SFRE: Safe and Fast Robotic Exploration for 3D Uneven Terrains

Shengkai Liu¹, Runhua Wang¹, Qingchen Bi¹, Guanghui Wen² and Xuebo Zhang¹

Abstract—Exploration strategies of ground robots are often applied in indoor structured environments. However, numerous challenges persist in outdoor 3D unstructured environments, particularly in rough and rugged terrains. This paper proposed an autonomous exploration framework (SFRE) for ground mobile robots in uneven environments. The realization of SFRE can be broken down into three stages. First, the 2D traversability grid map is obtained by analyzing the terrain features of 3D uneven environments. Second, to improve exploration efficiency, we partition the exploration space and utilize the sparrow search algorithm to determine the visit order of each subspace. Finally, to ensure that the robot explores unknown regions safely, a new frontiers selection criterion that combines the height and slope of the frontiers is proposed, which has not been considered in previous methods for frontiers selection. Experiments are conducted to validate the safety and high efficiency of the proposed autonomous exploration framework. All tests show a reduction of 27% in exploration time and 32% in traveling distance compared to the comparative method. (Supplemented video link: https://youtu.be/-eVXv8Zx6VA)

I. INTRODUCTION

Autonomous robotic exploration aims to search for feasible paths, guiding the robot to effectively explore unknown spaces, and create and update its map. Autonomous exploration has many functionalities, such as disaster rescue [1], target search [2], safety inspection [3], and so on. Currently, autonomous exploration technology for twodimensional indoor ground mobile robots [4], [5] is relatively mature. However, achieving autonomous exploration in uneven terrains remains a challenge, which limits the effective implementation of aforementioned functionalities. Therefore, it is of great significance to enable robots to achieve efficient and safe autonomous exploration in uneven terrains.

A. Related Work

In the past few decades, extensive research has been conducted on enabling robots to autonomously explore unknown environments. Many methods have been designed to achieve this goal, with classical methods including frontier-based methods and sampling-based methods.

Frontier-based methods utilize the frontiers of known and unknown regions to guide the robot into unexplored spaces. The core issue lies in selecting the frontiers to explore, that is, determining the order in which frontiers are visited.

¹Institute of Robotics and Automatic Information System, Tianjin Key Laboratory of Intelligent Robotics, College of Artificial Intelligence, Nankai University, Tianjin 300350, P. R. China wrunhua@nankai.edu.cn

²Department of Systems Science, Southeast University, Nanjing 211189,

P. R. China wenguanghui@gmail.com

Greedy selection of the nearest frontier often leads the robot into small unknown spaces, reducing overall exploration efficiency. Many researchers have improved upon greedy strategies. For example, the work in [6] proposed repetitive rechecking method and segmentation of structured indoor environments, which is applicable only to environments composed of several rooms and cannot be generalized to uneven terrains. The work in [5] utilized global and local frontier detector to obtain frontiers, enhancing efficiency and ensuring probabilistic completeness, but it is limited to two-dimensional environments. Moreover, neglecting terrain features associated with frontiers can potentially lead the robot into unsafe spaces in unstructured environments.

Sampling-based methods utilize viewpoints sampling in the free space to guide the robot towards unexplored areas by selecting a branche with the highest utility value. The NBVP method [7] is a typical sampling-based approach that employs Rapidly-exploring Random Trees (RRT) to generate sampled viewpoints and calculates the information gain for each viewpoint. However, this method lacks consideration for global exploration strategies, often resulting in the robot getting trapped in local regions and leading to insufficient exploration. Additionally, the Dual-Stage Viewpoint Planner (DSVP) method [8] extends the RRT more towards the current exploration direction using a biased sampling scheme. The process of evaluating the information gain of nodes by searching for unmapped voxels requires a significant amount of computation in these methods. Moreover, sampling-based methods tend to overlook certain regions, especially narrow areas with small openings, resulting in incomplete exploration. The work in [9] achieves fast exploration path planning with low computational cost, but its application in uneven terrains is limited due to the constraints of its 2D terrain-map functionality, which prevents it from finding frontiers in unstructured areas.

B. Contributions

Inspired by the aforementioned issues, a novel autonomous exploration framework is proposed specifically designed for ground mobile robots operating in uneven terrains. In summary, the main contributions of this work are as follows:

1) A hierarchical exploration strategy called SFRE is proposed, which provides a global route for the robot by partitioning the unknown space and obtaining the visit order of subspaces.

2) A new frontiers selection criterion is proposed to prevent the mobile robots from entering hazardous areas by considering both the height and slope of the frontiers in uneven terrains.

^{*}This work was supported in part by the National Natural Science Foundationof China under Grant 62303247, in part by Natural Science Foundation of Tianjin under Grant 22JCZDJCO0810, and in part by the Fundamental Research Funds for the Central Universities.



Fig. 1. The overview of exploration framework of SFRE

3) Extensive experiments are conducted to validate the superior performance of the proposed framework, demonstrating a reduction of 27% in exploration time and 32% in traveling distance compared to the comparative method.

II. METHODOLOGY

Define $S \subset \mathbb{R}^3$ as the work space to be explored. Let $S_{ob} \subset S$ be the observed known space and $S_{unk} \subset S$ be the current unknown space. As demonstrated in Algorithm.1, 2D traversability grid map is generated by analyzing the flatness, slope, and roughness of the terrain. Then, the multiple rapidly exploring randomized trees are employed to detect frontiers. The exploration space is divided into 9 regions of 3×3 , and the proposed method determines the visiting sequence for each subregion and the frontiers within the subregion. Exploration is considered complete when the number of frontiers to be visited reaches zero.

Algorithm 1 SFRE

1: Constructing a 2D traversability grid map Map 2: Frontiers $FRS \leftarrow$ Frontiers Detection(Map) 3: $SP_{best} \leftarrow Global Subspace Division(SP_0)$ for N = 0, 1, ..., n - 1 do 4: if $p_i \in FRS$ and p_i in sp_N^{best} then 5: Compute the total gain $TG(p_i)$ 6: if $TG(p_i) > BestGain$ then 7: 8: BestGain $\leftarrow TG(p_i)$ end if 9: end if 10: 11: end for if number(FRS) = 0 or time limit then 12: **Exploration Complete** 13: 14: end if

A. Terrain Analysis

The terrain analysis module obtains the normal vectors and flatness of a plane through covariance analysis within a specified fitted plane size range, and then the plane's slope, roughness, and height can be calculated. Based on these data, a 2D traversability grid map is constructed. The method of calculating the normal vector \mathbf{n} of the fitted plane P is presented in [10]. The plane analysis module is designed to obtain information such as the flatness, slope, roughness, and height of a plane, which can be used for constructing a 2D traversability grid map and selecting frontiers in subsequent steps.

(a) **Plane Flatness**: Plane Flatness f represents the scatter of the point cloud for a fitted plane. It can be represented by the surface variation

$$f = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2},\tag{1}$$

where $\lambda_0 \leq \lambda_1 \leq \lambda_2$ are are eigenvalues of the covariance matrix *C* for points in the fitted plane.

(b) **Plane Slope**: Plane Slope *s* reflects the steepness of the fitted plane. *s* represents the angle between the normal vector \mathbf{n} of the fitted plane and the vector $\mathbf{n}_z = [0,0,1]$.

(c) **Plane Roughness**: Plane Roughness r measures the ruggedness of the fitted plane, which is proposed in [10]. The distance between the highest and lowest points in the direction of the normal vector on the fitted plane is projected onto the normal vector as r.

(d) **Plane Height**: Plane Height *h* is the distance between the center point *p* of the fitted plane and the horizontal plane.

Based on this, *s* and *h* are used to select the frontiers for exploration. *f*, *s* and *h* are used to assess the traversability τ of the fitted plane

$$\tau = \begin{cases} 0, & f < f_{\text{crit}} \land s < s_{\text{crit}} \land r < r_{\text{crit}} \\ 1, & \text{otherwise} \end{cases}, \qquad (2)$$

where $f_{\rm crit}$, $s_{\rm crit}$, $r_{\rm crit}$ are the maximum flatness, slope, and roughness thresholds that allow the robot to traverse the fitted plane. When $\tau = 0$, it means the robot can pass through the fitted plane, conversely, when $\tau = 1$, it means the robot has difficulty passing through.

A 2D traversability grid map Map is constructed based on the traversability τ of the fitted plane, similar to the 2D occupancy grid map. When the center of the fitted plane can be projected onto the terrain surface, its traversability can be used to set the value of the corresponding grid. When $\tau = 0$, the grid is set to be free. When $\tau = 1$, the grid is set to be occupied. When the center of the fitted plane cannot be projected onto the terrain surface, the value of the corresponding grid is unknown.

B. Hierarchical Exploration Strategy

After constructing a 2D traversability grid map, the RRT exploration is utilized to obtain frontiers. Then a hierarchical exploration strategy is designed to visit the found frontiers, including global subspace division and local frontiers selection.

1) Frontiers Detection: On a 2D occupancy map, there are already many methods available for detecting frontiers. The work in [5] is a classic method for extracting frontiers on grid maps. Due to its probabilistic completeness and the tendency of RRT to favor unexplored areas, we utilize it to obtain frontiers (*FRS*).



Fig. 2. The 2D traversability grid map and the detection of the frontiers

2) Global Subspace Division: After obtaining the frontiers, the order of visiting these frontiers needs to be determined. Since the number and position of frontiers are varible during the exploration process, sorting all frontiers on the entire 2D traversability grid map requires a significant amount of computation, reducing the efficiency of our method. As shown in Fig. 3, inspired by [11], we divide the entire exploration space into 9 subspaces of 3×3 , each of which corresponds to a different part of the 2D traversability grid map. Then, by determining the visit order of these 9 subspaces, the global guidance for autonomous exploration can be provided.

During the exploration process, it is unnecessary to sort all 9 subspaces every time. We select subspaces with a larger number of frontiers and unknown grid cells as the subspaces worth exploring. Therefore, we only sort the subspaces that are worth exploring, reducing the computational load for subsequent subspace sorting and improving exploration efficiency.

When determining the visit order for the 3D uneven subspaces, instead of simply prioritizing minimizing distance as in solving the TSP problem, we balance exploration efficiency and consistency of exploration routes to ensure relatively short distances. The following requirements are taken into consideration:

- Minimize total length as much as possible;
- Prioritize the visitation of subspaces with a higher number of unknown grid cells;
- Ensure consistency between the newly solved route and the previous route.

Assuming that the subspaces selected are defined as $SP_0 = [sp_0, sp_1 \dots sp_{n-1}]$, we sort the centers $C_0 = [c_0, c_1 \dots c_{n-1}]$ of the grid maps corresponding to each subspace. *n* represents the number of subspaces selected. To meet the aforementioned conditions, we apply the Sparrow Search Algorithm [12] to determine the visitation order for each center. The algorithm initializes with a sparrow population size of *N*, comprising of explorers N_e , followers N_f , and vigilantes N_v in a ratio of 7:2:1. The maximum number of iterations is set

as T_{max} . We define the fitness function as follows:

$$f = \lambda_u F_{unk}(C) + \lambda_s F_{sim}(C) + \lambda_l \sum_{j=1}^{n-1} \left\| c_j - c_{j-1} \right\|, \quad (3)$$

where λ_u , λ_s , and λ_l are constants, $F_{sim}(C)$ is used to evaluate the similarity between the current route and the optimal route to ensure route consistency. We use the method proposed in [11] to calculate $F_{sim}(C)$. $F_{unk}(C)$ is used to calculate the difference between the sum of unknown rates of the selected subspace sequence in the first half and the sum of unknown rates in the second half. $F_{unk}(C)$ can be computed by

$$F_{unk}(C) = \begin{cases} \sum_{j=0}^{\frac{n}{2}-1} u_j - \sum_{j=\frac{n}{2}}^{n-1} u_j, & n\%2 = 0\\ \\ \sum_{j=0}^{\frac{n}{2}} u_j - \sum_{j=\frac{n}{2}+1}^{n-1} u_j, & n\%2 = 1 \end{cases}$$
(4)

where $u_j = \frac{N_{unk}}{N_{all}}$ represents the unknown rate corresponding to the subspace *j*. N_{unk} and N_{all} represent the number of unknown grids and total grids in the traversability grid map corresponding to the subspace, respectively. $F_{unk}(C)$ can ensure that subspaces with a higher unknown rate are prioritized for exploration, thus improving exploration efficiency.

The optimal subspace sequence C_{opt} is initialized as C_0 , and the sparrow population is initialized by randomly swapping the positions of two centers in C_0 . The position space of each sparrow is *n*-dimensional. Calculate the fitness of each sparrow, and then update the position of each sparrow. The update formula for the position of each sparrow is consistent with [11].

When the algorithm completes its iterations, the sequence corresponding to the optimal fitness is $C_{best} = [c_0^{best}, c_1^{best} \dots c_{n-1}^{best}]$, and thus the optimal subspace access order is $SP_{best} = [sp_0^{best}, sp_1^{best} \dots sp_{n-1}^{best}]$. The next step is to determine the selection criteria for the frontiers of each subspace.



Fig. 3. The result of global subspace division, with darker colors indicates earlier visits to the subspace.

3) Local Frontiers Selection: After obtaining the visit order for the global subspaces, the criteria need to be established for selecting the frontiers within the subspace. In contrast to the exploration of 2D structured environments, we cannot solely prioritize maximizing the information gain, as this may lead the robot into hazardous areas. With the premise of ensuring safety and exploration efficiency, we have combined the terrain features near the frontiers to propose the following criteria:

- Ensure the safety of the robot exploration process;
- Minimize the length of the exploration path as much as possible;
- Efficiently explore information about unknown environments.

By considering the above factors, four indicators including frontier height, frontier slope, information gain, and exploration distance, have been defined. The definitions of these indicators will be detailed below.

(a) **Frontier Height**: The frontier height $FH(p_i)$ is calculated based on the four-connected region near the grid where the frontier point p_i is located. $FH(p_i)$ is the average height of the fitted planes corresponding to the free grid states within the four-connected region.

In order to ensure that the robot first explores relatively safe spaces while also identifying some non-traversable regions, we prioritize the robot to visit frontiers with lower heights, which is a significant difference compared with other exploration methods.

(b) **Frontier Slope**: The calculation method for frontier slope $FS(p_i)$ is similar to that of frontier height $FH(p_i)$. $FH(p_i)$ is the average slope of the fitted planes corresponding to the free grid states within the four-connected region.

A smaller $FS(p_i)$ indicates relatively flat terrain near the frontiers, which is preferable for the robot to navigate. On the other hand, a larger $FS(p_i)$ value suggests steeper terrain near the frontiers, which is not conducive to safe exploration by the robot.

(c) **Information Gain**: Information Gain $IG(p_i)$ is a measure of the size of the unknown space near the frontiers. We use the number of other frontiers within a fixed radius near the frontier point to represent the information gain. During the exploration process, frontiers with higher information gain have higher priority.

(d) **Exploration Distance**: Exploration Distance $ED(p_i)$ is the distance between the current position of the robot and the frontier point. Generally, the Euclidean distance is used to represent the exploration distance. However, this is often inaccurate since there are usually non-traversable terrain or obstacles between the robot and the frontier point, causing the actual distance traveled by the robot to be greater than the Euclidean distance.



Fig. 4. Exploration Distance diagram. On the left: the green line represents Euclidean distance, the blue line represents the distance calculated by A*, on the right: the actual movement distance of the car.

As shown in the Fig. 4, in uneven terrain, even if the

Euclidean distance between the position of the robot and the frontier point is small, the robot may need to travel a long distance to reach the frontier point in actuality. Therefore, we use the A^{*} algorithm to calculate the distance between the robot and the frontier point on the 2D traversability grid map, which approximates the exploration distance. After obtaining the above indicators, we can calculate the total gain $TG(p_i)$ of each frontier point in a subspace by

$$TG(p_i) = \frac{\lambda_i IG(p_i) - \lambda_d ED(p_i)}{\lambda_s FS(p_i) + \lambda_h FH(p_i)},$$
(5)

where λ_i , λ_d , λ_s and λ_h are coefficients of four indicators, ensuring that they are of the same order of magnitude. We then select the point with the biggest total gain *BestGain* as our target point, send it to the motion planning module.

C. Supporting Modules

Fig. 1 shows the overview of the SFRE. Autonomous exploration tasks in uneven environments require the assistance of multiple modules. In addition to the design of exploration strategies, SLAM and motion planning are also indispensable parts. A-LOAM [13] is a classic 3D SLAM algorithm that can be used to construct 3D grid maps. PUTN [14] can achieve robot motion planning in uneven terrains due to its stability and safety.

III. EXPERIMENTS AND RESULTS

In this section, we conduct experiments to validate the superiority of the proposed exploration strategy. The scout 2.0 is used as the ground vehicle platform for the simulations. To ensure the smooth operation of the vehicle on uneven terrain, the maximum speed is set to 0.5m/s. The simulations are conducted on an Intel Core i9-13900 HX CPU and 16 GB RAM.

A. Simulation Experiments

Multi-RRT [5] is a classic 2D exploration method, and we use it as a comparative method for exploration on a 2D traversability grid map. This can verify the following two points: (1) to what extent the global subspace division improves exploration efficiency, (2) whether the selection of frontiers combined with terrain features ensures safety.

We have created three types of environments. Fig. 5(a) resembles a valley, Fig. 5(b) resembles an uneven outdoor environment, and Fig. 5(c) incorporates a cliff on top of Fig. 5(b) to test the safety of our algorithm.

By comparing our method with the comparative method, we evaluate the exploration capability of our method. Each method is run to obtain results from 10 complete explorations, with the stopping criteria being either the vehicle almost coming to a stop or reaching the time limit. The time limit is set to 10 minutes.

Exploration Rate: Table I presents the statistical data of the two methods in the simulation experiments. All indicators are the average values of 10 experiments. ε_1 represents the average explored volume divided by the average time, while ε_2 represents the average explored volume divided by the



(a) The uneven scene 1

(b) The uneven scene 2

(c) The uneven scene 3

Fig. 5. The simulation scenes and the trajectories generated by the two methods in them. The blue circle represents the starting point, and the red circle represents the endpoint.



Fig. 6. Comparison of exploration progress of the two methods in the three environments depicted in Fig. 5. The mean and standard deviation of successful exploration for 10 tests are shown.

 TABLE I

 Results of Simulations in Three Environments

scene size(m)	method	exploration time (s)		traveling distance (m)		ε_1	ε_2
		avg	std	avg	std	avg	avg
Scene1	Multi-RRT	431.66	74.63	156.99	26.87	1.88	5.16
30×30	Proposed	298.51	27.43	107.05	11.03	2.71	7.57
Scene2 20×20	Multi-RRT	220.57	17.31	84.07	8.66	2.05	5.41
	Proposed	158.95	12.72	55.75	3.84	2.89	8.24
Scene3	Multi-RRT	334.72	49.37	126.11	17.51	1.68	4.44
25×25	Proposed	262.54	13.51	88.67	6.34	2.15	6.38

average traveling distance. For a more intuitive display of the superiority of our method, Fig. 6 illustrates the exploration progress curves of the two methods in the three scenes. From Table I, it can be observed that our method significantly outperforms the comparative method in exploration speed. In all three scenes, our exploration time is reduced by approximately 27%, the distance traveled is decreased by around 32%, and ε_1 and ε_2 are improved by about 37% and 47%, respectively.

By considering path length, the extent of unknown areas within subspaces, and path consistency simultaneously, our method systematically visits each subspace to ensure completion of exploration before moving to the next, thereby avoiding backtracking caused by leftover unexplored spaces. In contrast, the comparative method only considers the size of unknown areas near frontiers and the distance traveled, leading to greedy selection of frontiers and resulting in wastage of time and path. Each scene depicted in Fig. 5 displays the trajectories of the two methods. Our method is able to systematically complete exploration of unknown environments without generating a large number of chaotic and disorderly paths.



Fig. 7. The robot falls off a cliff in scene 3 using Multi-RRT.

From Fig. 6, it can be observed that the volume we explored is slightly higher than that of the comparative method, and our exploration rate is also slightly higher. It is worth noting that in scene 3, the early exploration rate of the comparative method is higher than ours, which is due to the fact that the comparative method does not consider the height of frontiers and may initially ascend higher slopes, exploring more unknown spaces. However, this may also lead to the robot entering hazardous areas.



(a) The outdoor uneven scene 1



(b) The outdoor uneven scene 2

Fig. 8. Results of real world experiments.

Exploration Safety: In scene 1 and scene 2, our method and the comparative method complete exploration in the first 10 test runs. However, during testing in scene 3, we find that the comparative method does not consistently achieve successful exploration, with only 5 out of 10 test runs being successful. As shown in Fig. 7, due to the presence of frontiers at the cliff, the robot is guided into hazardous areas, thus leading to the exploration failure. After conducting 23 experiments, we obtain 10 instances of successful outcomes for the comparative method, as depicted in Fig. 6(c). Since our method prioritizes selecting frontiers with lower heights, it guides the robot to explore the unknown areas from the safe side of the cliff, enabling it to discover the terrain features of the cliff and then transform the unknown grid cells in the traversability grid map into non-traversable ones. With this strategy, the existence of frontiers at the cliff is effectively avoided. Our method completes exploration in the first 10 test runs, validating the safety of the proposed method.

B. Real-World Experiments

A Scout 2.0 is used as an exploration platform, and it is equipped with an NUC (NUC11PH, with an Intel Core i7-1165G7 CPU and 32 GB RAM) and a Livox MID360. The robot exploration range of the two experiments is limited to $18 \times 17 \text{ m}^2$ and $6 \times 26 \text{ m}^2$ respectively. As shown in Fig. 8, the starting point is marked with a blue circle, and the white line represents the trajectory of the robot. The experimental results show that the proposed method can safely and efficiently complete the exploration of uneven terrains, verifying the feasibility of the proposed method.

IV. CONCLUSION

In this paper, a hierarchical exploration framework for uneven environments is proposed. Firstly, terrain features are obtained through the terrain analysis of the fitted plane, and a 2D traversability grid map is constructed to explore frontiers. Then, the global subspace is divided, and a global exploration route is obtained through SSA. Finally, by combining the terrain features such as height and slope to select frontiers, they are used to guide robots into unknown spaces. Extensive simulations in multiple scenes validate the effectiveness and safety of our method in exploring uneven terrain. All tests show a reduction of 27% in exploration time and 32% in traveling distance compared to the comparative method.

References

- N. A. Sánchez O., Investigation of fundamental algorithms for cooperative exploration of mobile agents in search and rescue scenarios, in 2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C), pp. 14–15.
- [2] X. Cao, C. Sun, and M. Yan, Target search control of auv in underwater environment with deep reinforcement learning, *IEEE Access*, vol. 7, pp. 96549–96559, 2019.
- [3] M. Dharmadhikari, P. De Petris, M. Kulkarni, N. Khedekar, H. Nguyen, A. E. Stene, E. Sjøvold, K. Solheim, B. Gussiaas, and K. Alexis, Autonomous exploration and general visual inspection of ship ballast water tanks using aerial robots, *arXiv preprint arXiv:2311.03838*, 2023.
- [4] Q. Bi, X. Zhang, J. Wen, Z. Pan, S. Zhang, R. Wang, and J. Yuan, CURE: A hierarchical framework for multi-robot autonomous exploration inspired by centroids of unknown regions, *IEEE Transactions* on Automation Science and Engineering, no. 99, pp. 1–14, 2023.
- [5] 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.
- [6] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, Evaluating the efficiency of frontier-based exploration strategies, in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. VDE, 2010, pp. 1–8.
- [7] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, Receding horizon" next-best-view" planner for 3d exploration, in 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 1462–1468.
- [8] H. Zhu, C. Cao, Y. Xia, S. Scherer, J. Zhang, and W. Wang, DSVP: Dual-stage viewpoint planner for rapid exploration by dynamic expansion, in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021, pp. 7623–7630.
- [9] J. Huang, B. Zhou, Z. Fan, Y. Zhu, Y. Jie, L. Li, and H. Cheng, FAEL: Fast autonomous exploration for large-scale environments with a mobile robot, *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1667–1674, 2023.
- [10] Q. Bi, X. Zhang, S. Zhang, Z. Pan, and R. Wang, TMPU: A framework for terrain traversability mapping and planning in uneven and unstructured environments, in 2023 42nd Chinese Control Conference (CCC), 2023, pp. 01–07.
- [11] X. Zhao, C. Yu, E. Xu, and Y. Liu, TDLE: 2-D lidar exploration with hierarchical planning using regional division, in 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE), 2023, pp. 1–6.
- [12] J. Xue and B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, *Systems science & control engineering*, vol. 8, no. 1, pp. 22–34, 2020.
- [13] J. Zhang and S. Singh, LOAM: Lidar odometry and mapping in realtime. in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.
- [14] Z. Jian, Z. Lu, X. Zhou, B. Lan, A. Xiao, X. Wang, and B. Liang, PUTN: A plane-fitting based uneven terrain navigation framework, in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022, pp. 7160–7166.