

Active SLAM Utility Function Exploiting Path Entropy

Muhammad Farhan Ahmed

LS2N

Ecole Centrale de Nantes (ECN)

Nantes, France

Muhammad.Ahmed@ec-nantes.fr

Vincent Frémont

LS2N

Ecole Centrale de Nantes (ECN)

Nantes, France

vincent.fremont@ec-nantes.fr

Isabelle Fantoni

LS2N

CNRS

Nantes, France

isabelle.fantoni@ls2n.fr

Abstract—In this article we present a utility function for Active SLAM (A-SLAM) which utilizes map entropy along with D-Optimality criterion metrics for weighting goal frontier candidates. We propose a utility function for frontier goal selection that exploits the occupancy grid map by utilizing the path entropy and favors unknown map locations for maximum area coverage while maintaining a low localization and mapping uncertainties. We quantify the efficiency of our method using various graph connectivity matrices and map efficiency indexes for an environment exploration task. Using simulation and experimental results against similar approaches we achieve an average of 32% more coverage using publicly available data sets.

Index Terms—Active SLAM, Mapping, Information Theory, Entropy.

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) encompasses a range of methods used by robots to determine their own position while simultaneously creating a map of their surroundings during navigation. Most SLAM algorithms are considered passive, wherein the robot is moving freely, and the navigation or path-planning algorithm plays no active role in directing the robot's movements or trajectory. In contrast, A-SLAM aims to address the optimal exploration of unknown environments by proposing a navigation strategy that generates future target positions and actions. These actions are designed to reduce uncertainty in the map and the robot's pose, enabling a fully autonomous SLAM system.

In A-SLAM, the robot's exploration process begins by initially identifying potential target positions within its current map estimate. Once the robot has established a map of its surroundings, it proceeds to locate positions worth exploring. One commonly used method for this task is frontier-based exploration, initially introduced by [1]. Frontier-based exploration defines the 'frontier' as the boundary separating known map locations from unknown ones, as observed by the robot's sensors. After identifying these goal frontiers, the robot computes a cost or utility function. This function

*This work was carried out in the framework of the NExT Senior Talent Chair DeepCoSLAM, which was funded by the French Government, through the program Investments for the Future managed by the National Agency for Research ANR-16-IDEX-0007, and with the support of Région Pays de la Loire and Nantes Métropole.

relies on the potential reward associated with selecting the optimal action from a set of all possible actions. In an ideal scenario, this utility function should account for the complete joint probability distribution of both the map and the robot's poses. To quantify this uncertainty, we turn to two well-established domains: Information Theory (IT), and the Theory of Optimal Experimental Design (TOED), as detailed in [3]. The subsequent crucial step involves executing the optimal action, guiding the robot towards its goal position using path planning techniques.

In this article, we propose a utility function for selecting the goal frontier candidate for autonomous exploration by the robot. Our function takes into account the amount of uncertainty in the map measured as path entropy and Euclidean distance to each frontier candidate. We add this utility function to the one of [15] which selects the frontier weighted by the D-optimality criterion as the maximum number of spanning trees (in pose-graph) towards it. Using the proposed utility function brings the advantage that it incorporates not only the SLAM uncertainty but also entropy reduction in the environment to guide the robot to promising unknown areas for the exploration task.

This article is organized as follows: Section II, provides an insight into the related work. Section III gives preliminary knowledge about the structure of modern graph SLAM, A-SLAM, and how uncertainty is measured and related to graph connectivity. In Section IV we present the approach and formulation of our proposed method. Section V presents and discusses our simulation results. Finally, we conclude in Section VII summarizing our contributions and motivating future research aspects.

II. RELATED WORK

As discussed above, A-SLAM involves frontier detection, utility computation by quantifying and minimizing the uncertainty, and finally generating the action for robot navigation.

The method proposed by [5] formulates a hybrid control switching exploration method. Within the occupancy grid map, each frontier is segmented, a trajectory is planned for each segment, and the trajectory with the highest map segment covariance is selected from the global cost map, which renders this method computationally expensive and limited to static

obstacles. Meanwhile, the approach in [6] deals with dynamic environments with multiple ground robots and uses frontier exploration for autonomous exploration, and a utility function based on Shannon and Renyi entropy [2] is used for the computation of the utility of paths.

When dealing with uncertainty quantification from IT perspective, the authors of [7] address the joint entropy minimization exploration problem and propose two modified versions of Rapidly exploring Random Trees (RRT) [4], as dRRT* and eRRT* respectively. dRRT* uses distance, while eRRT* uses entropy change per distance traveled as the cost function. Actions are computed in terms of the joint entropy change per distance traveled. The simulation results proved that a combination of both of these approaches provides the best path-planning strategy. An interesting comparison between IT approaches is given in [8] where frontier-based exploration is deployed to select future candidate target positions. A comparison of joint Entropy reduction between the robot path and map is done against Expected Map Mean (EMM) and Kullback-Leiber Divergence. It was concluded that most of these approaches were not able to properly address the probabilistic aspects of the problem and are most likely to fail because of high computational cost and map grid resolution dependency on performance. The authors in [9] use entropy reduction only over map features and use an entropy metric based on Laplacian approximation with a unified quantification of exploration and exploitation gains.

Recently in the works of [11] and [12], the authors exploit the graph SLAM connectivity and pose it as an Estimation over Graph (EoG) problem, where each node (state vector) and vertex (measurement) connectivity is strongly related to the SLAM estimation reliability. By exploiting the spectral graph theory which deals with the Eigenvalues, Laplacian, and Degree matrix of the associated Fisher Information Matrix (FIM) and graph connectivity respectively the authors state that the graph Laplacian is related to the SLAM information matrix and the number of Weighted number of Spanning trees (WST) is directly related to the estimation accuracy in graph SLAM.

The authors in [14] [15] extend [12] by debating that the maximum number of WST is directly related to the Maximum Likelihood Estimate (MLE) of the underlying graph SLAM problem formulated over lie algebra. Instead of computing the D-optimality criterion over the entire SLAM sparse information matrix, a modern D-optimality criterion is computed over the weighted graph Laplacian where each vertex is weighted using edge D-Optimality. Furthermore, it is proven that the maximum number of WST of this weighted graph Laplacian is directly related to the underlying pose graph uncertainty.

III. PRELIMINARIES

A. Graph SLAM

Modern SLAM approaches adopt a graphical approach (bipartite graph) where each node represents the robot or landmark pose and each edge represents a pose to pose or pose to landmark measurement. As an example for a robot with four

pose states, shown in Fig. 1 $x_{0:3}$, $lm_{1:3}$ $u_{1:3}$, $m_{1:7}$ represent the robot pose, landmark pose, robot pose measurement, and landmark measurement respectively. The objective is to find the optimal state vector x^* which minimizes the measurement error $e_i(x)$ weighted by the covariance matrix $\Omega_i \in \mathbb{R}^{l \times l}$ where l is the dimension of the state vector x as shown in Equation 1. We direct interested readers to [22] for an introduction and comparison of SLAM methods.

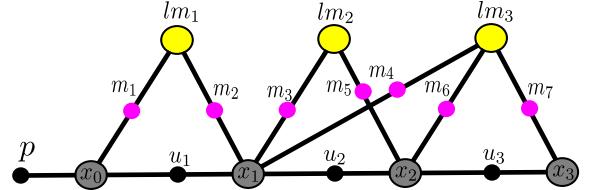


Fig. 1: Graph SLAM structure

$$x^* = \arg \min_x \sum_i \mathbf{e}_i^T(x) \Omega_i \mathbf{e}_i(x) \quad (1)$$

In A-SLAM the robot has to navigate in an unknown environment by performing actions in the presence of noisy sensor measurements that reduce its state and map uncertainties with respect to the environment. Such a scenario is modeled as an instance of the Partially Observable Markov Decision Process (POMDP) [16] [17] [18].

Let us consider a scenario where the robot is in state x and takes an action a to move to x' . The robot's goal is to choose the optimal policy χ^* that maximizes the associated expected reward (\mathbb{E}) for each state-action pair and it can be modeled as Equation 2:

$$\chi^* = \operatorname{argmax}_t \sum_{t=0}^{\infty} \mathbb{E} \Gamma^t \alpha(x_t, a_t) \quad (2)$$

Where x_t , a_t and Γ^t are respectively the state, action, and discount factor evolution at time t and $\alpha(x_t, a_t)$ is the reward function associated with the action performed to reach x . Although the POMDP formulation of A-SLAM is the most widely used approach, it is considered computationally expensive as it considers planning and decision-making under uncertainty. For computational convenience, it is divided into three main steps which identify the potential goal positions/waypoints, compute the cost to reach them, and then select actions based on a utility criterion that decreases map uncertainty and increases the robot's localization.

B. Uncertainty Quantification

The cost or utility is computed based on the reward value of the optimal action selected from a set of all possible actions according to Equation 2. For uncertainty quantification IT and TOED methods are used.

In IT, Shannon entropy/entropy measures the amount of uncertainty associated with a random variable or random quantity. Since robot pose and the map is estimated as a multivariate Gaussian distributions the authors in [19] describe Shannon

entropy of the map $E \in (0, 1)$ as in Equation 3 where the map M is represented as an occupancy grid and each cell $c_{i,j}$ is associated with a probabilistic estimation $P(c_{i,j})$ of its value of 1 being occupied and 0 as free. The objective is to reduce both the robot pose and map entropy. However, measuring entropy can be computationally demanding because computing probabilistic estimations of both the robot's position and the map over the entire map area is required along with its associated grid resolution.

$$E[p(M)] = - \sum_{i,j} (p(c_{i,j}) \log(p(c_{i,j})) + (1 - p(c_{i,j})) \log(1 - p(c_{i,j})) \quad (3)$$

Alternatively, if we consider task driven utility functions, the uncertainty metric is evaluated by reasoning over the propagation of uncertainty associated with the Fisher Information Matrix (FIM) $\Delta = \Omega^{-1}$ of graph SLAM. TOED provides many optimally criteria, which give a mapping of the covariance matrix to a scalar value. Less covariance contributes to a higher weight of the action set χ . For a covariance matrix $\Omega \in \mathbb{R}^{n \times n}$ and having eigenvalues ζ_n , A,D and E optimality criterion are defined which minimize the average variance, covariance ellipsoid and maximum Eigenvalue.

TOED approaches require both the robot pose and map uncertainties to be represented as covariance matrix and may be computationally expensive, especially in landmark-based SLAM where its size increases as new landmarks are discovered. Hence, IT-based approaches are preferred over TOED.

In [11] and [12] the authors debate how the graphical topology of SLAM has an impact on estimation reliability. They establish a relationship between the Weighted number of Spanning Trees (WST) and the D-Optimality criterion and show the graph Laplacian is closely related to the FIM. Three graph connectivity metrics are discussed with connection to the SLAM estimation accuracy, namely: 1) Algebraic Connectivity (A.C), defines the robustness/resilience of a graph to stay connected even after the removal of some nodes. For connected undirected graph $G = (v, \epsilon)$, having v vertices (poses) and ϵ edges (measurements), the A.C is defined as the second largest Eigen value of the weighted Laplacian $\lambda_2(L_0)$. 2) Average Degree (\bar{d}), indicates the number of edges incident upon its vertices. As $\bar{d} = \frac{1}{v} \sum_{i=1}^v \deg(i)$ increases, the number of measurements also increases which eventually improves the MLE estimate. 3) Tree Connectivity, it is shown that WST is directly related to the determinant of the weighted graph Laplacian and the MLE estimate. It is given as $t_w(G) = \det(L_w)$, where L_w is the weighted Laplacian.

The above graph connectivity indices provide alternate methods that are computationally less expensive to measure SLAM uncertainty as compared to TOED and IT approaches discussed previously. In Section V we will weigh the effectiveness of our proposed method using these indices.

IV. METHODOLOGY

The A-SLAM methods outlined in Section II assess uncertainty either through the overall map entropy or the SLAM's covariance matrix in its entirety, resulting in high computational costs. In this article, we propose a utility function that incorporates the path entropy and distance to the frontier candidate and adds it to the modern D-Optimality criterion as defined in [15]. Computing path entropy is computationally efficient along with penalizing frontiers with large distances. The final utility function not only provides a reliable SLAM estimate but also maximizes the map coverage by minimizing the unknown map area.

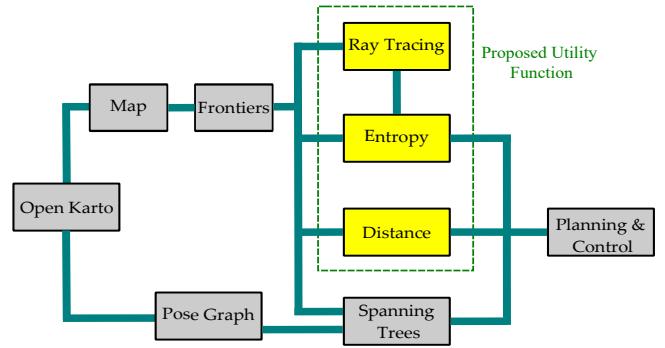


Fig. 2: Framework and proposed utility function

Figure 2 shows the overview of the proposed utility function. We add this utility function to that of [15] which uses Lidar-based Kart SLAM [21] back-end, with frontier detection over an Occupancy Grid (OG) map. For each frontier centroid (frontier candidate) we apply the Bresenham's line algorithm [20] to compute the number of pixels and their occupancy values in a straight line from the robot's current position. We then compute the path entropy to each frontier candidate and weigh the paths that have a higher number of unknown cells to favor the robot to explore the unknown environment resulting in better coverage. The lengthy candidate frontier paths are penalized so that the robot does not favor paths longer than a certain threshold and maintains its SLAM accuracy. Finally, the utility is computed by adding the proposed utility to that of [15]. Fig. 3 shows the implementation of the proposed utility function in ROS.

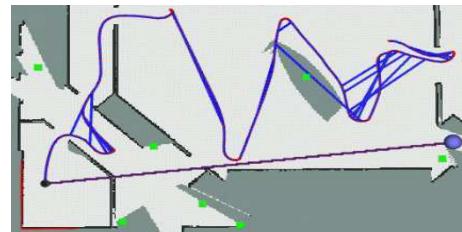


Fig. 3: Proposed utility function ROS implementation, purple line = Bresenham's line, sphere = frontier candidate, Green Squares = frontier centroids.

For each candidate frontier $F = \{f_1, f_2, \dots, f_N\} \in \mathbb{R}^2$, we get the occupancy values of each path as $G^n = \{m_0, m_1, \dots, m_L\}, \forall n \in N$, where $m_0 \dots m_L$ are the pixel occupancy values of path length L . We assign the probability value of $P_{unk} = 0.1$ for unknown pixels (with values = -1) to quantify for low entropy and high information gain (as we are more interested in unknown area of the environment). Occupancy values of obstacles and free space are mapped to probability $P_{Ofree} = 0.45$, weighing high entropy since we are not interested in places already known to the robot. Equation 4 computes the path entropy E^n for each frontier candidate with the above assigned probability values.

$$E^n = E^n[p(m)]_{m \in G^n} = - \sum_{m \in G^n} (P(m_{i,j}) \log_2(p(m_{i,j})) + P(1 - m_{i,j}) \log_2(1 - p(m_{i,j})), \forall m_{i,j} \in M \quad (4)$$

Once the path entropy is computed it is normalized with the number of pixels within the path K^n , as shown in Equation 5. $n = \{n_x, n_y\}$ and $R = \{R_x, R_y\}$ are the selected frontier and robot positions respectively.

$$K^n = \sum_{i=R_x}^{n_x} \sum_{j=R_y}^{n_y} m_{i,j} \quad (5)$$

The path entropy computed in Equation 4 may lead to select frontiers with large distances from the robot. These large distance frontiers may decrease the localization of the robot once it moves toward them. For penalizing these frontiers to contain the SLAM uncertainty, we apply an exponential decay operator γ^n as shown in Equation 6 and will use it in computing the utility function U_2^n in Equation 7. In Equation 6 λ is the decay rate operator which acts as a tuning factor for removing frontiers with large distances. We fix $\lambda = 0.6$ since we assume the environment is static and the frontier path entropy remains constant when the robot moves towards the frontier. n and R are frontier and robot locations in \mathbb{R}^2 respectively and $dist(*)$ measures the Euclidean distance between n and R .

$$\gamma^n = \exp^{-(\lambda * dist(R, n))} \quad (6)$$

The utility U_2^n as shown in Equation 7 is computed by weighing normalized entropy E^n/K^n with $\rho^n = 10^\beta$, where β is a factor which depends on the number spanning trees of the weighted graph Laplacian (L_w^n) computed in Equation 8. More specifically, β is the count of the number of digits before the decimal places of U_1^n and acts as a balancing factor between entropy and the number of spanning trees.

Finally, we obtain the proposed utility function U_{tot} in Equation 9 as the maximum of the sum U_1^n and U_2^n , where Equation 8 is adopted from [15]. The advantage of U_{tot} is that it not only provides a good SLAM estimate based on the modern D-Optimality criterion but also increases the coverage of the unknown map by reducing the frontier path entropy and distance.

$$U_2^n = (1 - E^n/K^n) * \rho^n + \gamma^n \quad (7)$$

$$U_1^n = \text{Span}(L_w^n) \quad (8)$$

$$U_{tot} = \max(U_1^n + U_2^n) \quad (9)$$

V. SIMULATION RESULTS

The simulations were carried out on ROS Noetic, Ubuntu 20.04 on Intel Core i7®, with a system RAM of 32GB and NVIDIA RTX 1000 GPU. We used the approach of [15] and implemented the proposed approach as mentioned in IV using Open Kart as SLAM backend, Turtlebot® equipped with Lidar, Dynamic Window Approach (DWA) [13] and Dijkstra's algorithm as local and global planners from the ROS navigation stack. We compared the proposed approach against two different methods which use Frontier Detection based Exploration (FD) [23] and Active Graph SLAM (AGS) of [15] and using two different simulation environments namely the modified Willow Garage (W.G) ³ measuring 2072m², having no dynamic obstacles, and modified office (Office) measuring 741m² with obstacles ⁴. The ground truth occupancy grid maps were generated using the gazebo_2Dmap_plugin ⁵ which uses wavefront exploration.

For qualitative comparison, we weighed our approach with performance metrics relating to graph connectivity, map efficiency, and % of the area covered. We used graph connectivity metrics like Algebraic Connectivity (A.C), average degree (\bar{d}), normalized tree connectivity ($\hat{\tau}(\mathcal{G})$), and evolution of uncertainty measured as edge D-optimality in the resulting pose-graph. These metrics are strongly related to SLAM estimation accuracy as described in Section III. Regarding map efficiency matrices, Structural Similarity Index (SSIM) and Root Mean Square Error (RMSE) are used. SSIM ∈ (0, 1) indicates the similarity between two maps by computing the covariance of their pixel values.

We conducted 15 simulations of 30 minutes each for both W.G. and office environments using FD, AGS, and our methods. The results from these simulations along with above mentioned performance matrices are elaborated in Table I and Fig. 5.

TABLE I: Average graph connectivity and map quality comparison of 15 simulations (30 minutes each for every method)

Env.	Meth.	A.C	\bar{d}	$\hat{\tau}(\mathcal{G})$	SSIM	RMSE
W.G	FD	0.104	3.290	1.016	0.05	0.70
	AGS	0.426	2.907	1.139	0.05	0.64
	Our	0.845	2.925	1.205	0.08	0.60
Office	FD	3.061	3.179	1.229	0.09	0.83
	AGS	5.740	2.742	1.312	0.07	0.80
	Our	9.617	2.612	1.941	0.11	0.77

The first benchmark is FD in which the robot is guided towards the nearest frontier and once it reaches the frontier,

³<https://github.com/arpq/Gazebo/>.

⁴<https://github.com/mlherd/>

⁵<https://github.com/marinaKollmitz/gazebo>.

it is added to the blacklist of chosen frontiers in order to avoid detecting it again. This approach does not take into account the FIM uncertainty of the pose-graph and nor does it favor revisiting already visited frontiers for loop closing. From Table I on both environments, we can conclude that this approach provides good \bar{d} because Open Karto creates many loop closure constraints between nodes but does not contribute to uncertainty reduction because the nodes are very close to one another. This method severely lacks in A.C (especially in W.G) as compared to our approach. As described in Section III A.C is directly related to the accuracy of SLAM estimation and its higher value is encouraged. Further, we observe a lower SSIM and higher RMSE, which renders this approach not suitable for area coverage tasks as compared to preceding methods. Since this method uses a greedy frontier detection search without any quantification of SLAM uncertainly or loop closure, as a result, the SLAM covariance increases, and exploration halts after some time. Eventually the resulting SLAM pose graph has high unreliability and less coverage as compared to Our method.

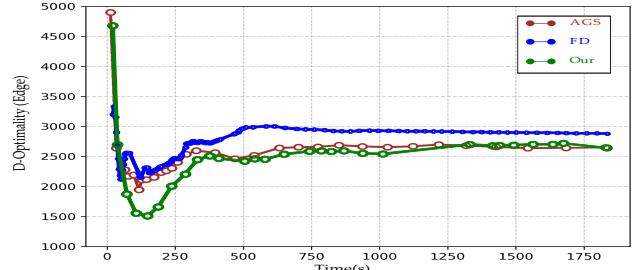
The second method in Table I is AGS. We can infer that this method results in good graph connectivity and map quality matrices in both environments as compared to FD. Especially in office environment the graph connectivity metrics are high. High SSIM and low RMSE indicate better SLAM estimation as compared to W.G. because this environment has more obstacles that bring structure to SLAM estimation.

Our method when compared to the preceding methods in Table I outperforms them, especially in A.C, $\hat{\tau}(\mathcal{G})$, SSIM, RMSE, and Map size for both environments. We can observe that for Willow Garage our method has 100% more A.C, 60% more SSIM, and 6% less RMSE error than the last best value. For modified Office we get 67% more A.C, 47% more $\hat{\tau}(\mathcal{G})$, 22% more SSIM, and 37% less RMSE. These promising values indicate the effectiveness of our proposed method.

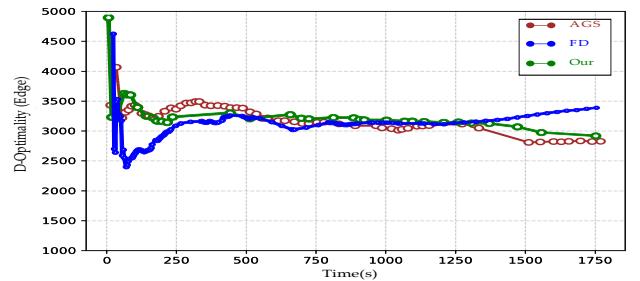
TABLE II: Uncertainty reduction (%R) comparison

Env.	Method	D-Opti(Max&Min)	Diff	%R
W.G	FD	3700&2900	800	20
	AGS	4800&2600	2200	45
	Our	4700&2600	2100	44
Office	FD	4600&3400	1200	26
	AGS	4100&2700	1400	34
	Our	4900&2900	2000	40

Figure 4 plots the uncertainty evolution over time (s) in the W.G and Office environment. We quantify the uncertainty as D-Optimality for each edge in the entire pose graph. The circles denote the selected goal frontiers. Each method has a different goal frontier detection frequency depending on whether the robot has reached it or not. From Fig. 4 and Table II we can deduce that initially the uncertainty is high and as the robot explores the environment, it decreases. Our approach and AGS manage to keep the uncertainty bounded to 44% and 45% of their maximum threshold respectively, while FD keeps it at 20% hence resulting in a poor SLAM estimate due to lack of loop closure and using a greedy frontier search.

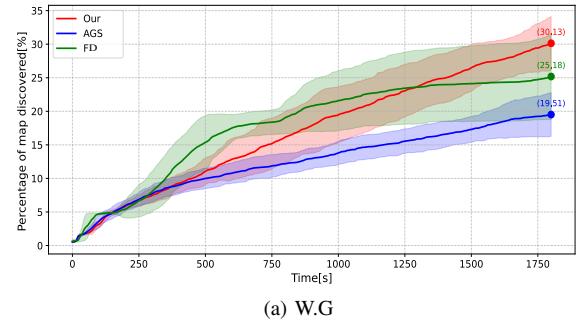


(a) W.G

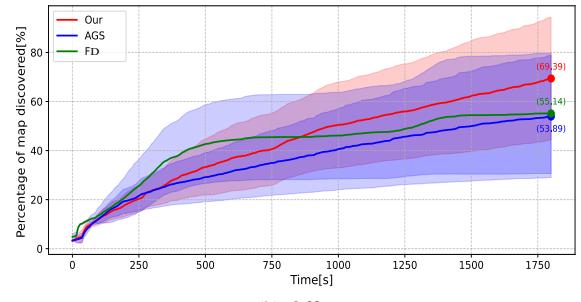


(b) Office

Fig. 4: Uncertainty evolution of AGS, FD and Our in 4a Willow Garage, 4b Office Environments.



(a) W.G



(b) Office

Fig. 5: Comparison of evolution of map discovered with average and standard deviation for our, AGS and FD methods. With 15 simulations (30 minutes each for every method)

In Fig. 5 a comparison of the evolution of the percentage of map discovered is presented. The average values along with the standard deviation of 15 simulations, 30 min each of FD,

AGS, and our are shown. Since FD uses a greedy frontier detection and exploration approach it starts with a higher slope than AGS and our method till 1000 and 500 seconds for W.G and Office respectively, after that it the SLAM covariance becomes unbounded, and the slope decreases with the final percentage of discovered map at 25.18% and 55.14% respectively. Both AGS and our method manage to keep the slope of average exploration values constant but our method eventually manages to explore 54%, 30% more area than AGS, and 20%, 25% more area than FD for both W.G and Office environments respectively. In Fig. 6 the resulting occupancy grid maps and pose graphs are overlapped on the ground truth maps to show the area covered by our approach in environments.

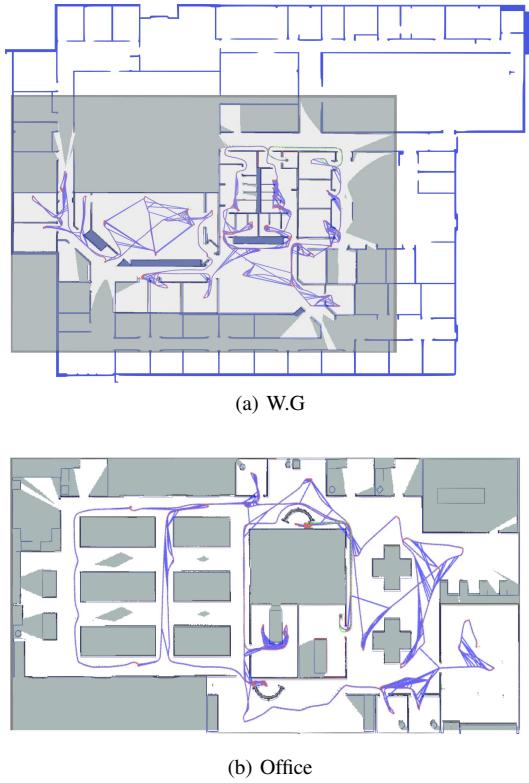


Fig. 6: Obtained pose graphs using Our approach and ground truth maps (blue).

VI. EXPERIMENTAL RESULTS

The experiments were performed using four-wheel diff-drive ROSBot2 robot³ as shown in Fig. 7a and ROS (noetic) on Ubuntu 20.04.6 (LTS) running on Intel Xeon® W-2235 CPU 3.80GHz x 12, 64Gb RAM and Nvidia Quadro RTX 4000 GPU.

The environment consists of a room (lab environment) with static obstacles and two corridors measuring $81m^2$ in total. We used mapping efficiency and exploration time as performance matrices for experimental results. Fig. 7b shows the computed

OG map, and SLAM pose graph along with robot start and end positions using our approach. The computed OG map by the robot is overlapped with that of the ground truth (blue). We can observe high similarity between the two maps, rendering the high mapping efficiency of our approach.

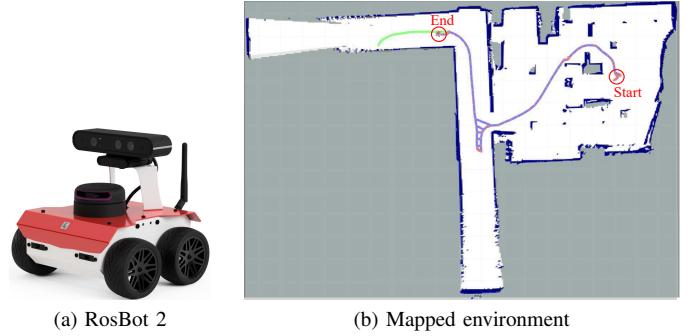


Fig. 7: Robot used for experiments and the resulting mapped environment overlapped with ground truth map(blue).

Figure 8 shows the average percentage of four experiments (2-AGS, 2-Our) for a total duration of 350 seconds. We can observe that initially both AGS and our method approximately map 35% of the environment this is because of the high range (up to 16m) of the Lidar sensor ⁴ of RosBot 2 robot. With the evolution of time we can observe that using our utility function, a steep slope is obtained from 60 to 210 seconds (approx.) which helps the robot to cover the entire environment in 230 seconds as compared to 280 seconds for AGS. These results are in contrast with Section V and indicate that our approach supersedes AGS and therefore can be utilized for efficient exploration of the environment while maintaining good SLAM efficiency.

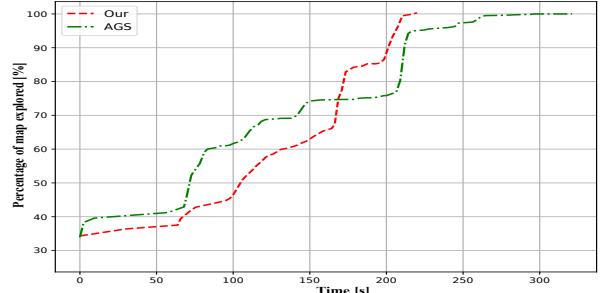


Fig. 8: Average Coverage Percentage in four experiments.

VII. CONCLUSIONS

In this article, we have presented a utility function that selects the most favorable frontier goal location within an occupancy grid map for a reliable A-SLAM with an area coverage task. The proposed utility function incorporates path entropy to select the frontier goal location which has the highest amount of unknown cells within its path thus maximizing

³<https://husarion.com/manuals/rosbot/> .

⁴<https://www.slamtec.com/en/Lidar/A2/> .

the area coverage. Using simulation and experimental results on publicly available environment maps we have proved the efficiency of our approach as compared to similar methods. As a future prospective, we plan to incorporate our method in a multi-robot scenario utilizing efficient frontier-sharing for maximum environment exploration.

ACKNOWLEDGMENT

This work was conducted within the framework of the NExT Senior Talent Chair DeepCoSLAM, funded by the French Government through the program "Investments for the Future" administered by the National Agency for Research (ANR-16-IDEX-0007). We also extend our gratitude to the Région Pays de la Loire and Nantes Métropole for their invaluable support in facilitating this research endeavour.

REFERENCES

- [1] B. Yamauchi, "A frontier-based approach for autonomous exploration", in Proceedings IEEE CIRA'97, IEEE Comput. Soc. Press, pp. 146–151, 1997.
- [2] H. Carrillo, P. Dames, V. Kumar, and J. A. Castellanos, 'Autonomous robotic exploration using a utility function based on Rényi's general theory of entropy', Auton Robot, vol. 42, no. 2, pp. 235–256, Feb. 2018.
- [3] A. Pázman, "Foundations of optimum experimental design", Springer, vol. 14, 1996.
- [4] Naderi, K.; Rajamäki, J.; Hämäläinen, P. RT-RRT*: A real-time path planning algorithm based on RRT*. 8th ACM SIGGRAPH Conference on Motion in Games, Paris, France, 16–18 November 2015; pp. 113–118.
- [5] Trivun, D., Salaka, E., Osmankovic, D., Velagic, J., & Osmic, N. "A-SLAM-based algorithm for autonomous exploration with Mobile Robot", IEEE ICIT, 2015.
- [6] Mammolo, D., "A-SLAM in Crowded Environments", master thesis, Autonomous Systems Lab, ETH Zurich, Zurich, 2019
- [7] Vallve, J. & Andrade-Cetto, J., "Active pose slam with RRT", IEEE ICRA, 2015.
- [8] Du, J., Carbone, L., Kaouk Ng, M., Bona, B., & Indri, M., "A comparative study on A-SLAM and autonomous exploration with particle filters", IEEE/ASME AIM, 2011.
- [9] Mu, B., Giamou, M., Paull, L., Agha-Mohammadi, A., Akbar, Leonard, J., & How, J., "Information-based A-SLAM via topological feature graphs", IEEE 55th CDC, 2016.
- [10] Xue, W., Ying, R., Wen, F., Chen, Y., & Liu, P., "A-SLAM with prior topo-metric graph starting at uncertain position", IEEE Robotics and Automation Letters, 7(2), 1134–1141, 2022
- [11] Khosoussi, K., Huang, S., & Dissanayake, G., "Novel Insights into the Impact of Graph Structure on SLAM", In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, pp.2707–2714, 2014.
- [12] Khosoussi, K., Giamou, M., Sukhatme, G.S., Huang, S., Dissanayake, G., & How, J.P., "Reliable Graphs for SLAM", IJRR, 38, 260–298, 2019.
- [13] Fox, D.; Burgard, W.; Thrun, S. The Dynamic Window Approach to Collision Avoidance. IEEE Robot. Autom. Mag. 1997, 4, 23–33, <https://doi.org/10.1109/100.580977>.
- [14] J. A. Placed & J. A. Castellanos, "A General Relationship Between Optimality Criteria and Connectivity Indices for Active Graph-SLAM", IEEE Robotics and Automation Letters, vol. 8, no. 2, pp. 816-823, 2023.
- [15] Placed, J.A., & Castellanos, J.A., "Fast Autonomous Robotic Exploration Using the Underlying Graph Structure", IEEE/RSJ IROS, Prague, Czech Republic, pp. 6672–6679, 2021.
- [16] Fox, D., Burgard, W., & Thrun, S. "Active Markov Localization for Mobile Robots", Robotics and Autonomous Sys., 25, 195–207, 1998.
- [17] Placed, J.A.; Castellanos, J.A. A Deep Reinforcement Learning Approach for Active SLAM. Applied Sciences 2020, 10, 8386, doi:10.3390/app10238386.
- [18] Placed, J. A., Strader, J., Carrillo, H., Atanasov, N., Indelman, V., Carbone, L., Castellanos, J. A. (2023). A survey on active simultaneous localization and mapping: State of the art and new frontiers. IEEE Transactions on Robotics (T-RO)
- [19] Stachniss, C., Grisetti, G., & Burgard, W., "Information Gain-Based Exploration Using Rao-Blackwellized Particle Filters", In Proceedings of the Robotics, Science and Systems I, Robotics: Science and Systems Foundation, 2005.
- [20] Bresenham, J. E., "Algorithm for computer control of a digital plotter", IBM Systems Journal, 4 (1): 25–30, 1965.
- [21] Kurt Konoli, Giorgio Grisetti, Rainer Kummerle, Benson Limketkai & Regis Vincent, "Efficient Sparse Pose Adjustment for 2D mapping[J]", IEEE/RSJ IROS, 2010.
- [22] Z. Xuexi, L. Guokun, F. Genping, X. Dongliang & L. Shiliu, "SLAM Algorithm Analysis of Mobile Robot Based on Lidar," CCC, Guangzhou, China, pp. 4739-4745, 2019.
- [23] H. Jiří, "Map-Merging for multi-robot system," B.S. thesis, Faculty of Mathematics and Physics, Charles University in Prague., Prague, Czech Republic, 2016.