* We generate a three-dimensional grid. Its dimensions are x, y, and the orientation of each laser ray. For each ray orientation layer, a 2D obstacle occlusion mask is created, annotating whether the nearest non-free cell along that ray direction is a frontier or not. The mask is computed with a one-dimensional convolution with an inverse exponential motion kernel over a positive value for frontier cells and a negative value for obstacles. Binary thresholding the positive values we obtain the desired occlusion mask. See Fig. 4b.
2. *Frontier convolution.* Given the radial nature of the sensor being simulated, each frontier cell will receive a different density of ray casts from the same scan, thus it is necessary to compensate for this in order not to overestimate the number of frontier cells being observed. The ray cast density $n(c)$ at each cell c is modeled as a function of the distance from the robot to that cell $r(c)$ and the angle β between two consecutive sensor rays

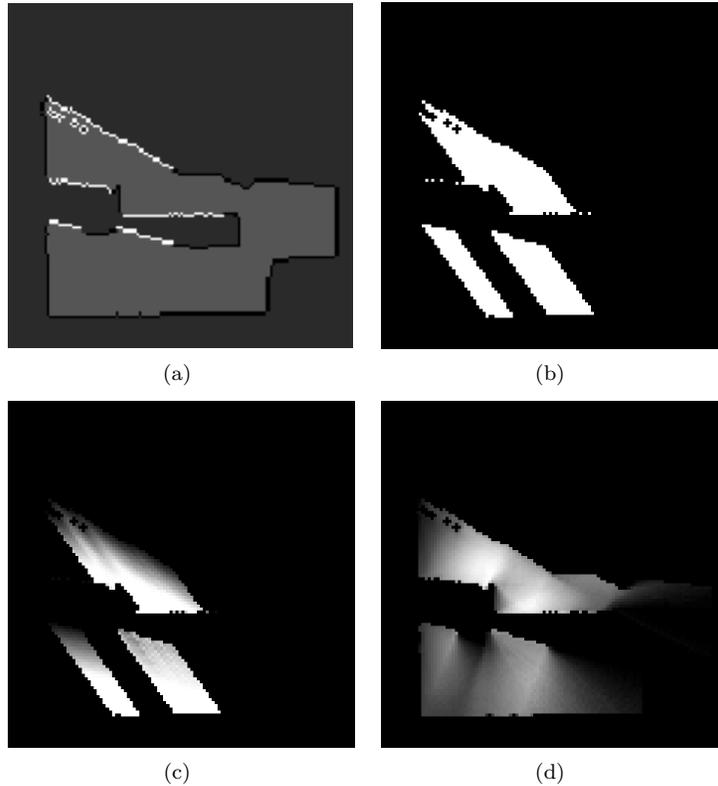$$n(c) = \frac{1}{r(c) \, \tan \beta} \, . \qquad (13)$$

9

Figure 4: Computation of entropy change. (a) Occupancy map with obstacles (black), frontiers (white) and free cells (light gray). (b) Obstacle occlusion mask in one ray direction. (c) Map entropy decrease of one ray direction after the frontier convolution. (d) Sum over entire sensor spread for one robot orientation.

For each ray orientation layer, a convolution is made with a one-dimensional motion kernel weighted with $\min(1, r(c) \ \tan \beta)$. The result of this frontier convolution is shown in Fig. 4c.

3. *Sum over entire sensor spread.* We now build up the final 3D grid in C-space to annotate map entropy reduction for each hypothetical robot pose. Once the frontier convolution layers for all ray directions have been computed, we sum all the layers within the sensor orientation range to annotate it in the corresponding cell in the C-space map entropy reduction grid. The result of this step for one particular robot orientation is shown in frame d of the same figure.

4.2.1. Localization uncertainty compensation

The factor accounting for the path probability distribution $\alpha(p(x|u, z))$ in the joint entropy decrease estimate (Eq. 9) has an intuitive meaning. It is obvious

10

that exploratory trajectories that depart from well localized priors produce more accurate maps than explorations that depart from uncertain locations. In fact, sensor readings coming from robot poses with large marginal covariance values may spoil the map adding bad cell classifications, i.e., adding entropy. Since we already have localization uncertainties encoded in the Pose SLAM graph, these are used to weight the entire entropy reduction map. It suffices to weight the entire entropy reduction map with the inverse of the determinant of the marginal covariance at the current configuration

$$\alpha(p(x|u,z)) = \frac{1}{|\mathbf{\Sigma}_{kk}|} \tag{14}$$

Exploratory trajectories that depart from uncertain configurations will be weighted negatively, giving predominance to the path entropy reduction term in those cases. In this way, we achieve the desired effect of alternance between exploratory and relocalization paths.

4.3. Potential field and gradient path

Once the path entropy decrease and the map entropy decrease estimation are computed densely for all C-space cells, we combine them weighing the map entropy by the localization uncertainty compensation as described to obtain a dense estimate of information gain. Dense in the sense that it has been computed for each cell in the C-space grid. But, for this estimate to become the needed scalar function ϕ, we still need one more step. To avoid long valleys with no entropy change, and to enforce attraction to the informative poses, the grid is turned into a potential field, smoothing it using a harmonic function of the form

$$\phi_{xy\theta} = \frac{1}{6}(\phi_{x^\pm y\theta} + \phi_{xy^\pm\theta} + \phi_{xy\theta^\pm}) \tag{15}$$

where the superscript $\pm$ is used to indicate neighbor cells in the C-space grid. See Fig. 5 for a one-dimensional example.

4.4. Obstacle avoidance and boundary conditions

For our gradient descent exploration strategy, we have identified two types of local minima. One type of local minima refers to those entropy reduction configurations at the selected goals, that when reached by the robot, become entropy reduction valleys in the next computation of the potential field. These do not constitute a problem. The second type of local minima are bottlenecks and dead ends near obstacles.

To account for obstacles, in our computation of the grid we have considered them to adequately propagate entropy change along sensor rays taking into account occlusions, but we have still not penalized configurations that get close to them. We resort to the use of boundary conditions of the form

$$\nabla\phi(c) = g(c) \qquad \forall c \in \partial\mathcal{S}_{\text{occupied}} \tag{16}$$
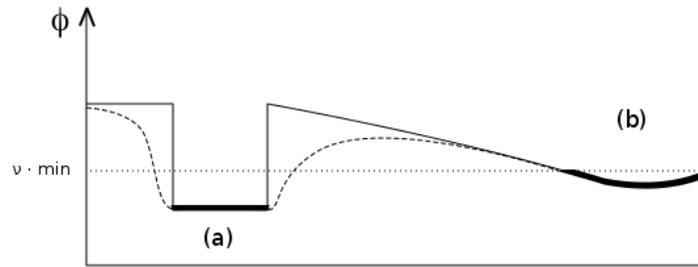
Figure 5: The C-space entropy change grid is smoothed with a harmonic function updating the values below v at each smooth iteration. The solid and the dotted lines represent the initial joint entropy decrease and the potential information field resulting after smoothing, respectively. Zone (a) represents a region with steep entropy reduction within the sensor range to guarantee loop closure. Zone (b) represents an area worth exploring.
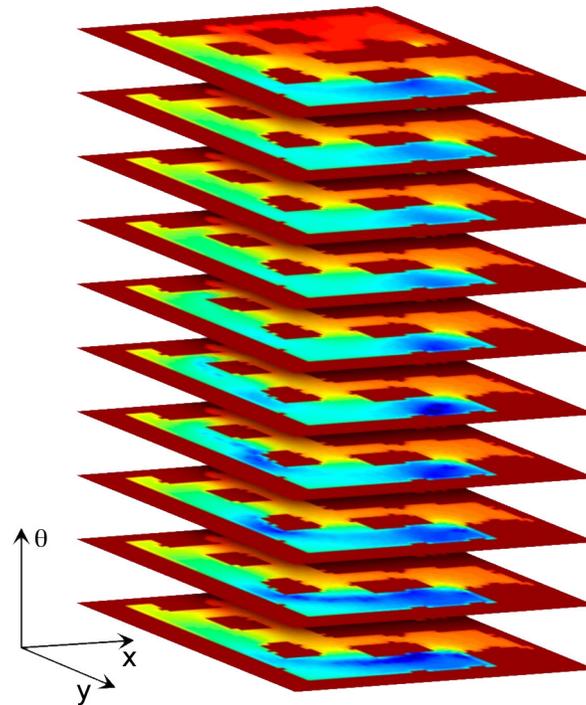

Figure 6: Potential field values for some C-space orientation layers. The blue regions indicate competing exploratory and loop closure candidate configurations at that robot orientation.
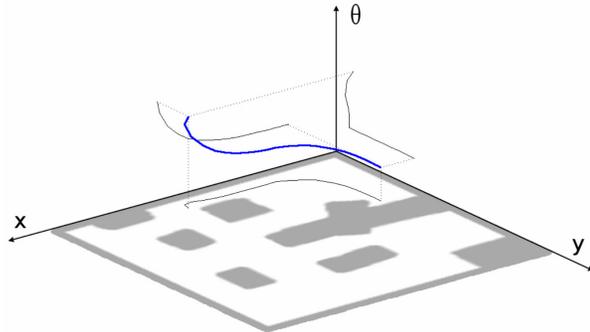
12

Figure 7: Exploration trajectory computed as the gradient descent of the potential information field from Fig. 6.

as in [12], with the difference that instead of using Neumann boundary conditions to guarantee flow parallel to obstacles, we still want some repulsive perpendicular effect from them. This effect can be achieved by mirroring weighted inner cell values near obstacles. In the method reported here we start each planning step weighting the repulsion term with a small constant to avoid bottlenecks at local minima and increasing it and re-planning in case a collision is detected. This iterative increment of the repulsion weights has the final effect of pushing the robot away from the bottleneck in the next planning iteration.

The final path is obtained by traversing the gradient field from the current robot configuration to the robot configuration with largest joint entropy reduction. Some C-space orientation layers of the resulting information potential field are shown in Fig. 6, and the resulting trajectory of its gradient descent is depicted in Fig. 7.

5. Simulations

5.1. Scenarios

Simulations are carried out in three different scenarios. The first one is the widely used cave-like environment available from [5], scaled to a size of $20\,\text{m} \times 20\,\text{m}$. The second one is a modification of this cave-like environment with more obstacles and corridors to enforce bottleneck situations. This scenario was used to investigate whether the gradient descent in the potential information field allowed for easy transitions through these narrow passages. The third scenario is the larger Freiburg indoor building 079, also available from [5]. It is a more challenging scenario, allowing us to show the limitations of the algorithm with regards to scalability when compared to the other methods. The scenarios used are shown in Fig. 8.
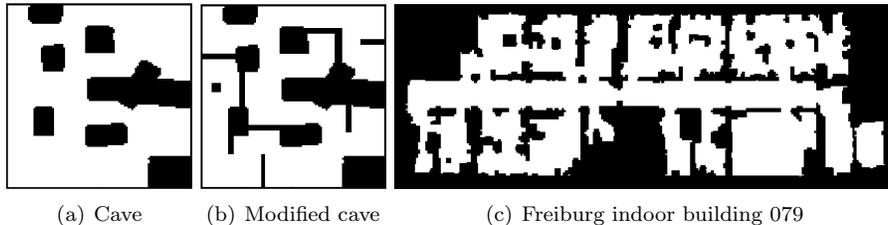
(a) Cave   (b) Modified cave   (c) Freiburg indoor building 079

Figure 8: Scenarios used in the simulations.

5.2. Methods

Several simulations were performed to compare the proposed solution with other exploration strategies. We wanted to evaluate to what extent path entropy reduction plays a role on the improvement of map quality. To this end, we compare our method, which aims to reduce both path and map entropies, with a greedy coverage technique such as frontier-based exploration, which seeks only map entropy decrease.

The frontier-based method used drives always the robot to the closest frontier larger in size than a predefined threshold, without considering neither localization nor map uncertainties. In our simulations, frontiers larger than 9 cells (90 cm) were considered first. Once there were no frontiers of that size, the threshold was reduced until there was no frontier left to visit or the simulation steps limit was reached. The trajectory to the chosen frontier was computed with an RRT* optimal planner [8].

Secondly, we wanted to evaluate whether dense estimation of the entropy reduction field for one step look ahead optimization is a better choice than entropy reduction for a sparse set of goals but with larger look ahead optimization sequences. To that end, we compare the method against Active Pose SLAM [17], a technique that evaluates joint entropy reduction for a limited set of exploratory and path revisiting goals.

Active Pose SLAM evaluates entropy as in Eq. 8, with the difference that it does not allow for localization uncertainty compensation, so $\alpha(p(x|u,z) = 1$, and with the disadvantage that it is not computed densely, but only for a limited number of candidate trajectories. The method estimates, at each iteration, the posterior for a reduced number of exploratory trajectories seeking frontiers and a number of loop closing trajectories computed using the PRM algorithm. Using such posteriors it evaluates joint entropy decrease at those locations and chooses the trajectory with maximum joint entropy decrease. This method also uses a naive implementation of occupancy grid which is shown to be significantly more expensive to compute than ours.

For each environment and exploration method, we performed 5 simulations due to the effects of the simulated noise in motion and sensors and the randomness of the RRT* used in the frontier-based method and the PRM used by the Active Pose SLAM. The simulation steps limits were 200 and 400 for the

14

cave-like and the Freiburg maps, respectively.

All the simulations used the same parameter values. The robot is fitted with a laser range sensor with a match area of $\pm 1\,\mathrm{m}$ in x and y, and $\pm 0.35\,\mathrm{rad}$ in orientation. This corresponds to the maximum range in configuration space for which a link between two poses can be established. The robot motion was estimated with an odometric sensor with noise covariance $\boldsymbol{\Sigma}_u = \mathrm{diag}(0.1\,\mathrm{m},\ 0.1\,\mathrm{m},\ 0.0026\,\mathrm{rad})^2$. Measurement noise covariance was fixed at $\boldsymbol{\Sigma}_y = \mathrm{diag}(0.05\,\mathrm{m},\ 0.05\,\mathrm{m},\ 0.0017\,\mathrm{rad})^2$ and laser scans were simulated by ray casting over a ground truth grid map of the environment using the true robot path. Relative motion constraints were measured using the iterative closest point algorithm. The initial uncertainty of the robot pose was set to $\boldsymbol{\Sigma}_0 = \mathrm{diag}(0.1\,\mathrm{m},\ 0.1\,\mathrm{m}, 0.09\,\mathrm{rad})^2$. Informative loop closures were asserted at $\mathcal{I} = 2.5\,\mathrm{nats}$.

5.3. Metrics

A number of different metrics were used to compare the three methods. First, to evaluate the speed-up of our new log odds occupancy grid computation, we measured the time taken by the algorithm to produce the whole occupancy map vs. the time taken by Active Pose SLAM for the same task, without considering other processes (state and entropy estimation, planning, navigation, etc).

Secondly, to compare the three exploration methods at run time, we stored on average and at each iteration, path and map entropy values for each of the methods and scenarios; the map coverage, measured as the number of cells labeled in the occupancy map; and the map error, measured as the number of cells in the occupancy map which were inconsistent with at least one rendered sensor data point measured at the respective mean of the estimated path pose.

As shown in the following subsection, map entropy decrease is almost equivalent to measuring coverage in the same way that map error is correlated to path entropy, i.e, an accurate path is a good indicator of adequate sensor registration and hence, map consistency.

Two other measures of performance computed were total execution time, including all the different processes of each method except for the mapping, and the total number of loop closures computed by each of the methods, which can also be related to the overall path entropy reduction.

5.4. Results

5.4.1. Mapping

Table 1 shows the average computational cost required to build a map for each of the different scenarios by the Active Pose SLAM method and by our new implementation, which aggregates local log odds occupancy maps into a common reference frame. The table shows how in general, our new implementation is an order of magnitude faster than the one used in Active Pose SLAM. This speed-up increase is accentuated even more for the larger Freiburg scenario.

This speed-up in computation is achieved mainly thanks to the affine transformations computed through sheers and translations that aggregate the local log odd maps into a global reference frame as described in Sec. 3.

	Active Pose SLAM	Aggregated log odds
Cave	$47.83\,\text{s}$	$2.32\,\text{s}$
Modified Cave	$47.94\,\text{s}$	$3.12\,\text{s}$
Freiburg 079	$153.50\,\text{s}$	$5.80\,\text{s}$

Table 1: Average computational time required to build the occupancy map in seconds, without taking into account other processes.

(a) Average map entropy.

(b) Average path entropy.

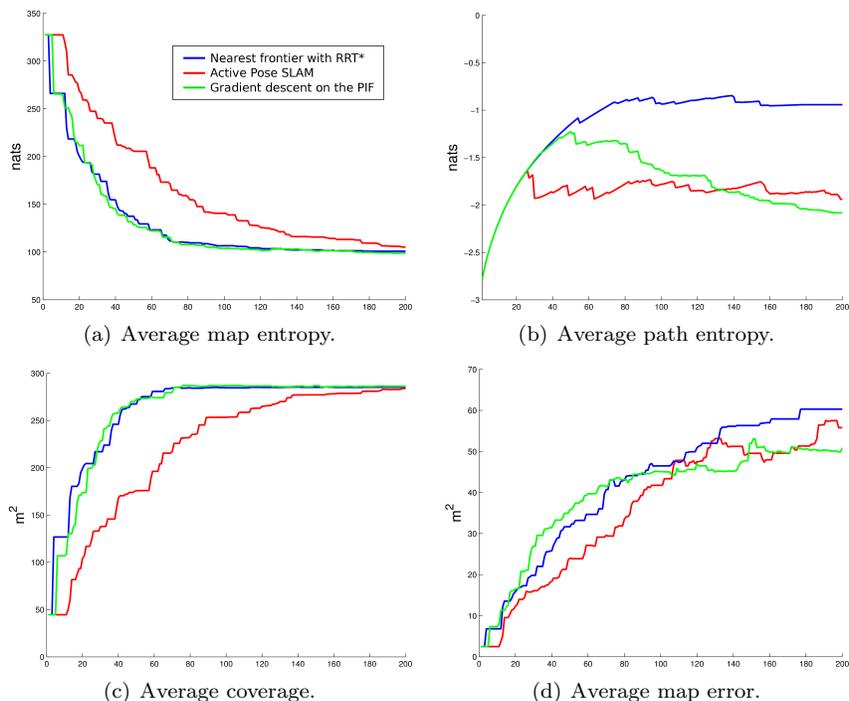(c) Average coverage.

(d) Average map error.

Figure 9: Cave exploration. Average map and path entropies, coverage and map error over 5 simulation runs. In blue, the frontier-based method with RRT* exploratory paths. In red, Active Pose SLAM. In green, gradient descent on the potential information field. Best viewed in color.

5.4.2. Cave map

Figure 9 shows plots for average map and path entropies, coverage, and map error, over 5 simulation runs of a 200 step exploration sequence of the cave scenario. The plots readily show the strong correlation between map entropy and coverage. Map error on the other hand is not only linked to path accuracy, but also to the size and morphology of the environment explored. Such difference is more acute for the other two scenarios which contain larger amounts of walls and obstacles.

As a consequence of path entropy minimization in both methods (Active

16

Performance metric	Nearest frontier with RRT*	Active Pose SLAM	Gradient descent on the PIF
Final path entropy	-0.94 nats	-1.94 nats	-2.08 nats
Final map entropy	100.87 nats	105.30 nats	98.70 nats
Total time	1480.84 s	21357.03 s	3487.74 s
Loops closed	5.2	19.8	27.4
Coverage	$285.17\,\mathrm{m}^2$	$283.80\,\mathrm{m}^2$	$286.38\,\mathrm{m}^2$
Map error	$60.30\,\mathrm{m}^2$	$55.78\,\mathrm{m}^2$	$50.91\,\mathrm{m}^2$

Table 2: Cave map exploration. Average comparison of different performance metrics for 5 runs of the three exploration methods discussed.

(a) Closest frontier with RRT*.

(b) Active Pose SLAM.

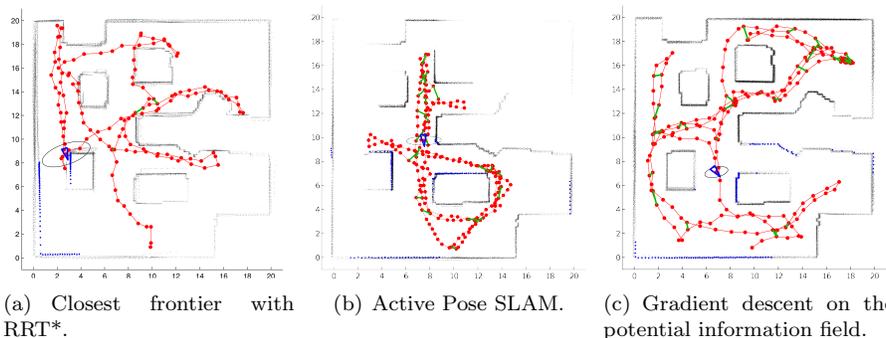(c) Gradient descent on the potential information field.

Figure 10: Final trajectories after an execution of 200 steps in the cave map for the three exploration methods. In red the robot path, in green the loop closure links, in black the whole raw sensor data rendered at the last path estimate, and in blue the marginal robot pose estimate for the current state (mean and variance) along with the sensed data at that location.

Pose SLAM and gradient descent on the potential information field), lower levels of map error are obtained for similar final coverage areas than pure frontier-based exploration. On average, for this scenario, the proposed method, gradient descent of the potential information field, achieves the largest coverage with the least amount of map error, as shown in Table 2.

Figure 10 shows single runs for the three methods for this map. The effect of path entropy reduction can be easily observed in these plots as well. The blue dots indicate the final robot pose estimate and its corresponding sensor data. It can be seen how the two methods that take into account path uncertainty reduction in their exploration strategies provide more accurate final pose estimates (blue and black sensor registration matches), whereas pure frontier based exploration does not account for that. Since there is no loop closure enforcement, there is no guarantee that the final path will be optimized.

It tuns out however that without localization uncertainty compensation, Active Pose SLAM results in an exploration strategy that is too conservative. For the same plant and sensor noise levels, the technique weights more localiza-

tion than exploration and hence coverage grows slower than in the other two methods.

Our method outperforms both other methods reaching full coverage much faster, and with lower path entropy. In addition, the computational time for the aggregated exploration and planning routines is significantly lower than Active Pose SLAM and competitive with the frontier-based method.

In the figure, it can also be observed how the frontier-based strategy results in many collisions with many frontiers misclassified due to the larger path uncertainties. In contrast, the exploration method using the gradient descent on the potential information field produces valleys of high information at loop closures and away from the repulsive obstacles.

Another difference between the two methods is that frontier-based exploration ceases once full coverage is reached and there are no further frontiers to visit. On the contrary, our method even when reaching full coverage, might continue optimizing the path, seeking revisiting trajectories to close loops, and hence improving the map estimate.

5.4.3. Modified cave map

This second scenario is a modification of the cave map in which we have added walls and obstacles to create a more challenging environment. The objective in designing this setting was to analyze whether the gradient descent approach would get stuck in local minima at corridors or dead ends.

Figure 11 shows once more averaged metrics for the three methods in this scenario for five simulation runs. One interesting thing to note is that frontier-based exploration rapidly enforces coverage, reaching low map entropy values sooner than the other methods, but being unable to decrease such value at the end of the simulation. The reason for this is that disregarding path entropy minimization in frontier-based does not enforce loop closure. For that same reason frontier-based exploration produced the largest map errors. The two joint entropy minimization schemes performed reasonably well with regards to path entropy reduction. However, our method was more conservative in exploration at the beginning, reaching better coverage at the end, whereas Active Pose SLAM was too conservative heavily refining its localization estimates for this scenario and failed to fully explore it.

These same findings are also contrasted in Table 3 and in the exemplary test runs plotted in Fig. 12. See for instance the large number of collisions that needed to be accounted for in the frontier-based method due to its larger map error values.

Computationally speaking, our gradient descent takes similar effort to compute than frontier-based exploration, and is about six times faster than Active Pose SLAM on average for this scenario.

5.5. Freiburg 079 map

The Freigurg 079 building is quite more complex than the two previous scenarios. We choose this environment in order to test not only the bottleneck
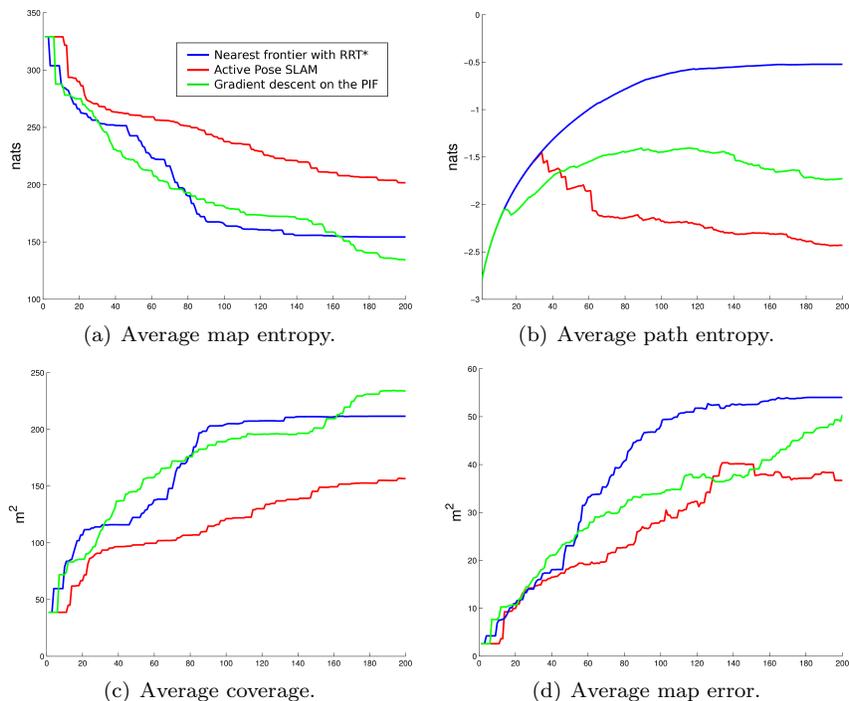
(a) Average map entropy.

(b) Average path entropy.

(c) Average coverage.

(d) Average map error.

Figure 11: Modified cave exploration. Average map and path entropy, coverage and map error over 5 simulation runs. Color labels as in Fig. 9.

Performance metric	Nearest frontier with RRT*	Active Pose SLAM	Gradient descent on the PIF
Final path entropy	$-0.52\,\mathrm{nats}$	$-2.43\,\mathrm{nats}$	$-1.73\,\mathrm{nats}$
Final map entropy	$154.33\,\mathrm{nats}$	$201.65\,\mathrm{nats}$	$134.04\,\mathrm{nats}$
Total time	$530.39\,\mathrm{s}$	$3102.84\,\mathrm{s}$	$595.87\,\mathrm{s}$
Loops closed	10.8	35.8	27.6
Coverage	$211.55\,\mathrm{m}^2$	$156.70\,\mathrm{m}^2$	$233.90\,\mathrm{m}^2$
Map error	$54.02\,\mathrm{m}^2$	$36.70\,\mathrm{m}^2$	$50.38\,\mathrm{m}^2$

Table 3: Modified cave map exploration. Average comparison values of different performance metrics for 5 runs of the three exploration methods discussed.

effect since it is the main drawback of the potential field method, but also to study how the method scales for larger settings.

In this scenario, there are no big loops and the corridor is straight. This helps those methods that do not plan for loop closure such as frontier-based exploration, beause these loop closures will occur anyways each time the robot comes out from an explored room back into the corridor.

As in the previous scenarios, Active Pose SLAM performance is characterized

19

(a) Nearest frontier with RRT*.

(b) Active Pose SLAM.

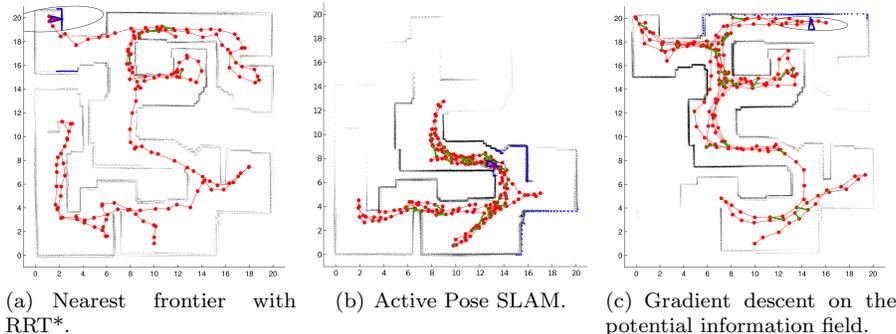(c) Gradient descent on the potential information field.

Figure 12: Final trajectories after an execution of 200 steps in the modified cave map of the three exploration methods. Color meanings as in Fig. 10.

by a very conservative behavior with regards to localization uncertainty. For a simulation run of 400 steps, the method did not reach full coverage. The result was a largely connected graph of nodes around the initial robot pose, with very low path entropy, leaving the rest of the scene highly unexplored. As expected, the closest frontier method behaves the opposite way, it reaches nearly full coverage and, due to the topology of the scenario, closes a significant number of loops. Path entropy and map error however are still larger than in the other two methods.

Our method shows a more balanced performance with regards to exploration vs. exploitation, as shown in Fig. 14 which plots the resulting map for one instance of the simulations after 400 steps for each of the three methods discussed.

For the potential information fields method, the local minima drawback is detected in two different cases, as a bottleneck effect near obstacles and also as local minima of path entropy. The first case is mainly a planning drawback, narrow door passages tend to trap the robot because of the repulsive boundary conditions at obstacles. To get away from these local minima, it sufficed to increase the potential field at that node, turning a valley into a ridge. This iterative potential field increase constituted a two-fold increase in computation time when compared with the frontier-based method, as shown in Table 4.

The second case of local minima appears in the second half of the simulation. At that point marginal pose uncertainties have become large enough to populate the Pose SLAM graph with plenty of nearby nodes. A gradient descent path will seek loop closure with such nodes but with little path entropy reduction. Navigating around these nearly flat potential fields will not significantly reduce map entropy either, but will introduce motion and sensor error, hence producing map error in the end as shown in Fig. 13. The end result is that for the second half of the simulation, coverage does not increase as fast as in the first part. A possible solution to this problem would be to change our state representation to local coordinates instead of global. In that case, marginals close to the current robot location would always have smaller variance, and path entropy reduction

20

(a) Average map entropy.

(b) Average path entropy.

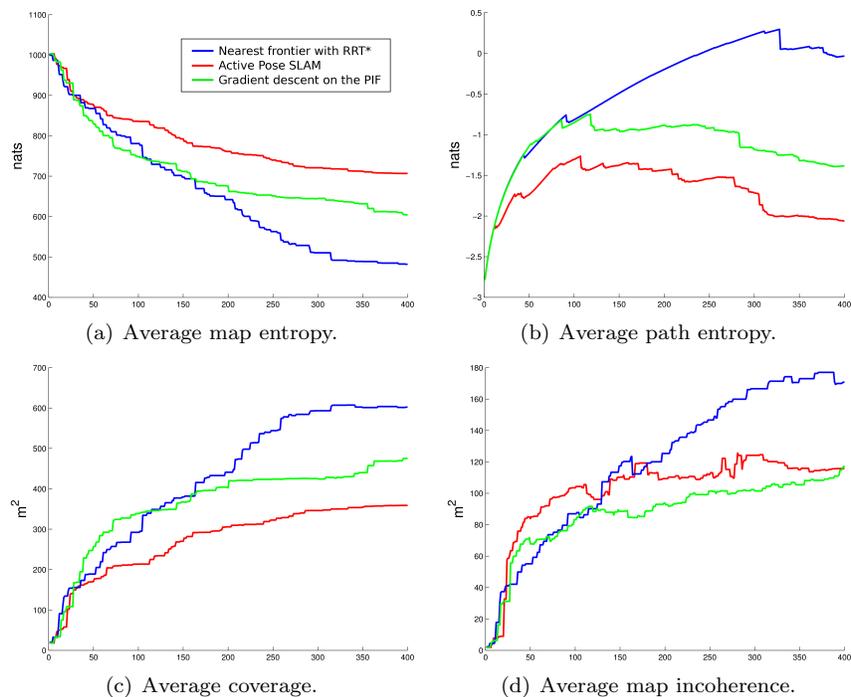(c) Average coverage.

(d) Average map incoherence.

Figure 13: Freiburg 079 building exploration. Average map and path entropy, coverage and map error over 5 simulation runs. Color labels as in Fig. 9.

Performance metric	Nearest frontier with RRT*	Active Pose SLAM	Gradient descent on the PIF
Final path entropy	-0.03 nats	-2.06 nats	-1.3 nats
Final map entropy	481.74 nats	706.84 nats	603.68 nats
Total time	1466.91 s	33899.97 s	3727.48 s
Loops closed	18.0	67.5	49.0
Coverage	$602.24 \, \mathrm{m}^2$	$358.82 \, \mathrm{m}^2$	$475.56 \, \mathrm{m}^2$
Map error	$171.02 \, \mathrm{m}^2$	$115.74 \, \mathrm{m}^2$	$117.62 \, \mathrm{m}^2$

Table 4: Freiburg 079 building exploration. Average comparison values of different performance metrics for 5 runs of the three exploration methods discussed.
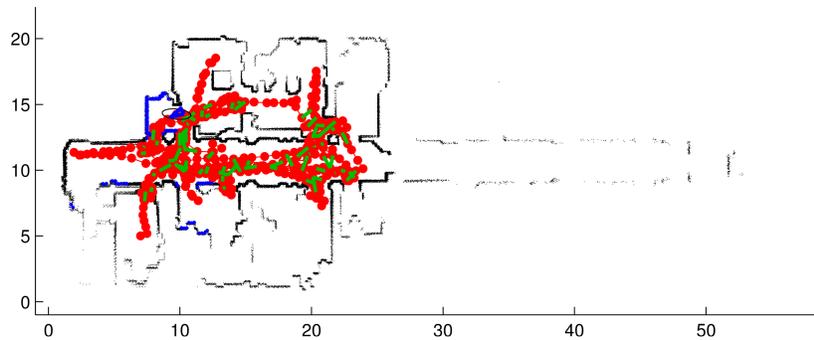
will always be maximal away from the current robot location. We leave this issue as a topic for further research.
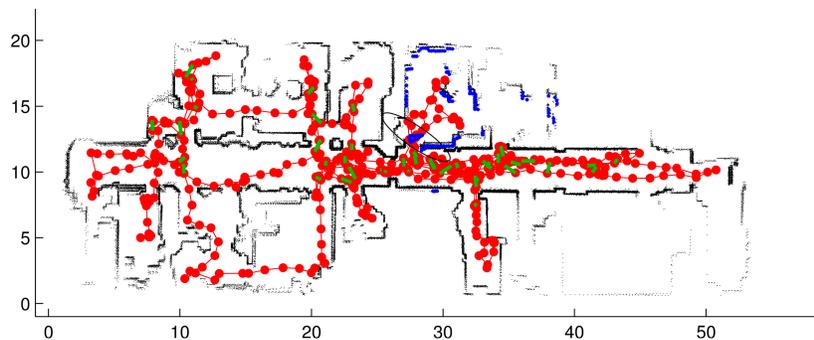
6. Conclusions

We have proposed a decision theoretic approach for autonomous mobile robot exploration. The goal of the approach is to determine the movements
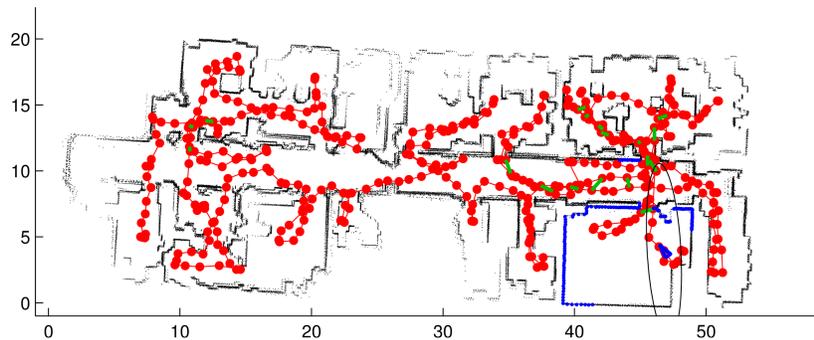
(a) Nearest frontier with RRT*.

(b) Active Pose SLAM.

(c) Gradient descent on the potential information field.

Figure 14: Final trajectories after an execution of 400 steps in the Freiburg 079 map of the three exploration methods. Color meanings as in Fig. 10.

that aim at reducing the entropy about both the robot state and the map. Map entropy is computed on an occupancy grid, whereas path entropy is computed from the uncertainties of an underlying graph-based SLAM solution.

Exploratory paths are computed by traversing a gradient descent on a C-

space field of path and map entropy reduction estimates. The technique makes use of very efficient convolutions first, to project boundaries along sensor rays, and secondly, to integrate entropy measures at independent robot orientation layers.

The method outperforms, in terms of map quality, frontier-based exploration which only seek coverage. The method also outperforms another method that seeks joint path and map entropy minimization but only for a limited number of exploratory trajectories. In this case, both in terms of coverage and speed of computation.

Contrary to other exploration methods, joint path and map entropy decrease is computed densely over the C-space. In computing the gradient descent on this field, the method assumes an holonomic platform. Also, the use of the gradient descent makes the method sensitive to some local minima drawbacks. Future work is on using this dense joint entropy decrease estimation to compute potential exploratory goals, but leaving the path planning strategies to other methods that can account for the non-holonomic restrictions and also be less sensitive to this local minima [19].

Future work also includes, the switch from global to local representations, an implementation in ROS, and comparison against competing approaches on real scenarios.

References

[1] F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte. Information based adaptative robotic exploration. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 540–545, Lausanne, Oct. 2002.

[2] E.P. de Silva, P.M. Engel, M. Trevisan, and M.A.P. Idiart. Exploration method using harmonic functions. *Robotics Auton. Syst.*, 51(1):25 – 42, 2002.

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

[4] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mobile robot navigation and mapping. *Int. J. Robotics Res.*, 18:650–668, 1999.

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

[6] 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 *Proc. IEEE Int. Conf. Robotics Autom.*, pages 1091–1096, Barcelona, Apr. 2005.

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

[8] S. Karaman, M.R. Walter, A. Perez, E. Frazzoli, and S. Teller. Anytime motion planning using the RRT*. In *Proc. IEEE Int. Conf. Robotics Autom.*, pages 1478–1483, Shanghai, May 2011.

[9] A. Kim and R. M. Eustice. Perception-driven navigation: Active visual SLAM for robotic area coverage. In *Proc. IEEE Int. Conf. Robotics Autom.*, pages 3196–3203, Karlsruhe, May 2013.

[10] C. Leung, S. Huang, and G. Dissanayake. Active SLAM for structured environments. In *IEEE Int. Conf. Robotics Autom.*, pages 1898–1903, Pasadena, 2008.

[11] C. Leung, S. Huang, N. Kwok, and G. Dissanayake. Planning under uncertainty using model predictive control for information gathering. *Robotics Auton. Syst.*, 54(11):898–910, Nov. 2006.

[12] R. Shade and P. Newman. Choosing where to go: Complete 3D exploration with stereo. In *Proc. IEEE Int. Conf. Robotics Autom.*, pages 2806–2811, Shanghai, May 2011.

[13] R. Sim and N. Roy. Global A-optimal robot exploration in SLAM. In *Proc. IEEE Int. Conf. Robotics Autom.*, pages 661–666, Barcelona, Apr. 2005.

[14] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *Int. J. Robotics Res.*, 5(4):56–68, 1986.

[15] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Robotics: Science and Systems I*, pages 65–72, Cambridge, Jun. 2005.

[16] R. Valencia, M. Morta, J. Andrade-Cetto, and J.M. Porta. Planning reliable paths with Pose SLAM. *IEEE Trans. Robotics*, 29(4):1050–1059, 2013.

[17] R. Valencia, J. Valls Miró, G. Dissanayake, and J. Andrade-Cetto. Active Pose SLAM. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 1885–1891, Vilamoura, Oct. 2012.

[18] J. Vallvé and J. Andrade-Cetto. Mobile robot exploration with potential information fields. In *Proc. Eur. Conf. Mobile Robots*, pages 222–227, Barcelona, Sep. 2013.

[19] J. Vallve and J. Andrade-Cetto. Dense entropy decrease estimation for mobile robot exploration. In *Proc. IEEE Int. Conf. Robotics Autom.*, pages 6083–6089, Hong Kong, May 2014.

[20] T. Vidal-Calleja, A. Sanfeliu, and J. Andrade-Cetto. Action selection for single camera SLAM. *IEEE Trans. Syst., Man, Cybern. B*, 40(6):1567–1581, Dec. 2010.

[21] P. Whaite and F. P. Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(3):193–205, Mar. 1997.

[22] B. Yamauchi. A frontier-based approach for autonomous exploration. In *IEEE Int. Sym. Computational Intell. Robot. Automat.*, pages 146–151, Monterrey, 1997.

[23] B. Yamauchi. Frontier-based exploration using multiple robots. In *Int. Conf. Autonomous Agents*, pages 47–53, Minneapolis, 1998.