

Active Pose SLAM with RRT*

Joan Vallvé and Juan Andrade-Cetto

Abstract— We propose a novel method for robotic exploration that evaluates paths that minimize both the joint path and map entropy per meter traveled. The method uses Pose SLAM to update the path estimate, and grows an RRT* tree to generate the set of candidate paths. This action selection mechanism contrasts with previous approaches in which the action set was built heuristically from a sparse set of candidate actions. The technique favorably compares against the classical frontier-based exploration and other Active Pose SLAM methods in simulations in a common publicly available dataset.

I. INTRODUCTION

A representation of the environment is essential for several problems in mobile robotics such as localization, motion planning and autonomous navigation. Some of these methods build a map while at the same time compute an estimate on the robot localization. This is commonly referred as SLAM [1]–[5]. However, SLAM methods often do not take any decision on the robot trajectory.

Other methods to build maps tackle instead the problem of exploration, which entails finding a path that maximizes the knowledge about the environment. Most exploration techniques [6], [7] do not take into account the uncertainties in robot motion and sensing and limit themselves to pursue full coverage.

Conversely, active localization methods [8], [9] do not produce a map, but instead drive the robot trying to minimize the localization uncertainty in previously mapped environments.

We refer to the problem of simultaneously localizing the robot, mapping the environment, and also planning for the path that improves both the localization and the map estimates as Active SLAM, and we consider it as one of the most challenging problems of mobile robotics. The Active SLAM problem has been approached in the past as an action selection problem [10], [11], and entails two issues, the computation of a utility or cost function to evaluate the effect of each candidate action, and the process of generating such action set. While the evaluation of actions is critical since it determines which actions are finally executed, the action set generation is also important since a good evaluation is useless if the action set is sparse or naively built.

As the title of the paper unveils, our method uses Pose SLAM which only estimates the robot trajectory [12]. To build a map, an occupancy grid can be rendered at any time from the trajectory means and their corresponding sensor

This work has been supported by the Spanish Ministry of Economy and Competitiveness under project PAU+ 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.



(a) $dRRT^*$



(b) $eRRT^*$

Fig. 1. Two examples of the resulting path candidates using the two different cost functions proposed. Color of branches represent the value of the information gain per meter travelled (Eq.1) of the path that ends in the corresponding node. Red paths indicate better candidates. In gray, the Pose SLAM nodes.

observations. The method discussed in this paper minimizes the joint map and path entropy as in our previous work [13], but additionally it also minimizes the distance traveled. In contrast to previous approaches, an efficient method to estimate the joint entropy change for several action candidates as computed by the RRT* algorithm is also presented.

II. RELATED WORK

Active SLAM has been approached in the past as the minimization of two independent terms, the map entropy and the path entropy. Feder et al. [14], propose a metric to evaluate uncertainty reduction as the sum of these two independent entropies, but only for the last robot pose and a limited set of landmarks with a one step exploration horizon. Bourgault et al. [15] alternatively propose a utility function that computes independently the potential reduction of vehicle localization uncertainty from a feature-based map, and the information gained over an occupancy grid. To consider joint map and path entropy reduction, Vidal et al. [16] tackled the issue taking into account robot and map cross correlations for the Visual SLAM EKF case.

Torabi et al. [17] jointly computed the entropy reduction directly in configuration space but for a limited number of

configurations. Our previous work [13], [18] computes joint map and path entropy decrease estimates densely for the entire configuration space finely discretized. But both, path and map entropy decreases depend on the path taken to arrive to each configuration, and our entropy decrease estimates did not take into account the changes in entropy induced during the path and only the estimated change at the final robot configuration.

Action selection in Active SLAM has been approached in the past as the analysis of the effect of a small heuristically chosen set of path candidates with respect to various entropy-based utility functions [10], [11]. Action candidates can be paths to different goals containing revisiting configurations or exploring frontier-based poses. But the optimal path in terms of entropy reduction may not be included within these few candidates. Moreover, the path itself may be as important for entropy reduction as the goal itself. Not taking into account the effect of the whole path implies renouncing to optimality with respect to the chosen utility function.

The adaptation of probabilistic planning methods to the exploration problem was introduced by Oriolo et al. [19]. In their approach called SRT, the robot executed motion commands to a random pose sampled inside the newly observed area that was not observed in previous nodes. Posteriorly, Freda et al. presented a frontier-based SRT [20] which randomly selected the frontier centroids of the last observation as goals. However, both approaches directly executed the first random sample without any optimality evaluation.

Karaman et al. [21] presented the asymptotically optimal planner RRT* designed for path planning given a goal and using cost functions that satisfy the triangular inequality. The expansive nature of the RRT* algorithm suggests that it can be useful in the action set generation for Active Pose SLAM. In this paper we benefit from the RRT* tree expansion strategies and suggest the use of the same cost function for tree expansion as for action selection.

In the following section we describe succinctly what Pose SLAM is, and refer the reader to the original paper for a full understanding for sake of space [12]. Then, in section IV we explain our proposed optimization for Active Pose SLAM and its efficient implementation, whereas in section V we detail how we use the RRT* for the action set generation. The last two sections include simulations and results and the conclusions.

III. POSE SLAM

Pose SLAM [12] maintains a probabilistic estimate of the robot pose history as a sparse graph with a canonical parametrization $p(x) = \mathcal{N}^{-1}(\eta, \Lambda)$, (with information vector η and information matrix Λ). This parametrization has the advantage of being exactly sparse [22] compared to the traditional Kalman form (mean μ and covariance Σ). State transitions result from the composition of motion commands u to previous poses, and the registration of sensory data z also produces relative motion constraints 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 in the filter itself. Instead, it can be synthesized at any instance in time using the pose means in the graph μ and the raw sensor data z . We use the method presented in [18] that stores the observations in locally referenced images and updates the occupancy map with new observations by aggregating them rotated and translated to their corresponding pose means. After a loop closure, all local coordinate images are once again rotated and translated to their new pose means to synthesize a new occupancy map.

IV. EVALUATION OF ACTIONS

We refer to specific parts of the state x using subindexes. So for instance, after executing a set of motions $u_{1:t}$ we end up with a trajectory $x_{1:t}$, which Pose SLAM estimates. At this point we can render a map m_t from its means $\mu_{1:t}$. Now consider the exploration problem, an action (or path) candidate a_t is defined as a sequence of relative motions $u_{t+1:T}$, which would produce a sequence of new robot configurations $x_{t+1:T}$, from where tentative measurements $z_{t+1:T}$ would be made, and an expected map m_T could be rendered.

Continuing the work previously done in our group [11], we approach exploration as a joint entropy minimization problem. As seen in [18], the map entropy has a strong relationship with coverage, whilst path entropy has some relationship with the map quality since a better localization produces a better map. In this work we add to the equation the distance traveled, so that minimization is not independent of the path length. In this way we can compare paths of different lengths when searching for the optimal entropy reduction action set and also minimize the distance traveled. Consequently, we propose the minimization of the joint entropy change divided by the distance of the path candidate, maximizing information gain per meter traveled

$$a_t^* = \operatorname{argmin} \frac{H(x_{1:T}, m_T | u_{1:T}, z_{1:T}) - H(x_{1:t}, m_t | u_{1:t}, z_{1:t})}{\operatorname{dist}(u_{t+1:T})} \quad (1)$$

The entropy terms in this equation take the form

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

As in [18], we approximate this function with

$$H(x, m | u, z) \approx H(x | u, z) + \alpha(p(x | u, z)) H(m | \mu, z) \quad (2)$$

where instead of averaging over the entropy of the resulting map for each of the trajectories in the probability distribution, we compute it only for the map rendered from the mean trajectory estimate μ . And, different from the approximation used by Valencia et al. [11], we add the factor $\alpha(p(x | u, z))$.

This factor has an intuitive meaning, 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 may spoil the map adding bad cell classifications, i.e., adding entropy. Our approach is to weight the entire entropy reduction map with the inverse of the determinant of the marginal covariance at the current configuration $\alpha(p(x|u, z)) = |\Sigma_{tt}|^{-1}$. Exploratory trajectories that depart from uncertain configurations will be weighted giving predominance to the path entropy reduction term and vice versa. In this way, we modulate the importance of the exploratory and relocalization behavior as proven in [18], improving the strong sensibility with respect to the motion noise.

Then, at the planning step t the current joint entropy is directly obtained from Eq. 2. The following is an exhaustive description of how we efficiently compute the other term in Eq. 1, i.e. the hypothetical joint entropy estimate for any action candidate $H(x_{1:T}, m_T | u_{1:T}, z_{1:T})$.

A. Path entropy estimation

In order to evaluate the effect of a path candidate a_t , we must estimate the path entropy after executing each motion command u_k and obtaining the measurements z_k for $k \in [t+1, T]$. If we consider path entropy as the entropy of the multivariate Gaussian $x_{1:k}$, we might end up dealing with an ill defined covariance matrix $\Sigma_{1:k, 1:k}$ as explained in [10]. Hence, we opt for the same approximation also used in [11], [13], [18], which averages over all individual pose marginals

$$H(x_{1:k} | u_{1:k}, z_{1:k}) \approx \frac{1}{k} \sum_{i=1}^k \ln \left((2\pi e)^{\frac{n}{2}} |\Sigma_{ii}| \right), \quad (3)$$

being n the dimension of the individual pose vector, $n = 3$ in our case.

One alternative would be to simulate a Pose SLAM filter for each path candidate and to evaluate Eq. 3. Since we want to evaluate several path candidates, this method would not be efficient at all. Instead, we compute an iterative estimate of the path entropy change efficiently in open loop and after a loop closure as follows.

1) *Path entropy estimation in open loop:* In open loop, a new node is added to the filter without changing the rest of the marginal covariances. Then, the path entropy at the k th time step from Eq. 3 becomes

$$H(x_{1:k} | u_{1:k}, z_{1:k}) \approx \frac{k-1}{k} H(x_{1:k-1} | u_{1:k-1}, z_{1:k-1}) + \frac{1}{k} \ln \left((2\pi e)^{\frac{n}{2}} |\Sigma_{kk}| \right). \quad (4)$$

So we compute an estimate of the path entropy in open loop from the entropy estimate at the previous pose and the current marginal covariance Σ_{kk} which can be effectively computed by linearly propagating the previous one using the motion Jacobians as in Pose SLAM.

2) *Path entropy estimation at loop closure:* In Pose SLAM, the parameter match area is defined as the intervals in x , y and θ where loops can be closed with a given sensor. When a candidate robot configuration x_k falls inside the match area of any pose within the Pose SLAM estimate x_l , $l \in [1, t]$, the observation z_k may produce a loop closure, hence a state update. A state update entails changes in the state estimate, and new marginal covariances need to be computed for the estimation of the path entropy decrease to evaluate Eq. 3.

To compute these marginals, the natural step would be to simulate the Pose SLAM filter and to recover the marginal covariances from the new estimate. As said before, doing so for several path candidates would be computationally expensive. We could instead use the information gain from closing such loop as defined in [12]. However, that technique assumes that we are estimating entropy for the whole multivariate Gaussian, and as we said before this is an ill-posed problem. Instead we approximate this value as follows.

After a loop closure, all marginal covariances Σ_{ii} ($\forall i \in [1, k]$) change to new values Σ'_{ii} , so the path entropy change is, following Eq. 3,

$$\Delta H(x_{1:k} | u_{1:k}, z_{1:k}) = \frac{1}{k} \sum_{i=1}^k \ln \frac{|\Sigma'_{ii}|}{|\Sigma_{ii}|} = \frac{1}{k} \ln \prod_{i=1}^k \rho_i. \quad (5)$$

The marginal covariance determinant ratios $\rho_i = |\Sigma'_{ii}|/|\Sigma_{ii}|$ determine the path entropy change after a loop closure. We approximate them with two assumptions. First, we assume ‘‘clean’’ loops, meaning that no node in the loop has more than two connections. Assuming that, a loop closure to the l th node will affect only the marginal covariances included in the loop, so $\rho_i = 1, \forall i < l$ (eq. ??). Knowing the loop closure sensor noise, the new loop nodes marginal covariances Σ'_{ll} and Σ'_{kk} can easily be computed so also their respective determinant change ratios ρ_l and ρ_k . Secondly, we approximate the rest of the determinant change ratios linearly:

$$\begin{aligned} \Delta H(x_{1:k} | u_{1:k}, z_{1:k}) &\approx \frac{1}{k} \ln \prod_{i=l}^k \rho_i \\ &\approx \frac{1}{k} \ln \prod_{j=1}^{k-l+1} \left(\rho_l + \frac{\rho_k - \rho_l}{k-l+1} j \right). \end{aligned} \quad (7)$$

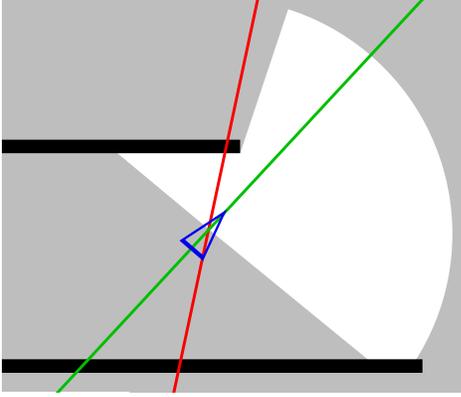
B. Map entropy estimation

The entropy of the occupancy map m_T with cell size s can be computed as the scalar sum

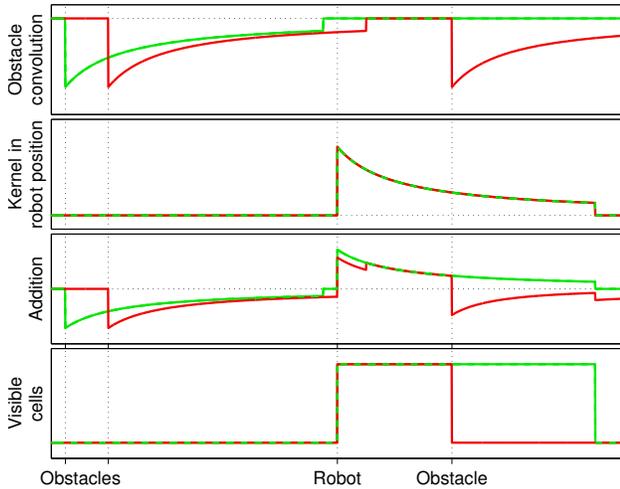
$$H(m_T | \mu_{1:T}, z_{1:T}) = -s^2 \sum_{c \in m_T} (p_c \ln p_c + (1-p_c) \ln(1-p_c)), \quad (8)$$

where p_c is the classification probability of cell c . The reduction in entropy that is attained after moving to new locations and sensing new data depends on the number of cells that will change its classification probability, i.e. will be discovered either obstacle or free.

Each intermediate pose x_k of each action candidate will produce a different observation of the environment z_k , so



(a) For a robot configuration example, the expected visible cells (white) taking into account the known obstacles (black). In red and green the ray directions detailed in the frame below.



(b) Example of convolution results in the two specific ray directions depicted in the above frame in their respective colors.

Fig. 2. Example of ray casting for a particular robot configuration.

different cells will be classified producing different map entropy changes. For each action candidate, the map entropy change will depend on the number of cells discovered in all observations made during the action a_t . In order not to overcount cells discovered by different poses of the same path, we do not need a method for computing how many cells but which ones will be discovered from any robot pose x_k . Then, for the sequence of poses of a candidate path a_t , we will be able to estimate the accumulated number of discovered cells.

Extending the work in [13], [18], we developed an efficient method to compute ray casting from several robot configurations over the known environment making use of convolutions and pre-computing most of the process.

Given a robot configuration candidate x_k , for each ray direction within the sensor spread we compute which cells are between the sensor and the nearest obstacle (if there

is any) and not further than the maximum simulated sensor range (see Fig. 2.a). We use a convolution using a kernel of zeros except for the cells in a specific direction which have exponentially decreasing values. For each ray direction, convolving this kernel with an image of ones at the obstacles, and subtracting it from the kernel centered at the robot position, the discovered cells will be those with positive values (see Fig. 2.b).

This operation must be computed for each ray direction included in the sensor spread and for all robot configurations of all action candidates. However, we can pre-compute most of it. Firstly, the kernel for each of all 360° discretized directions is computed once at the beginning of the algorithm. Secondly, at each planning step, the convolution over the obstacles can be precomputed for all discretized directions. Then, for a robot configuration candidate x_k , we only need to translate all direction kernels, subtract the convoluted obstacles and binarize. Then we accumulate the discovered cells from the directions within the sensor spread corresponding to the robot orientation (Fig. 2.a). Finally, accumulating the discovered cells in all robot configurations $x_{t+1:T}$, we can estimate its map entropy change using Eq. 7.

Tuning as a parameter the sensor maximum range, i.e. the kernel size, we can be more conservative or optimistic in the estimation of cells discovered. Using the same value as the real sensor, we will assume that all the unknown cells are free whereas when using a lower value, we will assume the existence of obstacles. This value depends on the environment morphology and is fixed experimentally.

V. RRT*: GENERATING AND EVALUATING THE ACTION SET

The RRT* algorithm presented by Karaman et al. [21] is an asymptotically optimal version of the rapidly-exploring random trees by Lavelle et al. [23]. It was designed to find an optimal path from the current robot pose to a given goal in a known environment.

The RRT has a simple *extend* routine: a new node in the RRT is added in the direction of a random sample from the nearest node with a distance ν . The RRT* *extend* routine is more complex. After creating a new node as in the RRT, it is not automatically connected to the tree from its predecessor. Instead, it is connected to one of its neighbors according to a given cost function. After that, is checked if the new node can yield a lower cost to any of each near neighbors and it is rewired if it is the case. This two actions allow the RRT* be asymptotically optimal.

The expansive nature of the RRT and RRT* algorithms suggests that they can be useful to generate the action set in Active Pose SLAM since they provides several collision free paths from the current robot pose to several free configurations. Knowing that, we propose to grow an RRT* in the known and free environment and then take each one of the RRT* nodes and the path to reach it as an action candidate a_t . After the tree is grown, the best action candidate a_t^* is chosen from all the paths to each node according to the evaluation introduced in section IV.

	Frontier-based	Heuristic Active Pose SLAM	EDE	d RRT* Active Pose SLAM	e RRT* Active Pose SLAM
Final map entropy	527.92 nats	670.93 nats	584.23 nats	526.58 nats	558.80 nats
Final path entropy	-5.06 nats	-10.11 nats	-6.55 nats	-6.34 nats	-5.57 nats
Total time	1089.15 s	78997.9 s	1083.70 s	8752.13 s	59785.42 s
Loops closed	22.6	43.2	25.4	18.8	34.4
Coverage	741.32 m ²	514.10 m ²	660.86 m ²	749.96 m ²	707.10 m ²
Map error	130.80 m ²	87.84 m ²	106.34 m ²	126.56 m ²	112.22 m ²

TABLE I
AVERAGE COMPARISON OF THE PERFORMANCE OF SEVERAL EXPLORATION METHODS IN THE FREIBURG 079 MAP.

The cost function used in RRT* *extend* routine will affect the resulting paths of the tree. Using the distance as cost function, as usual, the algorithm will provide the shortest paths to several free configurations. However, if we use other cost function the tree will grow and rewire differently and the resulting RRT* paths will not be optimal in distance traveled but in that cost function. So, we propose two action set generation alternatives using the RRT*. The first is using the distance as the cost function (d RRT*) and the second one is to use the entropy change divided by the distance (e RRT*), which is the cost function defined in Eq. 1.

In RRT*, once a solution path is found, the extension process is continued for a time in order to keep improving this solution. Some heuristics can be set to end the extension once the solution have been sufficiently improved. In our case, we do not have a specific goal to reach with the RRT*. Instead, our objective is to have several paths to several free configurations spread over the free discovered environment and we want these paths to be optimal in terms of the cost function used. We stop extending after certain amount of nodes per free square meter is reached. This value is fixed experimentally since it depends on the morphology of the environment.

The evaluation of the cost function is computed using the method explained in section IV during the extension of e RRT*, and once the extension processes is stopped for d RRT*. Figure 1 shows the resulting action set of d RRT* and e RRT*. The color corresponds to the evaluation of the cost function of each path. Both frames share the same color scale, the more red, the better the path. It can be seen how e RRT* branches are better than the d RRT* since the rewiring occurred according to the action selection cost function as well.

VI. SIMULATIONS

We now compare the performance of the two RRT* variants of the presented Active Pose SLAM method against three other exploration approaches. The first method is the typical frontier-based exploration [6]. This method drives the robot to the closest frontier larger than a threshold (90 cm in our case), without considering the localization and map uncertainties. When there are no frontiers of that size, this threshold is reduced progressively. The path to the selected frontier centroid is planned using the RRT* planner.

The second method to which we compare is Active Pose SLAM by Valencia et al. [11], which evaluates three heuristically generated action candidates including one re-visiting path and the two closest frontiers. And finally, the third method evaluated is our entropy decrease estimation (EDE) method [13] that precomputes an estimate of the joint entropy decrease at each robot pose of the discretized configuration space and plans a path to the most informative robot configuration using RRT*. In contrast to our approach, this method does not take into account the change in entropy during path traversal for each explored path, but only the one occurring at the end of the paths.

Five simulations are performed in the commonly used Freiburg 079 map [24]. In all of them, robot motion was estimated with an odometric sensor with a noise covariance factor of 15%. The robot is fitted with a laser range finder sensor with a match area of ± 1 m in x and y , and ± 0.35 rad in orientation. 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 with noise covariance 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, and corrupted with similar values of Gaussian measurement noise. The initial uncertainty of the robot pose was set to $\Sigma_0 = \text{diag}(0.1 \text{ m}, 0.1 \text{ m}, 0.09 \text{ rad})^2$. Pose SLAM asserted loops more informative than 2.5 nats.

A number of different metrics were used to compare the performance of the five methods with respect to the distance traveled. We stored average values for the 5 runs of path and map entropy for each of the methods; the average map coverage, measured as the number of cells labeled in the occupancy map; and the average 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. Two other measures of performance computed were total execution time, including all the different processes of each method except for the map rendering, and the total number of loop closures computed by each of the methods. Table I shows the final average values of each metric for each method. The average evolution of the map and path entropies along the traveled distance for all methods can be observed

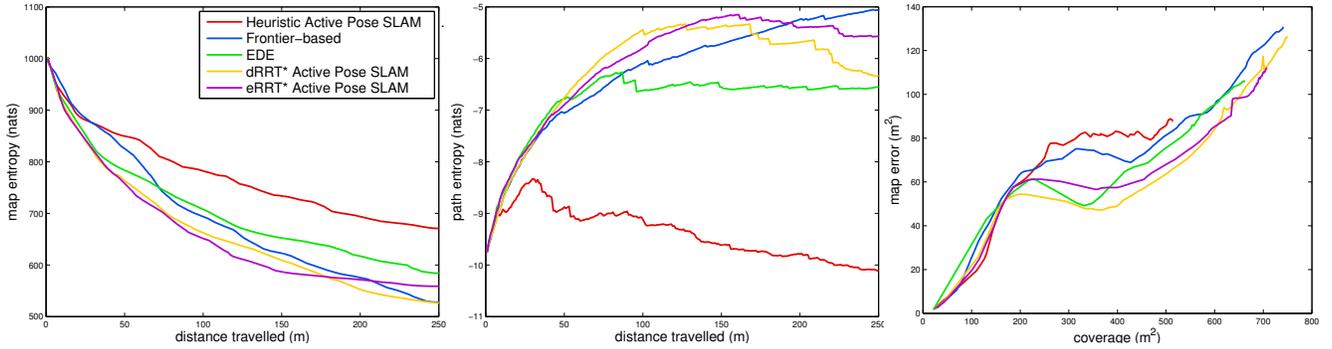


Fig. 3. Average results for the five simulation runs in Freiburg 079 map.

in the left and center frames of Fig. 3. The map error with respect to coverage is plotted in the right frame of the same figure.

The computational cost of *eRRT** is significantly larger than *dRRT** because the cost function evaluation is computed several times for each node in the *eRRT** extension process due to the rewiring, whereas in the *dRRT** it is only computed once for each node. This is a time consuming step because the cost function requires also the estimation of the joint path and map entropy. Nonetheless, both approaches are less computationally expensive than the use of the Heuristic Active Pose SLAM method. The frontier-based and the EDE methods are much faster but they do not evaluate the effect of the paths with regards to entropy reduction, but only seek to minimize distance to precomputed goals.

The frontier-based method, reaches a low level of map entropy but with a high path entropy value, and thus large map error, since it only pursues to maximize coverage. Active Pose SLAM performance is characterized by a very conservative behavior with regards to localization uncertainty because of the absence of the factor $\alpha(p(x|u, z))$ in its entropy approximation function (see Eq. 2). For the same plant and sensor noise levels, the technique weights more localization than exploration and hence coverage grows much slower than in the other methods.

While the EDE method ends the simulations with lower path entropy values, the two Active Pose SLAM methods proposed, *dRRT** and *eRRT**, are better in terms of map entropy on average. At the final part of the simulations, both methods have lower levels of map error for the same coverage levels than the rest (Fig. 3.c). Moreover, we see how the *dRRT** reaches higher coverage in the same distance traveled while the *eRRT** improves over map error. The performance of *eRRT** is slightly better than *dRRT** with regards to map error, and significantly better than the heuristic Active Pose SLAM with regards to both coverage and error. However, entropy evaluation is time consuming in *RRT** extension, thus an action set generation that only takes into account distance traveled as in *dRRT** is an adequate compromise.

Next, we analyze the effect of loop closures in the final exploration results. For instance, the frontier-based method finalizes with higher path entropy values on average than

EDE and *dRRT**, even after closing a similar amount of loops. This is because the loops closed by frontier-based were not optimally chosen to reduce uncertainty, but rather closed by chance. Obviously, the conservative behavior of heuristic Active Pose SLAM results in a large amount of loop closures in average. The *eRRT** also closed a large amount of loops in average because using the entropy based cost function, the *RRT** generates paths that include such loop closure poses.

Figure 4 shows single simulation runs for the five methods. In the first frame, we can observe the final localization error of the frontier-based exploration with the last sensor observation in blue. The heuristic Active Pose SLAM final graph is largely connected and all the trajectories remained near the initial robot pose, leaving the rest of the scene largely unexplored, as can be observed in Fig. 4 b. The EDE and *dRRT** final graphs (frames c and d) contain straight paths due to the distance cost function used in the *RRT**. Conversely, the *eRRT** trajectory presents neither straight paths to the goal, nor strong loop closing trajectories, but rather a combination of the two for which the cost function is minimal.

VII. CONCLUSIONS

We presented an Active Pose SLAM approach that maximizes coverage while maintaining accurate map and path estimates. Actions are evaluated in terms of the joint entropy change per distance travelled. Our method optimally generates a set of path candidates to explore using *RRT**.

The method has been tested with two strategies to generate the action set: growing an *RRT** with the typical Euclidean distance as cost function, and doing so with entropy change per meter traveled. The results show that the combined strategy of evaluating path candidates using entropy minimization per traveled meter, but generating the action candidates minimizing Euclidean distance provides the best compromise between execution time, coverage, and map and path errors.

Further comparisons with other datasets and on real time experiments are left as immediate actions for further research.

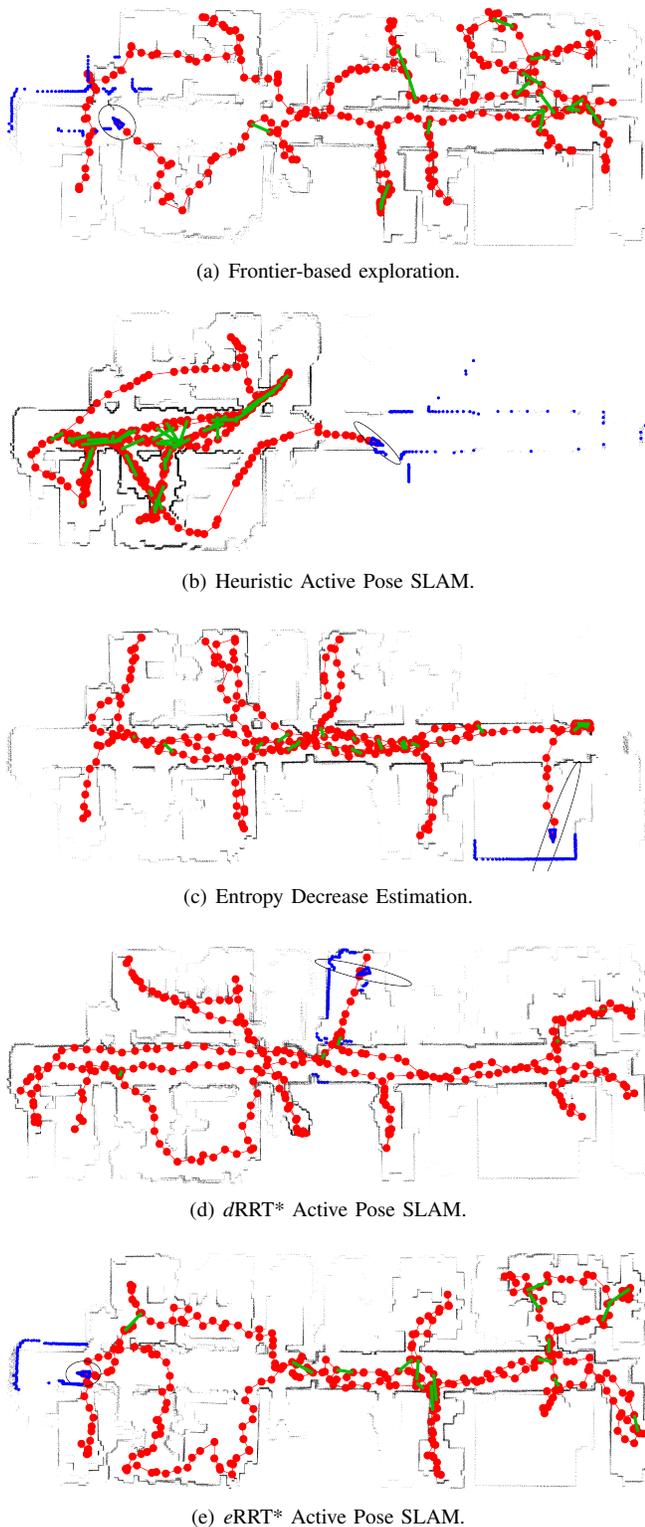


Fig. 4. Final trajectories after a 250m exploration simulation of the Freiburg 079 map for all methods compared. In red the path estimate, in green loop closure links, in black the whole raw sensor data rendered at the path estimate, and in blue the marginal robot pose estimate for the current state (mean and variance) along with the sensed data at that location.

REFERENCES

- [1] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robotics Autom.*, vol. 17, no. 3, pp. 229–241, 2001.
- [2] M. Montemerlo and S. Thrun, *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*, ser. Springer Tracts in Advanced Robotics. Springer, 2007, vol. 27.
- [3] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. J. Robotics Res.*, vol. 23, no. 7-8, pp. 693–716, 2004.
- [4] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.
- [5] S. Williams, V. Indelman, M. Kaess, R. Roberts, J. Leonard, and F. Dellaert, "Concurrent filtering and smoothing: A parallel architecture for real-time navigation and full smoothing," *Int. J. Robotics Res.*, vol. 33, pp. 1544–1568, 2014.
- [6] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *IEEE Int. Sym. Computational Intell. Robot. Automat.*, Monterey, 1997, pp. 146–151.
- [7] 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.
- [8] A. Corominas-Murtra, J. Mirats-Tur, and A. Sanfeliu, "Action evaluation for mobile robot global localization in cooperative environments," *Robotics Auton. Syst.*, vol. 56, no. 10, pp. 807–818, 2008.
- [9] D. Fox, W. Burgard, and S. Thrun, "Active Markov localization for mobile robots," *Robotics Auton. Syst.*, vol. 25, no. 3-4, pp. 195–207, 1998.
- [10] 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.
- [11] 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.
- [12] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact Pose SLAM," *IEEE Trans. Robotics*, vol. 26, no. 1, pp. 78–93, 2010.
- [13] J. Vallvé and J. Andrade-Cetto, "Dense entropy decrease estimation for mobile robot exploration," in *Proc. IEEE Int. Conf. Robotics Autom.*, Hong Kong, May 2014, pp. 6083–6089.
- [14] 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.
- [15] 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.
- [16] T. Vidal-Calleja, A. Sanfeliu, and J. Andrade-Cetto, "Action selection for single camera SLAM," *IEEE Trans. Syst., Man, Cybern. B*, vol. 40, no. 6, pp. 1567–1581, 2010.
- [17] 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.
- [18] J. Vallvé and J. Andrade-Cetto, "Potential information fields for mobile robot exploration," *Robotics Auton. Syst.*, 2015, in press.
- [19] G. Oriolo, M. Vendittelli, L. Freda, and G. Troso, "The SRT method: Randomized strategies for exploration," in *Proc. IEEE Int. Conf. Robotics Autom.*, New Orleans, Apr. 2004, pp. 4688–4694.
- [20] L. Freda and G. Oriolo, "Frontier-based probabilistic strategies for sensor-based exploration," in *Proc. IEEE Int. Conf. Robotics Autom.*, Barcelona, Apr. 2005, pp. 3881–3887.
- [21] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robotics Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [22] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Trans. Robotics*, vol. 22, no. 6, pp. 1100–1114, 2006.
- [23] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Comp. Sc. Dept., Iowa St. Univ.*, Tech. Rep. 98-11, 1998.
- [24] A. Howard and N. Roy, "The robotics data set repository (Radish)," <http://radish.sourceforge.net>, 2003.