A Benchmark of Planning-based Exploration Methods in Photo-Realistic 3D Simulator

Xuan Du[3*], Xinyi Yang[1*], Chao Yu[1*], Jiaxuan Gao[2], Qingmin Liao[3], Huazhong Yang[1], and Yu Wang[1]

Abstract— **Autonomous exploration is an essential ability of mobile robots and has been widely investigated. A reliable evaluation is required to compare exploration methods in different aspects. However, existing benchmarks are mostly implemented on gird-based simulators which are lack of real-world environmental factors. Besides, most benchmarks only focus on single-agent exploration or multi-agent exploration, lacking comprehensive analysis on both of them. In this paper, we select representative planning-based exploration techniques in both single-agent and multi-agent settings and evaluate them on large and highly cluttered maps. The experiment is conducted on a photo-realistic 3D simulator, Habitat, and the results indicate the performance of these methods in aspects of map size, cooperation, efficiency and agent's team size.**

I. INTRODUCTION

Autonomous exploration is an important ability for building intelligent robot systems, where a robot should adaptively locate its position and explore unknown regions based on its sensory signals with limited time budget. This problem is often formulated as active Simultaneous Localization and Mapping (SLAM) [1], [2]. Autonomous exploration has been extensively studied in enormous domains including rescue [3], autonomous driving [4], drone [5] and mobile robots [6].

It has been a recent trend to apply classical planning-based methods to tackle this challenge. The frontier-based techniques such as [7], [8], [9] are the most widely used heuristics which adaptively select navigation goals that locate on the boundary between the explored and the unexplored region. Although single-agent exploration has been extensively researched, there are problems with limited receptive field and low exploration efficiency [10]. An intuitive solution is to adopt the multi-agent paradigm, where multiple agents explore an unseen scenario with broader vision in a cooperative manner. Many works have extended classical planning-based exploration methods from the single-agent setting to the multi-agent setting by sharing the reconstructed map or applying some explicit cooperation heuristics [11], [12], [13], [14].

These exploration algorithms are usually developed and tested in simulators where the performance of a method is closely related to simulator settings, such as the size of the exploration space, the number of agents, and the errors in the constructed map. For a comprehensive analysis of the pros and cons of different algorithms, comparisons need to be made under uniform and representative simulator settings. Several researchers have investigated this topic. [15] evaluates the performance of seven single-agent algorithms for autonomous exploration in Gazebo-based RotorS simulator. [16] compares four single-agent exploration algorithms on a two-dimensional simulator with obstacles in the shape of simple geometry. Besides, there are also a lot of works evaluating the performance of different multi-agent exploration algorithms [17], [18], [19], [20]. [17] evaluates the performance of five methods aiming at search and rescue problems with 10 to 30 robots. [18] tests five frontier-based strategies with 3 to 7 robots. [20] designs a multi-agent exploration dataset and proposes a set of evaluation metrics of single-agent and multi-agent algorithms.

However, these works mainly focus on simplified simulators such as the grid world and the particle world [21]. These simulators ignore real-world errors such as the spatial structure and errors of perception, location as well as constructed maps. This results in a huge difference or even invalidation between the evaluation conclusion and the real world.

Therefore, this paper focuses on studying the performance of different algorithms in the exploration with localization and mapping errors, and thus benchmarks different planning-based exploration methods in a photo-realistic simulator, Habitat[22]. We select 3 representative single-agent methods and 3 representative multi-agent methods for a fair comparison. In order to further explore the impact of physical spaces with different sizes on the performance of algorithms, we divide the Habitat-Matterport 3D dataset (HM3D) [23] into 3 groups, i.e. middle, large and super large, according to the size of the explorable space. In addition, we also evaluate the impact of different numbers of agents on the performance of exploration algorithms.

II. RELATED WORK

A. Single-Agent Exploration

Autonomous exploration has been widely investigated, where the mobile robot explores an unseen environment based on its sensor signals to maximize coverage. Simultaneous Localization and Mapping (SLAM) [24] provides the agent's location and a top-down occupancy map in planning-based methods. [7], [8], [25], [26], [27] solve exploration problems by selecting frontiers located on the boundary between the explored and the unexplored regions. [7] chooses the nearest frontier to the agent's position as next navigational goal, and [25], [26], [27] consider the distance between the agent's location and the selected frontier directly. In contrast, [28]

*Equal contribution

[1]C. Yu, X. Yang, Y. Wang, H. Yang, are with the Department of Electronic Engineering, Tsinghua University.

[2]J. Gao is with Institute for Interdisciplinary Information Sciences, Tsinghua University.

[3]X. Du, Q. Liao are with Shenzhen International Graduate School, Tsinghua University.

†Corresponding author: Yu Wang yu-wang@tsinghua.edu.cn

formulates a cost-utility function where a frontier is selected if it provides the highest information gain. A representative of sample-based methods is Rapidly-exploring Random Trees (RRT) [8] which searches frontiers via rapidly-exploring random trees to improve the efficiency. In our work, we give a fair comparison among the above methods in the single-agent setting to figure out the performance by different metrics.

B. Multi-Agent Exploration

Several coordinated strategies have been proposed to tackle the multi-agent exploration problem. The frontier-based method, Multi-robot Multi-target Potential Field (MMPF) [10], adopts the potential field based in the occupancy map so that agents tend to walk towards targets with the lowest potential values. A representative sample-based technique for multi-agent exploration is Weighted Multi-Agent RRT (WMA-RRT) [14], a variant of RRT, that uses single query roadmap with a shared tree. [29] divides map into Voronoi cells and assigns agents the cells to avoid repeated exploration. We propose a benchmark to carefully analyse the pros and cons among these methods under the multi-agent setting.

C. Exploration Benchmarks

Several benchmarks on autonomous exploration have been proposed. [15] compares 7 single-agent algorithms in Micro-Aerial Vehicles domain and evaluates their performance on the Gazebo-based RotorS simulator. [16] analyses the strengths and weaknesses of 4 single-agent exploration algorithms on a two-dimensional simulator with obstacles in the shape of simple geometry. [18] compares 5 frontier-based exploration strategies with 3, 5, 7 robots in a well-defined evaluation environment and designs a methodology for a comparison which satisfies reproducibility and repeatability. [17] evaluates 5 state-of-the-art algorithms for multi-agent exploration using 10, 20, 30 agents and proves that sociobiological inspiration is useful for cooperation. [19] presents several metrics to quantitatively measure the performance of multi-agent exploration methods with 1 to 30 agents on maps where some different terrains are added on the basis of RoboCup Rescue competition setting. [20] provides public datasets, designs a grid-based simulator and develops a benchmark for both single-agent and multi-agent exploration. However, the previous benchmarks either lack a comprehensive comparison for both single-agent and multi-agent settings, or conduct experiments only on simplified simulators (e.g. Webots [30]). In our benchmark, we take single-agent and multi-agent methods into account for a comprehensive evaluation and adopts a 3D photo-realistic simulator, Habitat [22], for more reliable results.

III. METHODOLOGY

We divide representative exploration techniques into 2 subsets, including 3 single-agent methods and 3 multi-agent methods. Single-agent exploration methods comprise RRT, Utility and Nearest, and multi-agent exploration methods comprise MMPF, Voronoi and WMA-RRT.

Map Sizes	$N=1$	$N=2$	$N=3$	$N=4$
Middle	600	450	450	350
Large	950	800	720	640
Super Large	2100	1900	1700	1500

TABLE I: Maximum episode step of different map sizes and different number of agents N.

A. Single-Agent Exploration methods

1) Nearest: Frontiers locate on the boundary between the explored area and the unknown area in the occupancy map. A frontier cell is selected as next navigational goal if it is the nearest one to the agent.

2) Utility: A cost-utility function is designed to calculate the benefit of reaching a cell through the information gain. This objective function considers both utility and cost to select the target cell.

3) RRT: This method mainly includes two parts: a global frontier detector and a local frontier detector. In the global frontier detector, the tree is initiated and grows like RRT, and a leaf node in the unexplored region is added if there is no obstacle in the connection of it and its nearest node. The local frontier detector rebuilds the tree and starts next iteration once a new leaf node is found. In this way, the area far from the agent can be covered by the global frontier detector and the efficiency of detecting leaf nodes is improved by the local frontier detector. Frontiers from detectors are then filtered and assigned to the agent.

B. Multi-Agent Exploration Methods

1) MMPF: The potential field-based exploration method builds the field in the explored map, where obstacles are given high potential values while targets are given lower values. Agents tend to move towards targets with low potential values and keep away from obstacles with high values. As a result, collision to obstacles may be avoided and the fastest way to the target can be calculated by the gradient descent. To alleviate the local optimum caused by multiple targets, we choose to calculate the value by using the square of wave-front distance [31] to slower the decrease of the potential value, and thus the influence of other agents far away from targets becomes weak.

2) Vonoroi: Voronoi partition divides the explorable area into several cells based on agents' positions, that is, each agent owns a Voronoi cell and locates in the center of the cell, making every point in the cell nearest to the assigned agent. Note that Voronoi partition is updated every time step so that the overlap is avoided and disabled agents are excluded naturally.

3) WMA-RRT: The weighted Multi-Agent RRT is a single-query roadmap to acquire the connectivity of the explorable space, which guarantees the high probability of covering the whole environment. All the agents are enclosed by a convex polygon via convex hull algorithm, and the intersection point of two longest diameters of the polygon is selected as Main

Methods	*Cov.* ↑	*Steps*↓	*Repeat* ↓
Random	0.31(0.01)	600.00(0.00)	38.67(2.49)
Nearest	0.62(0.02)	584.06(2.13)	145.02(3.33)
RRT	**0.94**(0.00)	**464.69**(9.90)	**117.24**(3.51)
Utility	0.83(0.01)	527.54(4.46)	169.32(4.09)

TABLE II: Performance of different algorithms in middle maps with single agent.

Methods	*Cov.* ↑	*Steps*↓	*Repeat* ↓
Random	0.25(0.01)	950.00(0.00)	92.54(5.65)
Nearest	0.79(0.02)	**805.57**(7.60)	**227.23**(6.07)
RRT	**0.91**(0.00)	814.17(10.26)	243.59(2.49)
Utility	0.80(0.01)	895.54(23.93)	346.78(5.69)

TABLE III: Performance of different algorithms in large maps with single agent.

Root. Then Secondary Root for each agent is ϵ meters away from Main Root and is served as the root of each RRT tree. After that, the tree grows similar to RRT, where the weight of edges is 0 originally and then is updated every time step. Besides, utility function encourages agents to select edge with the least cost and a locking mechanism restricts agents to move along the edge of the tree to avoid the overlap.

IV. Experiment Results

A. Experiment Setting

Our experiments are employed on a photo-realistic 3D simulator, Habitat Platform [22], which offers embodied agents real-time visual signals and physical dynamics. Agents are equipped with RGB-D sensors and the sizes of RGB-D images are 640 X 480. The action space consists of *move_forward, turn_left, turn_right*. A depth image projection is used to update gird map that contains explored space and unexplored space and offers navigation information for the following exploration.

We choose Habitat-matterport 3d dataset (HM3D) [23] for our experiments. Unlike other indoor 3D datasets such as Gibson [32] and Matterport3D [33], each scene in HM3D contains an entire building, making it 1.4-3.7x larger than others. Performance of an exploration method is closely relevant to the map size and the terrain. There are various sizes of maps in HM3D and we carefully select 8 highly cluttered scenes that are difficult to explore and divide them into 3 groups: middle (>70 m^2), large (>100 m^2), super large (>200 m^2).

Methods	*Cov.* ↑	*Steps*↓	*Repeat* ↓
Random	0.24(0.01)	2100.00(0.00)	230.17(18.12)
Nearest	0.30(0.02)	2078.22(3.05)	**455.25**(36.07)
RRT	**0.85**(0.01)	**1792.91**(41.04)	596.27(20.52)
Utility	0.75(0.02)	1844.03(34.72)	801.21((33.82)

TABLE IV: Performance of different algorithms in super large maps with single agent.

We evaluate the performance of MMPF, Voronoi, WMA-RRT as the subset of multi-agent exploration algorithms (the number of agents is 2,3,4) and RRT, Utility, Nearest as the subset of single-agent exploration algorithms. The single-agent strategies RRT, Utility, Nearest are also tested on multi-agent setting since we empirically found single-agent strategies may perform well in multi-agent exploration. Every evaluation is performed with 100 episodes over 3 random seeds. The total step of each episode is conditioned on the number of agents and the size of the map. The details are shown in Table I. We propose a more challenging task where agents are born in random positions within 2 meters instead of simply scattering them on the map.

B. Evaluation Metrics

We consider 5 behavior statistics measurements to capture different characteristics of a particular exploration strategy.

- *Coverage*: the final ratio of explored area to total area when agents reach maximum episode step. Higher *Coverage* implies more exhaustive exploration.
- *Overlap*: the ratio of repeated area explored by multiple agents to current explored area when 90% coverage is reached. Lower *Overlap* implies better credit assignment. We remark that *Overlap* can not be used in single-agent setting.
- *Repeat*: the ratio of the repeat region explored by an agent to the entire area when 90% coverage is reached. A lower *Repeat* implies better exploitation capability.
- *Steps*: the total timesteps when the coverage ratio reaches 90%. Fewer *Steps* implies more efficient exploration.
- *Time*: the wall-time consumed to calculate next step. Lower *Time* implies more time-efficient exploration.

C. Main Results

The test results of single-agent exploration methods are shown in Table II-IV and the results of multi-agent exploration methods are shown in Table V-VII. Time consumption of each algorithms are shown in Table VIII. We draw the conclusion of each algorithm with 4 metrics mentioned above, showing different performance in autonomous exploration.

1) Comparison of Map Size: According to the total explorable area we classify maps into three groups where larger maps are more uncertain to explore. We found that RRT performs the best on all kinds of maps with the highest *Coverage* and the lowest *Steps*, expressing that the randomly growing tree can tackle arbitrary terrains (e.g. narrow passages) as its nodes are abstract in shape. That is more robust than the methods trying to navigate to the frontiers which are commonly misjudged. MMPF represents its good exploration ability with almost achieving the best *Coverage* and *Steps* among the subset of multi-agent exploration strategies, indicating that the potential field is

#Agent	Metrics	Random	Nearest	RRT	Utility	MMPF	WMA-RRT	Voronoi
2	*Coverage* ↑	0.39(0.01)	0.68(0.01)	**0.96**(0.00)	0.91(0.00)	0.92(0.01)	0.90(0.01)	0.93(0.01)
	Steps ↓	450.00(0.00)	448.13(0.45)	**275.84**(3.61)	362.62(4.60)	329.10(7.16)	362.89(3.41)	323.48(7.72)
	Overlap ↓	**0.31**(0.02)	0.71(0.01)	0.46(0.01)	0.67(0.01)	0.54(0.01)	0.65(0.01)	0.43(0.01)
3	*Coverage* ↑	0.49(0.02)	0.72(0.01)	**0.97**(0.00)	0.94(0.00)	0.94(0.01)	0.91(0.01)	0.94(0.00)
	Steps ↓	450.00(0.00)	445.26(1.43)	**208.10**(3.28)	300.87(5.96)	280.45(7.05)	310.79(7.25)	285.18(3.87)
	Overlap ↓	**0.26**(0.01)	0.66(0.01)	0.39(0.02)	0.58(0.01)	0.41(0.01)	0.57(0.01)	0.29(0.01)
4	*Coverage* ↑	0.47(0.01)	0.66(0.01)	**0.97**(0.00)	0.93(0.00)	0.92(0.01)	0.88(0.01)	0.92(0.01)
	Steps ↓	350.00(0.00)	349.91(0.09)	**179.42**(2.46)	254.66(3.08)	249.43(4.90)	265.22(5.90)	257.42(2.99)
	Overlap ↓	**0.20**(0.01)	0.63(0.01)	0.34(0.01)	0.53(0.01)	0.32(0.01)	0.51(0.01)	**0.20**(0.00)

TABLE V: Performance of different algorithms in middle maps with $N = 2, 3, 4$ agents.

#Agent	Metrics	Random	Nearest	RRT	Utility	MMPF	WMA-RRT	Voronoi
2	*Coverage* ↑	0.34(0.01)	0.78(0.01)	**0.97**(0.00)	0.91(0.00)	0.90(0.02)	0.93(0.01)	0.89(0.01)
	Steps ↓	800.00(0.00)	710.79(5.54)	**486.77**(18.27)	649.88(9.25)	599.75(26.10)	625.45(0.36)	651.46(10.10)
	Overlap ↓	**0.34**(0.02)	0.82(0.01)	0.46(0.03)	0.70(0.01)	0.39(0.03)	0.66(0.01)	0.38(0.02)
3	*Coverage* ↑	0.38(0.01)	0.74(0.02)	**0.97**(0.00)	0.94(0.01)	0.95(0.00)	0.95(0.01)	0.93(0.00)
	Steps ↓	720.00(0.00)	666.18(2.80)	**398.32**(10.36)	546.56(13.56)	460.72(13.79)	522.17(8.23)	534.13(8.58)
	Overlap ↓	0.28(0.01)	0.80(0.01)	0.39(0.01)	0.62(0.01)	0.35(0.01)	0.59(0.00)	**0.24**(0.01)
4	*Coverage* ↑	0.38(0.02)	0.65(0.02)	0.95(0.04)	0.95(0.00)	**0.96**(0.01)	**0.96**(0.01)	0.91(0.01)
	Steps ↓	640.00(0.00)	628.04(1.89)	**385.11**(55.73)	476.54(2.91)	404.65(12.67)	465.39(10.55)	493.59(21.67)
	Overlap ↓	0.24(0.01)	0.80(0.01)	0.36(0.01)	0.57(0.00)	0.32(0.02)	0.54(0.01)	**0.19**(0.00)

TABLE VI: Performance of different algorithms in large maps with $N = 2, 3, 4$ agents.

#Agent	Metrics	Random	Nearest	RRT	Utility	MMPF	WMA-RRT	Voronoi
2	*Coverage* ↑	0.31(0.01)	0.45(0.04)	**0.94**(0.00)	0.84(0.04)	0.86(0.04)	0.83(0.03)	0.75(0.07)
	Steps ↓	1900.00(0.00)	1872.26(10.29)	**1159.52**(34.18)	1557.47(106.63)	1454.47(61.34)	1596.63(37.51)	1619.16(132.87)
	Overlap ↓	0.42(0.01)	0.70(0.02)	**0.32**(0.01)	0.63(0.02)	0.42(0.03)	0.56(0.01)	0.39(0.05)
3	*Coverage* ↑	0.39(0.02)	0.53(0.02)	**0.96**(0.01)	0.92(0.02)	0.91(0.02)	0.90(0.01)	0.81(0.01)
	Steps ↓	1700.00(0.00)	1638.06(16.47)	**880.92**(18.55)	1310.89(35.21)	1188.71(139.53)	1300.28(15.40)	1373.85(19.85)
	Overlap ↓	0.33(0.01)	0.63(0.01)	0.25(0.01)	0.55(0.02)	0.32(0.05)	0.48(0.01)	**0.23**(0.01)
4	*Coverage* ↑	0.33(0.00)	0.56(0.03)	**0.97**(0.00)	0.92(0.00)	0.91(0.04)	0.88(0.00)	0.81(0.01)
	Steps ↓	1500.00(0.00)	1447.72(22.19)	**741.43**(0.00)	1179.52(34.55)	1014.13(194.06)	1140.47(19.15)	1277.36(15.22)
	Overlap ↓	0.31(0.02)	0.62(0.02)	0.18(0.00)	0.48(0.02)	0.27(0.05)	0.42(0.00)	**0.17**(0.01)

TABLE VII: Performance of different algorithms in super large maps with $N = 2, 3, 4$ agents.

#Agent	Random	Nearest	RRT	Utility	MMPF	WMA-RRT	Voronoi
1	0.356	0.548	1.964	2.065	-	-	-
2	0.833	0.913	2.673	4.903	2.577	2.753	2.977
3	1.233	1.887	4.349	6.983	4.657	3.537	5.297
4	2.593	2.697	6.187	11.439	6.319	5.268	7.633

TABLE VIII: Time consumption for different strategies with $N = 1, 2, 3, 4$ agents.

helpful to find unexplored corners which are hard for frontier-based strategies. WMA-RRT, served as a multi-agent variant of RRT, reaches the highest *Coverage* on large maps but gets poor one on middle maps and super large maps since the designed mechanisms inevitably limit the randomness and the accidental coverage brought by random search. Besides, WMA-RRT is less efficient as it takes more steps to achieve 90% of coverage ratio. Voronoi performs the best on middle maps among other multi-agent algorithms where its *Coverage*, *Steps*, *Time* come out top but the they drops dramatically as the maps getting larger. A possible explanation is Voronoi partitions may cut off the navigable path so as to impede agents for efficient exploration.

2) Comparison of Cooperation: The ability of cooperation is inflected by *Overlap*, i.e. lower *Overlap* implies better credit assignment. Voronoi shows unbeatable capability of cooperation in all conditions that its *Overlap* is apparently much lower than others since the Voronoi partition map into nonoverlapping cells and repeated exploration is avoided. MMPF also does well in cooperation compared to WMA-RRT and other single-agent exploration strategies owing to the existence of potential field and repulsive force between agents. *Overlap* of WMA-RRT is lower than Nearest and Utility since a lock mechanism is applied to prevent other agents from entering the same edge simultaneously and the utility functions encourage agents to explore the unallocated branches. But at the beginning, the Secondary Roots limit the distance from agents to Main Root and consequently get trees close to each other, which may result in repeated exploration.

3) Comparison of Time Efficiency: From Table VIII we can see that *Time* of Nearest is obviously less than other methods as it simply takes the nearest frontier as next goal while Utility is the worst algorithm in terms of time efficiency for consuming too much time to calculate the objective function. Demanding much time to get next goal is very intractable in the dynamic real world. MMPF is comparable to RRT in *Time* and they are a bit faster than Voronoi. *Time* of WMA-RRT is less than Voronoi and MMPF. As the increase of agents, its computation time grows slower than other methods, indicating that WMA-RRT is more practical in dynamic environment.

4) Comparison of Agent's Team Size: Table V-VII shows that more agents lead to higher *Coverage* and lower *Steps*. RRT keeps very high *Coverage* and doesn't vary a lot in different team sizes while MMPF is sensitive to number of agents that with more agents the increase of *Coverage* is more apparent than WMA-RRT and Voronoi.

V. CONCLUSION

A comprehensive evaluation of representative planning-based exploration methods has been conducted in both single-agent and multi-agent settings on a photo-realistic 3D simulator, Habitat. We select large and highly cluttered scenes for complex exploration and evaluate these methods in efficiency, cooperation and time consumption for the thorough comparison. In single-agent setting, RRT performs the best, indicating a tree search algorithm encourages the exploration efficiency. On the other hand, Voronoi excels in the coordination and the efficiency for its partition method in multi-agent setting. Future work will concentrate on the benchmarks on exploration in the real-world application.

ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation of China (U20A20334, U19B2019 and M-0248), Tsinghua-Meituan Joint Institute for Digital Life, Tsinghua EE Independent Research Project, Beijing National Research Center for Information Science and Technology (BNRist), National Key Research and Development Program of China (No. 2019YFF0301500), and Beijing Innovation Center for Future Chips.

REFERENCES

[1] 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 Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.

[2] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using fastslam," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1985–1991.

[3] A. Kleiner, J. Prediger, and B. Nebel, "Rfid technology-based exploration and slam for search and rescue," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 4054–4059.

[4] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[5] L. von Stumberg, V. Usenko, J. Engel, J. Stückler, and D. Cremers, "From monocular slam to autonomous drone exploration," in *2017 European Conference on Mobile Robots (ECMR)*. IEEE, 2017, pp. 1–8.

[6] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419839596, 2019.

[7] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.

[8] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1396–1402.

[9] C. Dornhege and A. Kleiner, "A frontier-void-based approach for autonomous exploration in 3d," *Advanced Robotics*, vol. 27, no. 6, pp. 459–468, 2013.

[10] J. Yu, J. Tong, Y. Xu, Z. Xu, H. Dong, T. Yang, and Y. Wang, "Smmr-explore: Submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[11] K. M. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1160–1165.

[12] W. W. Cohen, "Adaptive mapping and navigation by teams of simple robots," *Robotics and autonomous systems*, vol. 18, no. 4, pp. 411–434, 1996.

[13] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[14] A. N. Nazif, A. Davoodi, and P. Pasquier, "Multi-agent area coverage using a single query roadmap: A swarm intelligence approach," in *Advances in practical multi-agent systems*. Springer, 2010, pp. 95–112.

[15] A. Brunel, A. Bourki, O. Strauss, and C. Demonceaux, "Flybo: A unified benchmark environment for autonomous flying robots," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 1420–1431.

[16] F. Amigoni, "Experimental evaluation of some exploration strategies for mobile robots," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 2818–2823.

[17] M. S. Couceiro, P. A. Vargas, R. P. Rocha, and N. M. Ferreira, "Benchmark of swarm robotics distributed techniques in a search task," *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 200–213, 2014.

[18] J. Faigl and M. Kulich, "On benchmarking of frontier-based multi-robot exploration strategies," in *2015 european conference on mobile robots (ECMR)*. IEEE, 2015, pp. 1–8.

[19] Z. Yan, L. Fabresse, J. Laval, and N. Bouraqadi, "Metrics for performance benchmarking of multi-robot exploration," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3407–3414.

[20] "Explore-bench: Data sets, metrics and evaluations for frontier-based and deep-reinforcement-learning-based autonomous exploration," *arXiv preprint arXiv:2202.11931*, 2022.

[21] C. Wakilpoor, P. J. Martin, C. Rebhuhn, and A. Vu, "Heterogeneous multi-agent reinforcement learning for unknown environment mapping," *arXiv preprint arXiv:2010.02663*, 2020.

[22] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9339–9347.

[23] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang *et al.*, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," *arXiv preprint arXiv:2109.08238*, 2021.

[24] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial intelligence review*, vol. 43, no. 1, pp. 55–81, 2015.

[25] Y. Mei, Y.-H. Lu, C. Lee, and Y. Hu, "Energy-efficient mobile robot exploration," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 505–511.

[26] S. Oßwald, M. Bennewitz, W. Burgard, and C. Stachniss, "Speeding-up robot exploration by exploiting background information," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 716–723, 2016.

[27] S. Wirth and J. Pellenz, "Exploration transform: A stable exploring algorithm for robots in rescue environments," in *2007 IEEE International Workshop on Safety, Security and Rescue Robotics*. IEEE, 2007, pp. 1–5.

[28] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters." in *Robotics: Science and systems*, vol. 2, 2005, pp. 65–72.

[29] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 413–14 423, 2020.

[30] O. Michel, "Cyberbotics ltd. webots™: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, p. 5, 2004.

[31] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, and W. Burgard, *Principles of robot motion: theory, algorithms, and implementations*. MIT press, 2005.

[32] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9068–9079.

[33] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv preprint arXiv:1709.06158*, 2017.