# Mode-Switching Frontier Search for Robust Autonomous Exploration in Complex Terrain

Jinhwan Seo[1], Chanmi Lee[1], Hyunsik Son[2], and Sung-eui Yoon[1]

[1]School of Computing, KAIST (Korea Advanced Institute of Science and Technology), Daejeon 34141, Republic of Korea
{jinhwan.seo, chanmi99, sungeui}@kaist.ac.kr
https://sgvr.kaist.ac.kr
[2]Hanwha Aerospace, 6, Pangyo-ro 319 beongil, Bundang-gu, Seongnam-si, Gyeonggi-do, 13488, Republic of Korea. hyunsik.son@hanwha.com

**Abstract.** Autonomous exploration in unstructured environments is often hindered when robots become trapped in navigational deadlocks, causing mission failure. Frontier-based planners, in particular, often lack a strategic mechanism to recover when all immediate paths are blocked. This limitation often leads to a state of 'navigation at a dead end', where an autonomous agent without a recovery strategy can become permanently stuck. The framework to detect such impasses, leverage memory of past states, and strategically recover is therefore a critical and often missing component for robust autonomous agents To address a critical failure mode in frontier-based exploration where robots become trapped in navigational deadlocks, we introduce a state-aware supervisory architecture. This framework enhances the resilience of standard local planners by integrating a robust recovery mechanism that systematically switches the robot between Search, Recovery, and Exploration modes. Upon detecting a stuck state by monitoring its trajectory history, the system enters a dedicated recovery mode. In this mode, it consults a global database of previously discovered frontiers—acting as a short-term memory—to select an optimal recovery point and escape the impasse. We validated our framework in challenging simulation scenarios where a standard local planner fails. Experimental results show that our framework transforms a local planner with a 0% success rate into a resilient system achieving up to a 100% success rate, demonstrating its effectiveness as a critical component for robust autonomous navigation.

**Keywords:** Autonomous Exploration · Frontier-based Exploration · Resilient Navigation.

## 1 Introduction

The development of autonomous ground robots capable of long-term exploration in inaccessible off-road environments offers transformative potential. A core capability for these missions is navigating toward a global goal through unmapped areas using only local sensors. In such scenarios, robots frequently encounter dead-ends or complex terrain that render immediate forward paths untraversable.

Frontier-based exploration [13] is a common paradigm for this task, extending the map by guiding the robot to the boundary between explored and unexplored space. However, a key limitation of this approach is its reliance on the availability of traversable frontiers. When a robot enters a concave region or a dead-end, all frontiers may become unreachable, leading to repetitive planning cycles and eventual mission failure. Many existing methods, including recent advancements using Gaussian Process (GPs) [1, 9] to model terrain traversability, tend to focus on identifying which frontier to approach next, suggesting limited strategies for scenarios where no frontier can be reached. This can result in repetitive, ineffective planning cycles within local minima which result in mission failure. Autonomous agents often lack robust recovery mechanisms may remain trapped in sub-optimal states. In GPS-denied scenarios without global context, such path-blockage events tend to recur, often resulting in local minima situations where the robot repeatedly attempts unproductive paths [6, 14]. This leads to stalled navigation, inefficient movements, and increased energy consumption and mission time, ultimately preventing the robot from reaching its destination. This limitation often leads to a state of navigation at dead end, where an autonomous agent without a recovery strategy can become permanently stuck, consequently leading to mission failure. The framework to recover from such impasses is therefore a critical component of robust autonomous agent. Our approach provides mode switch that recognize the deadlock state and leverage a memory of previously explored, safe location; finally, a strategic recovery to one of these points to resume exploration.

In this work, we present a framework that complements the optimization of forward progression with navigational resilience. Our framework is built upon a state-aware control architecture designed to systematically detect and recover from navigational dead end. By maintaining a global history of previously discovered frontiers, our system creates a spatial memory. This memory is not just a record of the past, but a strategic global management that allows the robot to make informed decisions not only about where to go next, but also about where to recover to when forward motion is impeded.

Our main contributions can be described as follows:

- The design of a practical, state-aware architecture that integrates three operational modes to systematically handle navigational deadlocks, a common point of failure for standard frontier-based planners.
- A global frontier management system that serves as a strategic spatial memory, enabling intelligent recovery to previously discovered vantage points, as opposed to simple, undirected backtracking.
- A demonstration through simulation that this supervisory module transforms a standard local planner with a 0% success rate into a resilient system achieving up to 100% success, proving its effectiveness as a critical component for robust navigation.

## 2 Related Works

Autonomous exploration is a fundamental task for mobile robots, and a variety of methods have been developed to enable efficient navigation and mapping of unknown environments. These approaches can be broadly categorized based on their underlying principles, ranging from geometric and topological strategies to advanced probabilistic and learning-based models.

**Graph-based approach** Graph-based exploration methods [5, 11] are another class of strategies that represent the environment as a topological graph, emphasizing the connectivity between regions rather than precise metric details. These approaches are particularly well-suited for planning in large-scale and complex environments, such as tunnels or subterranean spaces. These methods typically build roadmaps incrementally or define topological nodes using techniques such as Voronoi diagrams [2]. For instance, GBPlanner [5] utilizes a bifurcated architecture with a local and a global planner, constructing exploring random graphs within a local subspace and a sparse global graph for repositioning. In addition to single-robot scenarios, graph-based methods are highly effective for multi-robot systems. Space partitioning techniques, such as Voronoi diagrams, can be used to assign distinct regions to individual robots, thereby preventing redundant exploration and maximizing collaborative efficiency. This category also frequently employs hierarchical planning frameworks that decompose large-scale exploration problems into smaller subregions to manage complexity and provide global guidance.

**Gaussian process (GP) based approach** Gaussian Process (GP) based approaches [1, 9] leverage advanced probabilistic models to address challenges such as uncertainty and complex environments. These methods provide a powerful way to model continuous spatial phenomena and quantify uncertainty. Unlike traditional discrete grid maps, GPs [9] inherently account for spatial correlation and continuity, offering a more nuanced representation of environmental data such as elevation or occupancy. A key challenge with standard GPs is their high computational complexity, which can be prohibitive for real-time applications and large datasets. To mitigate this, Sparse Gaussian Processes (SGPs) are frequently employed, reducing the computational burden while maintaining effective performance. GP-based methods have diverse applications in exploration. They can be used for traversability analysis on uneven terrain, accurately identifying navigable free space. Furthermore, they are effective for assessing global exploration quality by generating uncertainty maps, where high variance regions can be defined as GP frontiers that guide navigation toward unexplored open spaces. This allows robots to make more informed decisions about where to explore next, prioritizing areas with the highest potential for new information.

**Frontier-based approach** Frontier-based exploration is a prominent class of strategies that leverages the concept of frontiers, which are defined as regions on the boundary between known and unexplored space [13]. The core principle is that a robot can maximize new information by navigating to these boundaries, thereby expanding its map into new territory. Pioneered by Yamauchi [13], traditional frontier-based methods direct the robot to the nearest accessible, unvisited frontier in an occupancy grid map. While effective, this approach can suffer from a decrease in efficiency with increasing map size and a potential for local optima. To address these limitations, more efficient frontier detection methods, such as the Wavefront Frontier Detector (WFD) [7] and Fast Frontier Detector (FFD) [7], have been developed. Many advanced approaches improve upon traditional frontier-based methods [3, 8–10] by incorporating information theory. These information-driven enhancements evaluate the information gain associated with each frontier, assessing how much new information a robot is expected to acquire by exploring a particular area. For instance, Long et al. [10] proposed the HPHS method, which employs a heuristic gain function that considers factors like traveling cost, robot orientation, and information gain to select the optimal target frontier, thereby enhancing multi-robot cooperation. Furthermore, Bi et al. [3] introduced CURE, which utilizes a parameter-insensitive utility function to guide robots to unexplored areas efficiently and prevent oscillation between targets.

While graph-based methods excel at topological understanding, GP-based models at handling uncertainty, and frontier-based strategies at providing intuitive information gain, they all share a fundamental assumption: the existence of a reachable next goal. When a robot enters a state of deadlock, such as in a complex tunnel where all paths are blocked or all known frontiers are unreachable, these methodologies fail to provide a strategy to continue navigation. This research aims to address this specific limitation by proposing a new framework that goes beyond simply 'selecting' the next optimal goal to also 'detecting' a navigationally stalled state and intelligently 'recovering' from it.

## 3 Method

The proposed framework is designed to facilitate robust and efficient navigation for an autonomous agent in complex environments. Our framework centers on a state-aware frontier search that handles robot's decision-making and switch between three modes based on its situation: Search, Recovery, and Exploration. A global frontier database supports this system by storing information about previously visited areas. This database helps the robot make better decisions by combining current sensor data with past experience. The overall architecture is shown in Fig. 1.

### 3.1 Frontier Search

**Cost Map Filtering** Our frontier search framework utilizes a unified cost map generated by combining three risk measurements, inclination, collision and steepness, from robot's local sensor. Each cell $c$ in the grid is assigned a cost $Cost(c)$ that categorizes three costs into one of three states: traversable, untraversable, or unknown. Let $v_{inc}(c), v_{col}(c)$, and $v_{steep}(c)$ be the risk values for inclination, collision, and steepness for cell $c$. A cell is classified as untraversable $Cost(c) = 1$ if any of its risk values exceed pre-defined safety thresholds ($\tau_{inc}(c), \tau_{col}(c)$ and

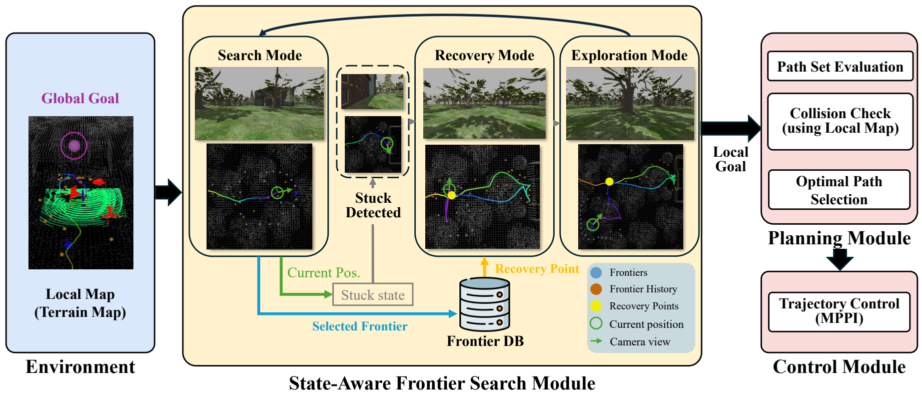


**Fig. 1.** The proposed architecture enables robust autonomous exploration in challenging environments. At its core, the state-aware frontier search module uses a dynamic mode-switching mechanism to offer the robot's behavior across three states: Search, Recovery, and Exploration. To maintain continuous progress, this mechanism consults a frontier database to select a recovery point whenever a stuck event is detected. After frontier search, the local frontier goal is forwarded to the Planning Module for optimization and finally executed by the Control Module using an Model Predictive Path Integral(MPPI) [12] trajectory controller.

$\tau_{steep}(c)$). If a cell has no sensor information (NaN), it is classified as unknown. All other cells are defined traversable ($Cost(c) < 1$), with lower values indicating safe terrain. This filtering process transforms raw data into a clear, traversable map where the boundaries between known, safe space and unknown territory can be explicitly identified for frontier search.

**Frontier Candidates** The system generates new exploration targets called frontier candidates using local sensor data. It searches in a 360-degree radial around the robot to find the farthest reachable points. These points are grouped together to create frontier candidates. Each candidate is scored based on how much new area the robot could see from that location. This score helps the robot choose targets that will discover the most unknown terrain.

**Global Frontier Management** To maintain short-term spatial information without incurring prohibitive computational costs, a sparse but topologically significant record of frontiers is maintained in a global frontier database. New frontiers are updated into global database only if they are sufficiently distant from existing entries, ensuring memory efficiency and a diverse set of strategic options. Moreover, this database functions as the system's short-term memory, annotating visited frontiers with state information (e.g., exploration value) that becomes instrumental during recovery operations.

### 3.2 Mode Switch

The robot's strategic behavior is managed by its state, which transitions between modes in response to a quantitative assessment of navigational progress.

**Search Mode** The search mode serves as the robot's default operational state. The primary objective is to balance progress toward the global goal with systematic information acquisition. The optimal frontier target $f^*$ is selected through a evaluation function that has three key criteria: proximity to the global goal $D(f^i)$, and expected information gain $E(f^i)$. The goal proximity score identifies frontiers that are closer to the global goal and information gain score quantifies the expected new area that will be mapped from a frontier location. The system maintains this mode while demonstrating consistent navigational progress.

$$D(f^i) = \sqrt{(x_f - x_G)^2 + (y_f - y_G)^2}, \tag{1}$$

$$I(f^i) = \sum_{c \in V(f)} \mathbb{I}(c \in Cost(c) < 1), \tag{2}$$

$$f^* = \texttt{argmax}_{f^i \in F}(\alpha_1 D(f^i) + \alpha_2 I(f^i)) \tag{3}$$

where $D(\cdot)$ is goal proximity score and $I(\cdot)$ is information gain score. Here, $\mathbb{I}(\cdot)$ is indicator function, which is 1 if cell $c$ is in the set of unknown cells, and 0 otherwise.

The system identifies stuck states through a trajectory history analysis rather than relying on single planning failures. It maintains a sliding window of recent robot positions and calculates their geometric mean. Our method identifies a **stuck state** through a trajectory history analysis rather than relying on single planning failures. The system maintains a sliding window of the robot's recent positions and calculates their geometric mean. A stuck state is then declared when the robot remains within a defined radius of this mean position during pre-defined duration, indicating that insufficient navigational progress has been made.

**Recovery Mode** When a stuck state is detected, the system transitions to Recovery Mode. The objective shifts from frontier search to selecting an optimal recovery location from the global frontier database. The selected recovery point $h^*$ facilitates strategic navigation away from the problematic area. The primary objective in this mode is to strategically navigate away from the problematic area. To achieve this, the selection algorithm prioritizes historical frontiers that maximize divergence from the robot's recent, stuck trajectory. We use the stored information gain, as a powerful proxy for this objective. The rationale is that a frontier with a high historical information gain was likely a vantage point overlooking a large, unexplored area, such as a major junction or open space. By selecting the point with the highest information gain we guide the robot back to a promising location for resuming exploration, effectively escaping the local minimum.

$$h^* = \texttt{argmax}_{h \in F_h} I_{exp}(h) \quad \text{subject to} \quad \sqrt{(x_h - x_{P_{st}})^2 + (y_h - y_{P_{st}})^2} \tag{4}$$

where $F_h$ is stored frontier history at frontier database and $I_{exp}(h)$ corresponds to the information gain score that was calculated and stored when $h$ was first discovered.

**Exploration Mode** Upon reaching recovery point, the system enters exploration mode. This mode prioritizes finding new target frontiers that potentially explore new area. The frontier selection algorithm in exploration mode is modified to penalize candidates near the recent stuck point, encouraging exploration away from the problematic area. This reduces the probability of encountering the same navigational obstacle. When the robot achieves sufficient distance from the failure location, the system returns to search mode to continue the primary mission.

$$E(f^i) = \sqrt{(x_f - x_{P_{st}})^2 + (y_f - y_{P_{st}})^2} \tag{5}$$

$$I(f^i) = \sum_{c \in V(f)} \mathbb{I}(c \in Cost(c) < 1), \tag{6}$$

$$f^*_{exp} = \texttt{argmax}_{f^i \in F_h} (\beta_1 E(f^i) + \beta_2 I(f^i)) \tag{7}$$

where $E(\cdot)$ is escape score which represents distance from the stuck position and $I(f^i)$ is stored information gain during search mode, measuring the potential for new map discovery.

# 4 Experimental Result

## 4.1 Experimental Setup

All experiments were conducted using the Autonomous Exploration Development Environment (AEDE) framework [4]. To provide a more comprehensive analysis, we compared the performance of four different planning and frontier methods with a detailed ablation study of our recovery framework.

**Baseline**: Local Planner. This is the default local planner in AEDE, which embodies a standard frontier-based approach that aggressively seeks forward progress but lacks a dedicated deadlock recovery mechanism.

**Baseline + Frontier**. This method enhances the basic local planner by incorporating a standard frontier-selection mechanism. While capable of identifying new areas for exploration, it does not have a strategy for recovering from navigational deadlocks where no immediate frontiers are reachable.

**Baseline + Frontier (Step)**. This method builds upon the "Local + Frontier" approach by adding a simple, heuristic-based recovery mechanism. When a stuck state is detected, the robot is programmed to backtrack a fixed number of steps (n-steps) along its previous path before attempting to replan.

**Proposed Framework** Our full method consists of integrating our state-aware frontier search module on top of the same Base System. This direct comparison allows us to perform an ablation study, quantifying precisely what level of resilience and success our module contributes over planners with no recovery or simple heuristic recovery.

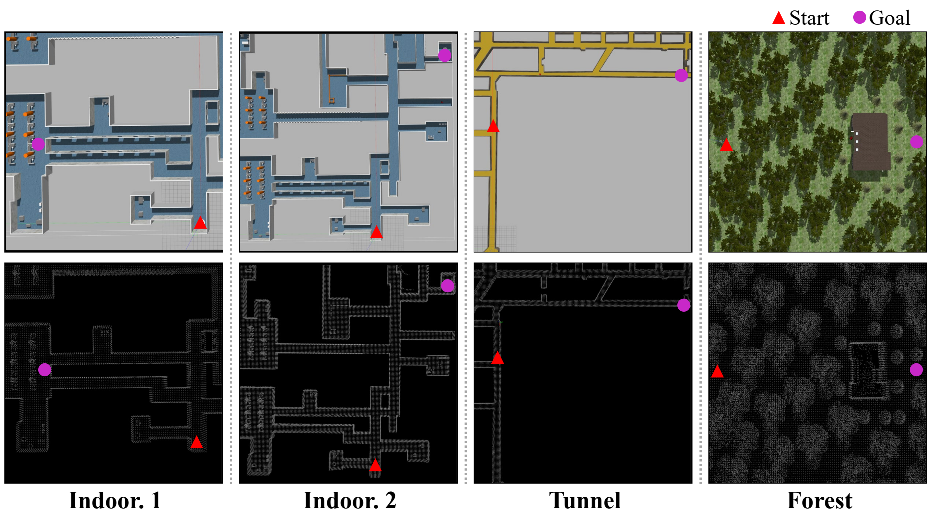


**Fig. 2.** The four maps used in the experiments. Map1: Indoor, Map2: Indoor, Map3: Tunnel, Map4: Forest.

### 4.2 Evaluation Metrics

To quantitatively assess and compare the performance of the proposed method against the baseline, we utilized the following metrics. A time limit of 180 seconds was set for each trial; exceeding this limit was considered a failure. For failed trials, only the success rate and the number of stuck points were recorded, as metrics like distance and area are not meaningful.

**Time (s)**: The total elapsed time from the start of the mission until the robot reached the goal. This was measured only for successful trials.

**Distance (m)**: The cumulative path length traversed by the robot. This was also measured only for successful trials.

**Explored Area** ($m^2$): The total area mapped by the robot during its navigation. This was measured for successful trials to gauge exploration efficiency.

**Success Rate (%)**: The percentage of trials in which the robot successfully reached the global goal within the predefined time limit of 180 seconds.

**Stuck Points (no.)**: The average number of times the robot encountered a situation where it could not proceed and a recovery action was needed. For the baseline method, a single stuck point resulted in mission failure (hence, a value of 1). For our proposed method, this metric represents the average number of times the Recovery Point mechanism was successfully activated.

### 4.3 Quantitative Results

To demonstrate the effectiveness of our proposed method in complex environment, we compare the success rate of baseline and our proposed method across four different scenarios. The results, summarized in Tab. 1, demonstrate the robustness of our framework in deadlock scenario. The baseline method failed to complete the mission in any trial across all four maps, resulting in a 0% success rate. This is because it could not recover after encountering a stuck point and exceeded the 180-seconds time limit. For a more detailed analysis on the Indoor. 1 map, we introduced intermediate methods. By adding a frontier-based exploration mechanism (+Frontier), the system gained a sense of global direction toward the goal. This was sufficient to solve some navigational challenges, but it still failed in more difficult cases like the Indoor. 1, Indoor. 2, and Forest maps, where simply moving toward the next frontier was not enough to escape complex traps. The "+ Frontier (Step)" method, which implements a simple recovery heuristic of backtracking a fixed number of steps before attempting to find a new frontier. This strategy further improved the success rate, as it allowed the robot to escape minor deadlocks. However, it consistently failed in scenarios that required a significant detour or a complete reversal of its path to find an alternative route. In contrast, our method achieved high success rate of 90% in Indoor. 1, 70% in Indoor. 2 and 100% of success rate in complex tunnel and forest environments. When the robot's trajectory history indicates a lack of forward progress, it identifies this as a stuck state and transition to recovery mode. It then selects a recovery point from its stored frontier history and backtracks to that location. To better understand this limitation, we analyzed the 30% of failed trials in the Indoor. 2 map. In these scenarios, the recovery point with the highest historical information gain was typically a large, open intersection. While this point offered multiple new paths, it was still located within the larger, maze-like structure of the environment. Consequently, the robot would successfully escape the initial dead-end but its subsequent exploration choices would lead it into a different dead-end within the same complex area, eventually exceeding the mission time limit. This highlights a limitation of our local-recovery approach in globally complex 'trap' environments and suggests an area for future work in topological map-based recovery. The rationale for selecting the historical frontier with the highest information gain is that such a point was likely a vantage point overlooking a large, open area (e.g., a major junction). By returning to a location of high topological significance, the robot has a greater diversity of escape routes compared to simply reversing its path. This strategic choice is intended to maximize the probability of escaping the local minimum entirely, rather than just temporarily backing away from it. An interesting observation was made in the Tunnel map. In this environment, where paths were not completely blocked, the +Frontier method alone was sufficient to navigate

**Table 1.** Comparison of performance for Map1: Indoor, Map2: Indoor, Map3: Tunnel, Map4: Forest.

| Map | Goal | Method | Time (s) | Distance (m) | Area ($m^2$) | Succ. (%) | Stuck Pts. (no.) |
|---|---|---|---|---|---|---|---|
| Indoor. 1 | (20,40) | Baseline | 180.00 | - | - | 0 | - |
| | | + Frontier | 37.00 | 58.07 | 928.87 | 10 | - |
| | | + Frontier (Step) | 77.99 | 115.78 | 1049.94 | 70 | - |
| | | + Frontier (Recovery) | 83.87 | 122.93 | 1059.60 | 90 | 1.66 |
| Indoor. 2 | (72, -28) | Baseline | 180.00 | - | - | 0 | - |
| | | + Frontier | 180.00 | - | - | 0 | - |
| | | + Frontier (Step) | 76.06 | 120.15 | 1565.02 | 50 | - |
| | | + Frontier (Recovery) | 97.66 | 159.02 | 1591.85 | 70 | 0.71 |
| Tunnel | (21.83, -79.47) | Baseline | 180.00 | - | - | 0 | - |
| | | + Frontier | 56.37 | 102.02 | 856.48 | 100 | - |
| | | + Frontier (Step) | 66.20 | 122.26 | 909.73 | 100 | - |
| | | + Frontier (Recovery) | 97.51 | 154.62 | 977.46 | 100 | 1.30 |
| Forest | (0.27, -48.32) | Baseline | 180.00 | - | - | 0 | - |
| | | + Frontier | 180.00 | - | - | 0 | - |
| | | + Frontier (Step) | 116.58 | 148.27 | 8202.62 | 10 | - |
| | | + Frontier (Recovery) | 103.87 | 144.25 | 8368.76 | 100 | 1.60 |

around obstacles, achieving the same 100% success rate as our full method but in significantly less time (56.37s vs. 97.51s). This suggests that in simpler deadlock scenarios, our method's tendency to select a recovery point far from the stuck position can introduce a time inefficiency. While crucial for escaping severe traps, this long-distance backtracking may not be optimal when a smaller, local maneuver would suffice.

### 4.4 Qualitative Results

For the qualitative results, we performed both simulation and real-world experiments to demonstrate the effectiveness of our proposed framework in complex environment. To validate our method, we compare its performance against baseline approach which utilizes the local planner provided by AEDE.

**Simulation** Fig. 3 illustrates comparison of trajectories from the baseline and our proposed method in deadlock scenario. The results in Fig. 3 highlights its capability to escape deadlocks where baseline method fails. In the figure, the red triangle denotes starting position, and purple circle represents target goal. The experimental setups were designed to include dead-ends and complex intersections, forcing the robot to navigate challenging spaces rather than simple straight paths. For the baseline method, it encounters a complex junction or a dead-end, at which point it enters a permanent stuck state, leading to mission failure. For instance, in the **indoor. 1, 2** scenario, baseline method that selects a local goal based on the closest distance to the target goal will get trapped in stuck point (blue square). Our method, however, recognizes the lack of forward progress and transitions to recovery mode. Upon detecting navigational impasse, the system consults its frontier history to select an optimal recovery point. This point, $h^*$, is chosen by maximizing the traversable area when robot reached. As

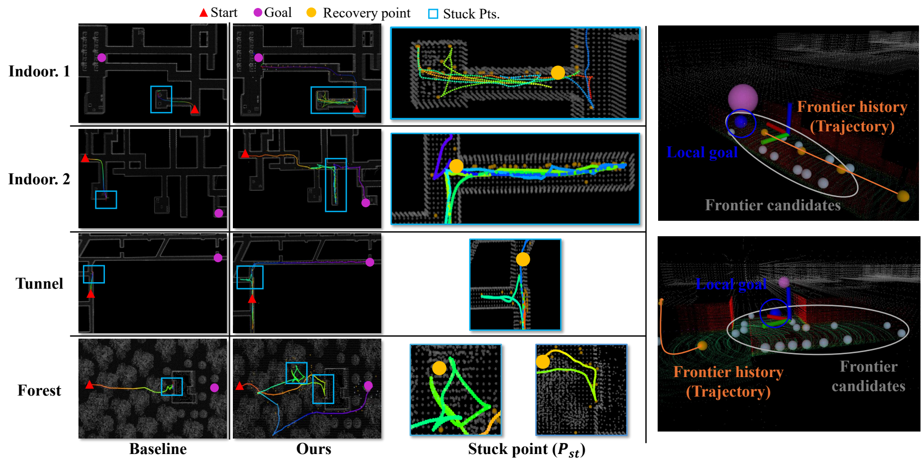


**Fig. 3.** Path picture for baseline, ours 1 (with stuck point 1), ours 2 (with more than 2 stuck point).

illustrated by the trajectory in the third column in Fig. 3, the robot initially proceeds into a dead-end where it cannot find a path forward. After recognizing the impasse, it backtracks to the recovery point (yellow circle) to find and follow a new route toward the goal. Once it reaches recovery point, the robot transitions to exploration mode. It alters the frontier selection criteria to penalize candidates in proximity to the recent stuck point, which encourages the exploration of new direction and minimizes the risk of re-entering the same impassable area. Similar to the indoor environment, we observed that our method escapes a local minimum induced by dense obstacles by backtracking to a more open area to find new path forward in **forest** environment. In the fourth column of Fig. 3, we can observe the frontier candidates (grey circle) in 360-degree radius around its position, while its past movements are tracked as a trajectory or frontier history (orange circle). The next local goal (blue circle) is selected among frontier candidates based on an evaluation that prioritizes proximity to the target goal.

**Real-World Experiment** To demonstrate the practical applicability of our framework, we conducted experiments in a real-world off-road environment. The test site was a mountainous region characterized by significant elevation changes, dense foliage, and narrow path. Operating in this unstructured setting introduced sensor noise from the surrounding trees and complex terrain geometry. A consequence of this was the observation that the generated frontier candidates lacked consistency, appearing in slightly different locations in successive planning cycles. As illustrated in the left panel of each column in Fig. 4, the frontier candidates (grey circle) are scattered throughout the low cost area. While the underlying cost map reflects the irregular shape of the path and instability of the sensor data. However, a key finding is that despite this input noise, the proposed method can select optimal local goal forward to the global goal by evaluating this scattered set of candidates based on proximity to the global goal. For the future work, we plan to address such inconsistency due to input noise by managing frontier candidates into frontier database over a short temporal window. We expect that this would allow the system to stabilize candidate positions, leading to more consistent and predictable local goal selection.

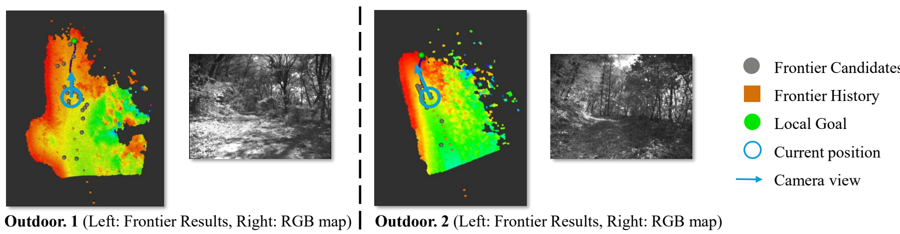


**Fig. 4.** Real-world experiments in rugged terrain (mountainous regions).

## 5 Conclusion

In this paper, we presented a novel state-aware exploration framework designed to enhance the robustness of autonomous robots in complex, GPS-denied environments where conventional planners often fail. The core of our approach is a dynamic behavior-switching mechanism that allows the robot to transition between Search, Recovery, and Exploration modes. This strategy directly addresses the critical challenge of local minima by introducing a dedicated Recovery Mode that leverages a historical database of previously discovered frontiers to escape stuck states and resume its mission. The effectiveness of our framework was validated through rigorous simulations in four distinct and challenging environments. The experimental results unequivocally demonstrate the superiority of our method. While the baseline planner consistently failed in all trials with a 0% success rate, the integration of our module transformed it into a resilient system capable of achieving up to 100% mission success. This highlights the value of our framework not as a mere planner, but as a crucial supervisory layer for resilient navigation.

**Limitations and Future Work.** We acknowledge that our experiments focused on demonstrating the necessity of a recovery mechanism against a planner with none. A valuable next step is to conduct a comparative analysis of our memory-based recovery strategy against other, simpler recovery heuristics, such as simple path backtracking or returning to the most recent frontier. This would further clarify the benefits of our strategic recovery point selection. Future work will also focus on deploying and validating this framework on physical robotic platforms in large-scale, real-world scenarios.

**Acknowledgements**. This project was funded and supported by the grant from Hanwha Aerospace as part of the development of autonomous driving technology for unstructured environment.

## References

1. Ali, M., Jardali, H., Roy, N., Liu, L.: Autonomous navigation, mapping and exploration with gaussian processes. Robotics: Science and Systems (RSS) (2023)
2. Aurenhammer, F.: Voronoi diagrams—a survey of a fundamental geometric data structure. ACM computing surveys (CSUR) **23**(3), 345–405 (1991)
3. Bi, Q., Zhang, X., Wen, J., Pan, Z., Zhang, S., Wang, R., Yuan, J.: Cure: A hierarchical framework for multi-robot autonomous exploration inspired by centroids of unknown regions. IEEE Transactions on Automation Science and Engineering **21**(3), 3773–3786 (2023)
4. Cao, C., Zhu, H., Yang, F., Xia, Y., Choset, H., Oh, J., Zhang, J.: Autonomous exploration development environment and the planning algorithms. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 8921–8928. IEEE (2022)
5. Dang, T., Tranzatto, M., Khattak, S., Mascarich, F., Alexis, K., Hutter, M.: Graph-based subterranean exploration path planning using aerial and legged robots. Journal of Field Robotics **37**(8), 1363–1388 (2020)
6. González-Banos, H.H., Latombe, J.C.: Navigation strategies for exploring indoor environments. The International Journal of Robotics Research **21**(10-11), 829–848 (2002)
7. Keidar, M., Kaminka, G.A.: Efficient frontier detection for robot exploration. The International Journal of Robotics Research **33**(2), 215–236 (2014)
8. Korf, R.E., Zhang, W., Thayer, I., Hohwald, H.: Frontier search. Journal of the ACM (JACM) **52**(5), 715–748 (2005)
9. Leininger, A., Ali, M., Jardali, H., Liu, L.: Gaussian process-based traversability analysis for terrain mapless navigation. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 10925–10931. IEEE (2024)
10. Long, S., Li, Y., Wu, C., Xu, B., Fan, W.: Hphs: hierarchical planning based on hybrid frontier sampling for unknown environments exploration. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 12056–12063. IEEE (2024)
11. Ryu, H.: Graph search-based exploration method using a frontier-graph structure for mobile robots. Sensors **20**(21), 6270 (2020)
12. Williams, G., Aldrich, A., Theodorou, E.: Model predictive path integral control using covariance variable importance sampling. arXiv preprint arXiv:1509.01149 (2015)
13. Yamauchi, B.: 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'. pp. 146–151. IEEE (1997)
14. Zhang, S., Zhang, X., Dong, Q., Wang, Z., Xi, H., Yuan, J.: Fsmp: A frontier-sampling-mixed planner for fast autonomous exploration of complex and large 3-d environments. IEEE Transactions on Instrumentation and Measurement (2025)