

# Dense Entropy Decrease Estimation for Mobile Robot Exploration

Joan Vallvé and Juan Andrade-Cetto

**Abstract**—We propose a method for the computation of entropy decrease in C-space. These estimates are then used to evaluate candidate exploratory trajectories in the context of autonomous mobile robot mapping. The method evaluates both map and path entropy reduction and uses such estimates to compute trajectories that maximize coverage whilst minimizing localization uncertainty, hence reducing map error. Very efficient kernel convolution mechanisms are used to evaluate entropy reduction at each sensor ray, and for each possible robot position and orientation, taking frontiers and obstacles into account. In contrast to most other exploration methods that evaluate entropy reduction at a small number of discrete robot configurations, we do it densely for the entire C-space. The computation of such dense entropy reduction maps opens the window to new exploratory strategies. In this paper we present two such strategies. In the first one we drive exploration through a gradient descent on the entropy decrease field. The second strategy chooses maximal entropy reduction configurations as candidate exploration goals, and plans paths to them using RRT\*. Both methods use PoseSLAM as their estimation backbone, and are tested and compared with classical frontier-based exploration in simulations using common publicly available datasets.

## I. INTRODUCTION

In robotic exploration, the policy to compute a path is usually motivated by two driving forces. On the one hand, one wishes to maximize coverage, while at the same time maintaining localization uncertainty to a minimum. Initial approaches to the problem compute separate estimates for the reduction of map and path uncertainties. Feder et al. [1], propose a metric to evaluate uncertainty reduction as the sum of the independent robot and landmark entropies with a one step exploration horizon. Bourgault et al. [2] 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. To consider joint map and path entropy reduction, Vidal et al., [3] tackled the issue taking into account robot and map cross correlations for the Visual SLAM EKF case.

Another way to take into account both the reduction of map and path entropies jointly is to compute entropy reduction directly in configuration space, rather than only for the occupancy map as most methods do. One approach to do this computation is the work of Torabi et al. [4]. The maximal expected entropy reduction is computed as the sum of marginal

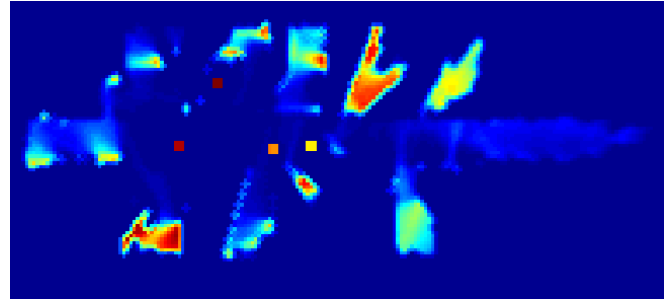


Fig. 1. Joint path and map entropy reduction estimates computed for one C-space orientation layer. The red regions indicate areas with large entropy reduction values, either for exploration (smooth areas) or for loop closing (isolated dots) at that particular robot orientation.

entropy terms for a number of possible configurations. But still in that work, correlations are ignored. Mutual map-robot entropies are not evaluated for computational efficiency. It is noted however, that the expected entropy reduction for each sensor location can be computed as the sum of the expected entropy reduction for each individual sensor beam. But, the double summation configurations-beams turned out to be computationally expensive, and thus it can only be computed for a limited number of configurations.

In this work, we compute a dense estimate of entropy reduction for the entire configuration space. To do so, we compute directly the map and path entropy reduction estimates for each possible robot configuration using frontier-visibility. Using the fact that sensor beams repeat for many orientations at the same position, and using very efficient kernel convolutions, we are able to compute these estimates very fast and densely. To illustrate the idea, Fig. 1 shows the entropy decrease values for one particular orientation layer and all robot locations at a given instance in time for the Freiburg map. By estimating the entropy decrease densely, i.e., for all robot configurations, we can now evaluate motion policies not only for a few candidate configurations [5], [6], [7], but for the entire C-space. The method uses PoseSLAM as its estimation backbone and hence performs path entropy reduction during state update when poses are revisited.

This fast estimation of dense joint path and map entropy reduction opens the possibility to new exploration strategies. We propose two such exploration strategies in this paper. The first one resorts to gradient descent on the entropy decrease field, as initially proposed in [8]. The second one, chooses the maxima in that field, and searches for a path to those configurations using RRT\* [9]. We report results comparing both methods against frontier-based exploration in different environments. We show through the experiments

This work has been supported by the Spanish Ministry of Economy and Competitiveness under Project DPI-2011-27510 and by the EU Project CargoAnts FP7-605598.

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

that whereas the gradient decrease strategy guarantees local optimality, by combining the use of an RRT-based path search with our entropy decrease goal selection mechanism, we can not only concentrate in global optimality instead, but also enforce more loop closing and hence better global estimates.

## II. POSE SLAM

The proposed exploration strategy uses Pose SLAM as its estimation backbone. In Pose SLAM [10], a probabilistic estimate of the robot pose history is maintained as a sparse graph. State transitions result from the composition of motion commands to previous poses, and the registration of sensory data also produces relative motion constraints, but now between non-consecutive poses.

Graph links indicate relative geometric constraints between robot poses, and the density of the graph is rigorously controlled using information measures. 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.

Pose SLAM does not maintain a grid representation of the environment. 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, finely grained maps can be produced for the computation of traversability maps [11], or for the computation of information distribution maps [12]. Occupancy maps at coarse resolutions are produced in [7] to evaluate the effect of candidate trajectories in a related exploration scheme. But grid maps are not always needed. In [13] for instance, there is no need to render such map to plan optimal trajectories in belief space.

## III. JOINT PATH AND MAP ENTROPY DECREASE ESTIMATION OVER C-SPACE

In this work, the joint entropy is approximated, as in [7], as the sum of the entropy of the path  $x$ , given all motions  $u$  and observations  $z$ , and that of the map given full confidence on the path

$$\begin{aligned} H(x, m|u, z) &= H(x|u, z) + \int_x p(x|u, z) H(m|x, u, z) dx \\ &\approx H(x|u, z) + H(m|u, z). \end{aligned} \quad (1)$$

The evaluation of map entropies at the pose means is a sensible approach, since we are confident that the estimated path is a good approximation to the real path. Moreover, map entropies at locations with poor localization estimates would most likely correspond to areas without salient sensory data, and the computation of map entropy difference at those means would be negligible, just as it would be the evaluation of the integral in Eq. 1 at those same locations, even if approximated with particles [5].

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 entropy 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, for the estimation of the overall entropy decrease we need only to evaluate separately the two terms in Eq. 1 for each discretized robot configuration in C-space.

### A. Map entropy reduction

For a map with size cell  $l$ , its entropy can be computed as the scalar sum

$$H(m|u, z) = -l^2 \sum_{c \in m} (p_c \log p_c + (1 - p_c) \log(1 - p_c)), \quad (2)$$

where  $p_c$  is the classification probability of cell  $c$ . We classify cells as free, occupied, or unknown. The value of  $p_c$  ranges from 0 for absolute certainty of being free to a value of 1 meaning absolute certainty of being occupied, and 0.5 at the middle range for total uncertainty about the cell state. Moreover, we maintain a frontier label on those unknown cells close to free ones.

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. Estimating the actual number of cells that will be discovered is impossible, but it is heavily linked to the size of the frontier visible to the sensor. We are content with approximating entropy reduction as the number of frontier cells visible to the sensor from each robot configuration.

We are able to compute this entropy change very efficiently with the following three steps:

- 1) *Obstacle occlusion mask.* We generate a 3-dimensional grid. Its dimensions are  $x$ ,  $y$ , and the direction of each laser ray. For each ray orientation layer, an obstacle occlusion mask is created, annotating whether the nearest non-free cell along the 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 2b.
- 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,

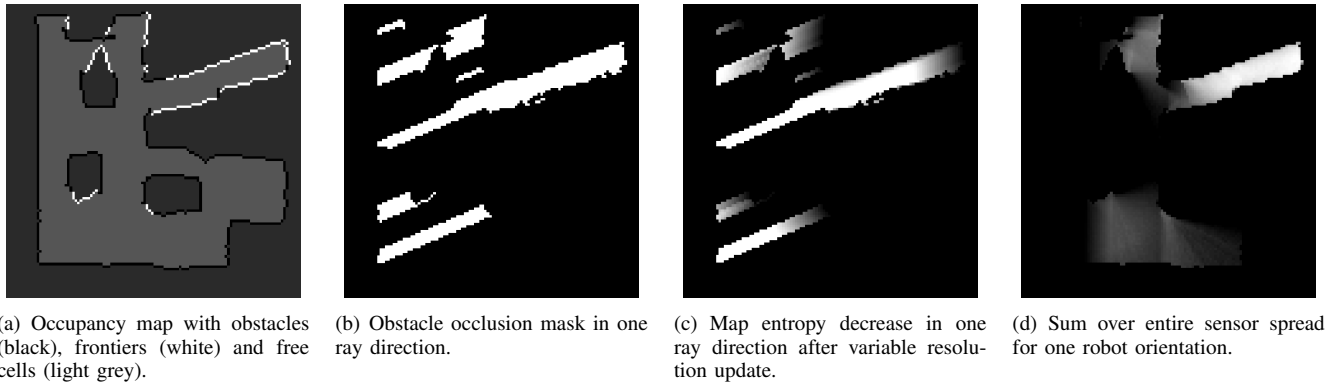


Fig. 2. Computation of map entropy change for one particular robot orientation.

thus it is necessary to compensate for this in order not to overestimate the number of frontier cells being observed. Ray cast density at each cell  $r$  is a function of the distance from the robot to that cell and the angle  $\beta$  between two consecutive sensor rays. For each ray orientation layer, a convolution is made with a one-dimensional motion kernel weighted with  $\min(1, r \tan \beta)$ . The result of this frontier convolution is shown in Fig. 2c.

- 3) *Sum over entire sensor spread.* We now define a 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 entropy reduction grid. The result of this step is shown in frame d of the same Figure.

Furthermore, it is obvious 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 to the determinant of the marginal covariance  $|\Sigma_{kk}|$  at the current configuration. In this way, exploratory trajectories that depart from uncertain configurations will be weighted negatively, giving predominance to the path entropy term in those cases.

### B. Path entropy reduction

To compute an estimate for the first term in Eq. 1, the entropy of the path could be approximated without taking into account correlations between poses [7], by averaging over the individual pose marginals

$$H(x|u, z) \approx \frac{1}{N} \sum_{i=1}^N \ln((2\pi e)^{(n/2)} |\Sigma_{ii}|). \quad (3)$$

Evaluating this term is not necessary since we are not interested in the individual entropy values, but only on

entropy change, i.e., information gain. And, as stated before, we are not evaluating entropy change for just one posterior pose, but for the whole discretized C-space. Assuming, as justified in the introductory paragraph to this Section, that we could establish a perception link from the current robot location to any other configuration of the C-space through a kinematic motion chain with equal cost, such motion would produce the same marginal posterior, and most importantly, with constant information gain, except at loop closure.

And, to evaluate the effect of each possible loop closure between the current configuration  $j$ , and any configuration  $i$  already in the PoseSLAM graph, path entropy reduction is given precisely by the information gain

$$\mathcal{I}_{ij} = \frac{1}{2} \ln \frac{\mathbf{S}_{ij}}{|\Sigma_y|}, \quad (4)$$

with  $\Sigma_y$  the sensor covariance, and  $\mathbf{S}_{ij}$  the innovation covariance of the Pose SLAM update.

The parameter match area of the sensor is defined as the intervals in  $x$ ,  $y$  and  $\theta$  where loops can be closed by the sensor. Thus, a loop can be closed in each configuration in the C-space inside the match area of any previous pose of the trajectory. Instead of iterating over each cell in the C-space 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 grid with the corresponding information gain.

Finally, we obtain the joint entropy decrease estimation over the entire discretized C-space by adding the map and the path entropy decrements. This dense entropy decrease estimation can be used for different exploration strategies. Fig. 3 shows several orientation layers of the dense entropy decrease estimate for the Freiburg map.

## IV. PLANNING FOR JOINT ENTROPY MINIMIZATION

We now present two exploration strategies that make use of this densely computed entropy reduction estimate in the whole C-space. Both strategies have as objective, to drive the robot minimizing both map and path entropies, i.e., maximizing coverage while maintaining the robot well localized. The first strategy considers the entropy decrease estimate as a field, and searches its minima using a gradient

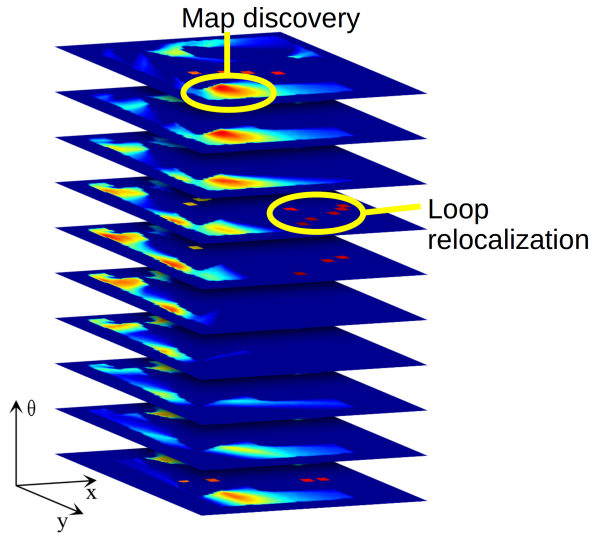


Fig. 3. Dense entropy decrease estimate in C-space. Each layer represents a slice of the configuration space at a different orientation value. Red areas indicate candidate configurations with maximum entropy reduction values.

descent method. The second approach chooses such critical points as goals but plans instead a trajectory through the free space with the aid of a RRT\* tree.

#### A. Gradient descent on an entropy decrease field

In gradient descent methods, the objective is to find a scalar function  $\phi$  defined over all C-space cells such that its gradient  $\nabla\phi$  will drive the robot to its minimum. In our case, the gradient of  $\phi$  will drive the robot to the configuration with largest joint path and map entropy decrease.

In contrast to our approach, [14] define a potential scalar function using attraction and repulsion fields on frontiers and 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 [15]. 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 reduction in map entropy since more frontier cells can be observed from these locations than from the frontier. See Fig 4.

For our entropy decrease grid to become the scalar function  $\phi$ , we still need one more step. To avoid long valleys with no entropy change, the grid is turned into a potential field, by cropping it first to a desired value  $v$  of 60% to define attraction areas. And then, smoothing it using a harmonic



Fig. 4. Configuration *a*: Exploration goal at a frontiers. Configuration *b*: Optimal map entropy reduction goal in C-space.

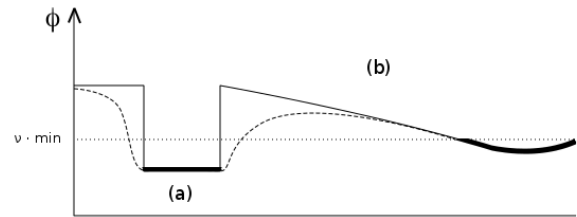


Fig. 5. The entropy decrease grid is cropped at a desired value  $v$  and smoothed with a harmonic function to produce the desired information potential field. Zone (a) represents a region with steep entropy reduction within the sensor range to guarantee loop closure. Zone (b) represents an area worth exploring.

function of the form  $\phi_{xy\theta} = \frac{1}{6}(\phi_{x\pm y\theta} + \phi_{xy\pm\theta} + \phi_{xy\theta\pm})$ , where the superscript  $\pm$  is used to indicate neighbor cells in the C-space grid. See Fig 5.

In our computation of the entropy grid we have considered obstacles to adequately propagate entropy change along sensor rays taking into account occlusions, but we have still not penalized configurations that get close to them. To this end, we resort to the use of boundary conditions as in [14], 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. A unity weight ( $w = 1$ ) means parallel traverse along the obstacle boundary, and larger values of  $w$  induce repulsion. In the method reported here we start each each planning step with low repulsion  $w$  to avoid bottlenecks at local minima and increase it and re-plan in case a collision is detected. The final path is obtained traversing the gradient field from the current robot configuration to the robot configuration with largest joint entropy reduction. Holonomic traverse is assumed in this planner.

#### B. RRT\* to the configuration with largest entropy decrease

Gradient descent methods performance poorly in large environments with many obstacles, requiring huge smoothing

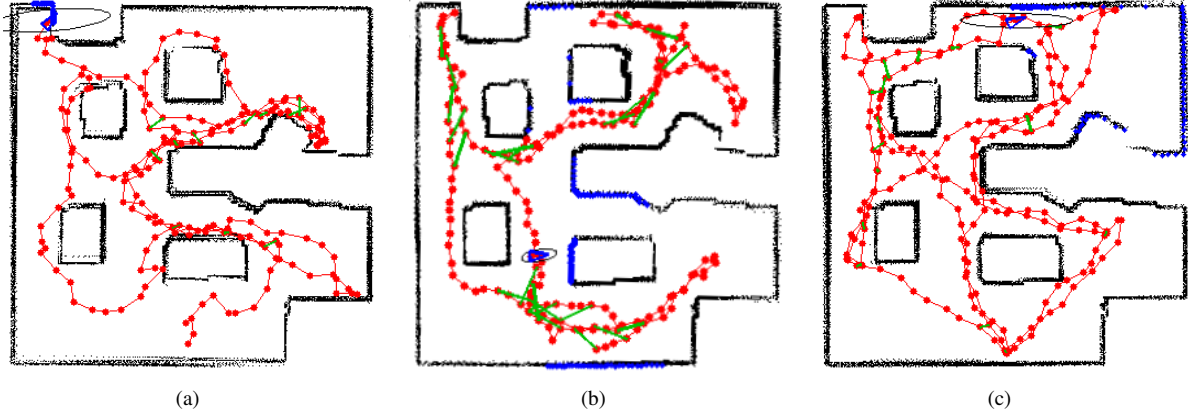


Fig. 6. Final trajectories after an each exploration strategy simulation of 200 steps. (a) Frontier-based with RRT\*. (b) EDE with Potential gradient descent. (c) EDE with RRT\*. In red the robot path, in green the loop closure links, and in black the occupancy map rendered at the last path estimate.

iterations and suffering severely of bottleneck effect and local minima (i.e. doors or thin passages). In addition, most mobile robots have non-holonomic restrictions. A second option explored in this work is to use a fast and optimal planner like RRT\* [9], [16], [17] to drive the robot to the configuration with largest entropy decrease. This method will renounce to the most entropy decreasing path and choose instead the shortest path in the free configuration space meeting non-holonomic restrictions if needed.

The RRT\* planner is more resilient to local minima effects and is faster since it does not require the above-mentioned smoothing iteration. In our implementation of the RRT\* planner, if exploration fails after a number of iterations to reach a goal, the node in the tree with largest value in the C-space entropy grid is chosen as goal.

An advantage of our joint estimation of map and path entropy reduction is that both these planning strategies nicely alternate between exploratory trajectories and loop closures, and that even for two heavy weighted loop closure candidates, the one with largest exploratory interest will be chosen as a goal.

## V. SIMULATIONS

We now compare the results of the two exploratory strategies with entropy decrease estimation (EDE) presented in this paper against frontier-based exploration [6]. Simulations are performed in two commonly used environments of varying size and complexity, the Cave and Freiburg maps [18].

For all simulations, robot motion was estimated with an odometric sensor with noise covariance  $\Sigma_u = \text{diag}(0.1 \text{ m}, 0.1 \text{ m}, 0.0026 \text{ rad})^2$ . The robot is fitted with a laser range finder sensor with a match area of  $\pm 1 \text{ m}$  in  $x$  and  $y$ , and  $\pm 0.35 \text{ rad}$  in orientation. That is, this is the maximum range in configuration space for which we can guarantee that a link between two poses can be established. 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 grid map

of the environment using the true robot path. The initial uncertainty of the robot pose was set to  $\Sigma_0 = \text{diag}(0.1 \text{ m}, 0.1 \text{ m}, 0.09 \text{ rad})^2$ . Informative loop closures were asserted at  $\mathcal{I} = 2.5 \text{ nats}$ .

The frontier-based exploratory method drives always the robot to the closest frontier larger than a threshold, without considering the localization and map uncertainties. In our simulation the frontiers bigger than 5 cells have been regarded first. When there are no frontiers of that size, this threshold is reduced to 1 cell. The trajectory for the frontier-based exploration method is planned with a RRT\*.

### A. Exploration in the cave map

The cave map is a simple scenario consisting of a single room resembling an industrial plant or a house room. In our simulations, the map was scaled to a resolution of  $20 \text{ m} \times 20 \text{ m}$ . To account for random effects of the sensor noise and of the RRT\* tree growth, each simulation was executed 5 times for each exploration strategy and limited to 200 simulation steps and 100 planning steps.

Frontier-based strategies do not consider the path entropy. This induces the accumulation of localization error, producing erroneous maps mostly around obstacles and frontiers. Planning over these maps complicates the finding of free trajectories to the goals, resulting in the large computation times for the frontier-based strategy as indicated in Table I. All experiments were carried out with a Quad core Intel Xeon system at 3.10GHz and with 8GB of memory.

The plots in Fig. 6 show one realization of the experiment for each strategy. The red dots and lines indicate the executed robot trajectories, the green lines indicate loop closures, and the black dots render occupancy using the complete path estimate. It can be observed for instance, how the frontier-based strategy results in many collisions since frontiers are mostly misclassified due to the larger path uncertainty. In contrast, EDE with gradient descent tends to produce valleys of high confidence. EDE with RRT\* clearly depicts the two different kind of goals, exploratory or revisiting.

Figure 7 shows how in average, map and path entropy evolution for the three strategies. For those methods that choose



	Frontier-based RRT*	EDE Potential fields	EDE RRT*
Computation time:	2814.14 s	601.02 s	487.15 s
Final map entropy:	127.43 nats	132.19 nats	133.20 nats
Final path entropy:	-1.28 nats	-1.73 nats	-2.07 nats
Loops closed:	15.4	22	17

TABLE I

AVERAGE COMPARISON OF EXPLORATION STRATEGIES IN THE CAVE MAP.

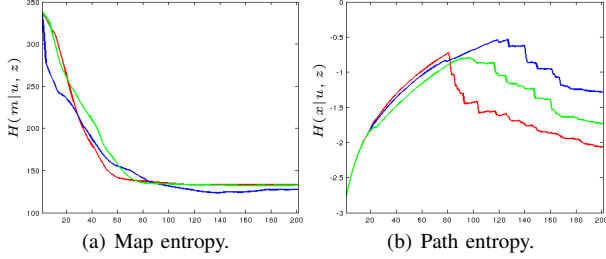


Fig. 7. Average entropies for the 5 simulation runs in the Cave map. In blue, the frontier-based with RRT\*; in green, gradient descent on the EDE field; in red, EDE with RRT\*.

goals using EDE, path entropy is significantly reduced and full coverage is reached. The frontier-based strategy closes loops only by chance, and does not improve path entropy, resulting in a final map of worse quality. Although all methods end up closing similar number of loops on average, the quality of the loops closed for the EDE strategies is significantly better.

### B. Exploration in the Freiburg map

The second environment analyzed is the Freiburg indoor building 079, of a significant larger complexity and size. Table II summarizes the average computation time as well as final entropy values and number of loops closed after 3 simulations. As with the previous scenario, the simulation steps limit is set to 200, whereas the planning steps limit is set to 100.

Both gradient descent EDE and frontier-based with RRT\* reach the planning steps limit before the simulation steps limit. This indicates how the problems cited previously become accentuated in a more complex and larger environment. The potential gradient descent method suffers of bottleneck at local minima and the frontier-based method easily gets trapped in a bad map construction due to the accumulated error since it does not plan revisiting actions.

The plots in Figure 8 show average map and path entropy values, respectively, for the 3 simulation runs, showing the reduction in map and path entropies for the three exploration strategies. Given the highly complex nature of this environment, EDE with gradient descent was the worst performing method in this case, getting trapped in local minima soon in the simulations. This is depicted by the flat green line in frame (a). EDE with RRT\* on the other hand outperformed the other two methods both with respect to map entropy reduction as well as accuracy in localization.

	Frontier-based RRT*	EDE Potential fields	EDE RRT*
Computation time	2505.44 s	2586.84 s	11951.18 s
Final map entropy:	395.48 nats	464.78 nats	390.22 nats
Final path entropy:	-1.17 nats	-1.38 nats	-1.75 nats
Loops closed:	12.5	3	12.5

TABLE II

AVERAGE COMPARISON OF EXPLORATION STRATEGIES IN THE FREIBURG BUILDING.

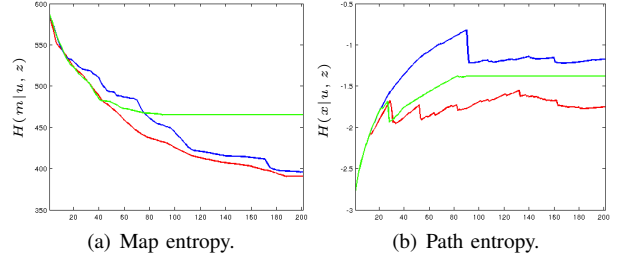


Fig. 8. Average entropies for the 3 simulation runs in Freiburg map. In blue, the frontier-based with RRT\*; in green, EDE with potential gradient descent; in red, EDE with RRT\*.

Figure 9 shows one realization of the rendered occupancy map for each of the three strategies on the Freiburg environment. The figure clearly depicts the various conclusions that had been already mentioned. Frontier-based exploration does not consider loop closing and thus produces maps with larger uncertainty, i.e., thicker grayish walls. Gradient descent on the EDE field ends up trapped in local minima. And, EDE with RRT\* produces larger and more accurate maps, but at significantly larger computation cost.

Figures 10 and 11 show intermediate steps in the computation of the EDE-RRT\* map. The first figure shows one instance of the computed PoseSLAM map. Note how the pose graph effectively covers the whole explored area with only a few most informative links between poses (green lines). The second figure shows an instance of the RRT\* map computed.

## VI. CONCLUSIONS

We have presented a method for the dense computation of joint map and path entropy decrease defined over the entire discretized C-space. 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.

Furthermore, two different exploration strategies are presented using this dense entropy decrease estimation. A gradient descent over an entropy decrease estimation field; and an RRT\* planner that chooses as goal the configuration of largest entropy decrease in the entire C-space.

Both methods outperform classical exploration methods that drive the robot to frontiers reaching similar coverage but producing significantly more accurate path estimates. The result is not only a path with less entropy, but a more accurate rendered occupancy map as well.

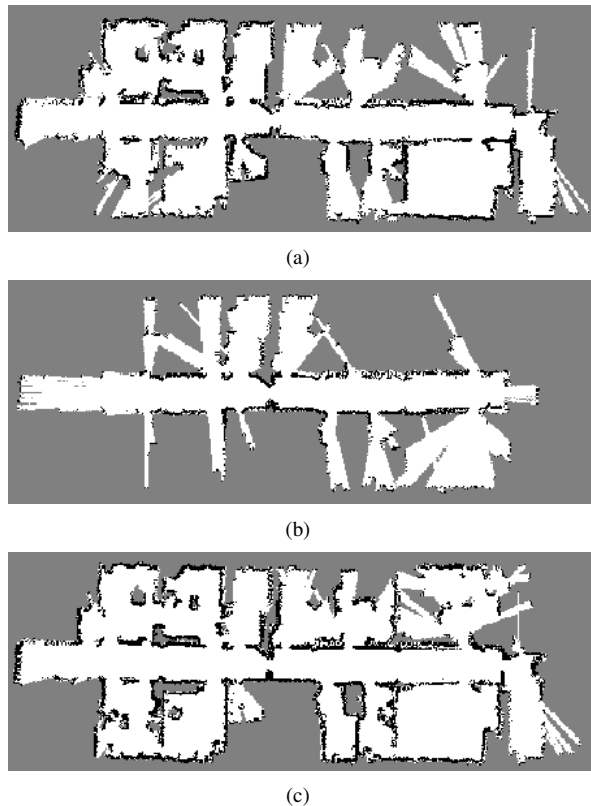


Fig. 9. Grid map constructed after a exploration simulation of 200 steps in the Freiburg environment. (a) Frontier-based with RRT\*. (b) EDE with potential gradient descent. (c) EDE with RRT\*.

One advantage of the gradient descent exploration method presented is that it produces exploratory trajectories along valleys of low path uncertainty, thus decreasing the probability of generating collisions. On the contrary, the technique gets stuck in local minima for moderately complex environments.

The entropy decrease RRT\* method outperforms all other solutions both in terms of map and path estimation accuracy and coverage.

The technique has been developed for planar environments and using laser range scanners as the sensing modality, but it can be easily extended to 3D and to accommodate other sensors. We leave this as future research.

## REFERENCES

- [1] H. J. S. Feder, J. J. Leonard, and C. M. Smith, "Adaptive mobile robot navigation and mapping," *Int. J. Robotics Res.*, vol. 18, pp. 650–668, 1999.
- [2] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte, "Information based adaptive robotic exploration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lausanne, Oct. 2002, pp. 540–545.
- [3] 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, Dec. 2010.
- [4] L. Torabi, M. Kazemi, and K. Gupta, "Configuration space based efficient view planning and exploration with occupancy grids," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, Nov. 2007, pp. 2827–2832.
- [5] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *Robotics: Science and Systems I*, Cambridge, Jun. 2005, pp. 65–72.
- [6] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *IEEE Int. Sym. Computational Intell. Robot. Automat.*, Monterrey, 1997, pp. 146–151.
- [7] R. Valencia, J. Valls Miró, G. Dissanayake, and J. Andrade-Cetto, "Active Pose SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vilamoura, Oct. 2012, pp. 1885–1891.
- [8] J. Vallvé and J. Andrade-Cetto, "Mobile robot exploration with potential information fields," in *Proc. Eur. Conf. Mobile Robots*, Barcelona, Sep. 2013, pp. 222–227.
- [9] S. Karaman, M. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT\*," in *Proc. IEEE Int. Conf. Robotics Autom.*, Shanghai, May 2011, pp. 1478–1483.
- [10] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact Pose SLAM," *IEEE Trans. Robotics*, vol. 26, no. 1, pp. 78–93, Feb. 2010.
- [11] R. Valencia, E. Teniente, E. Trulls, and J. Andrade-Cetto, "3D mapping for urban service robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Saint Louis, Oct. 2009, pp. 3076–3081.
- [12] M. Jadidi, R. Valencia, J. Valls-Miro, J. Andrade-Cetto, and G. Dissanayake, "Exploration in information distribution maps," in *Proc. RSS Workshop Robotic Explor., Monit., Inf. Collect.*, Berlin, Jun. 2013.
- [13] R. Valencia, M. Morta, J. Andrade-Cetto, and J. Porta, "Planning reliable paths with Pose SLAM," *IEEE Trans. Robotics*, vol. 29, no. 4, pp. 1050–1059, 2013.
- [14] R. Shade and P. Newman, "Choosing where to go: Complete 3D exploration with stereo," in *Proc. IEEE Int. Conf. Robotics Autom.*, Shanghai, May 2011, pp. 2806–2811.
- [15] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Int. Conf. Autonomous Agents*, Minneapolis, 1998, pp. 47–53.
- [16] E. Teniente and J. Andrade-Cetto, "HRA\*: Hybrid randomized path planning for complex 3D environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Nov. 2013, pp. 1766–1771.
- [17] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robotics Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [18] A. Howard and N. Roy, "The robotics data set repository (Radish)," <http://radish.sourceforge.net>, 2003.

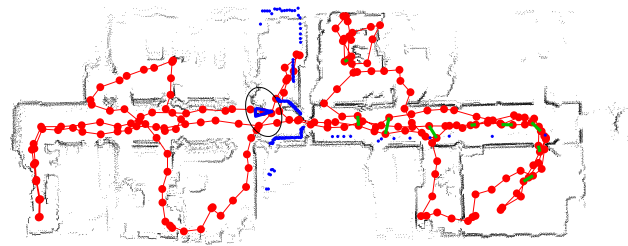


Fig. 10. EDE-RRT\* PoseSLAM graph. The red dots and lines indicate the robot path. The blue dots represent the current sensor reading, and the blue triangle and the hyper-ellipsoid indicate the current robot pose estimate. The occupancy map is rendered from sensor data at the mean of that estimate, and is shown in light gray.

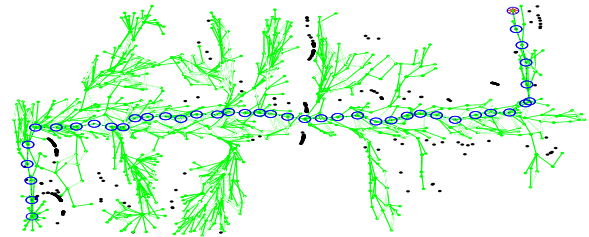


Fig. 11. EDE-RRT\* tree. This is an instance of the RRT\* tree and the computed path to the goal. The black dots indicate failed leaf extensions due to collision.