2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

FINENAV: A Versatile Framework Enhancing Ground Robot Navigation in Unstructured Environment

Jinghui Wang, Chenyang Wang, Yuxuan Cao, Zelong Sun, Wang Xi, and Jianping He

***Abstract*— Autonomous navigation of ground robots in unstructured 3D environments remains a fundamental challenge, as it requires accommodating dynamic obstacles, non-planar ground, and multi-story structures within a unified framework. In this paper, we propose a versatile navigation framework named FINENAV. It features a novel hierarchical mapping system that couples a high-rate local voxel grid for real-time perception with a scalable global octree for persistent storage. This design balances low-latency performance with large-scale mapping capabilities, enabling reliable navigation in unstructured environments. Moreover, the entire navigation pipeline is refactored into modular and reusable components, while maintaining compatibility with existing 2D navigation ecosystems. We validate FINENAV on a wheeled robot, demonstrating its versatility across diverse scenarios. FINENAV is released as open-source software for the community.**

I. INTRODUCTION

Navigation is a fundamental problem in robotics. Although algorithmic research in this field has achieved remarkable results [1], [2], [3], deploying a fully functional navigation stack remains challenging. A navigation stack typically integrates perception, mapping, and planning. Owing to its multidisciplinary nature and extensive workload, such a system is highly complex to design from scratch.

Several representative frameworks exist for 2D planar environments [4]. These frameworks significantly reduce the development effort for autonomous navigation. However, the planar assumption of the environment restricts their applicability to broader scenarios. Real-world environments are inherently three-dimensional, requiring robots to navigate non-planar ground and multi-story structures. Replicating the success of navigation frameworks in such contexts is crucial for advancing the next stage of mobile robotics navigation.

However, developing next-generation frameworks faces challenges on multiple fronts. First, reasoning in 3D space inherently demands a strict balance between real-time computational efficiency and the heavy memory footprint of large-scale maps. Second, while advanced planning algorithms exist [5], [6], integrating them into a deployable robotic system remains difficult. Third, practical usability requires modular architectures and compatibility with existing ecosystems—features that are rarely prioritized. Thus, the evolution of navigation frameworks requires multi-level co-optimization spanning from algorithms to system design.

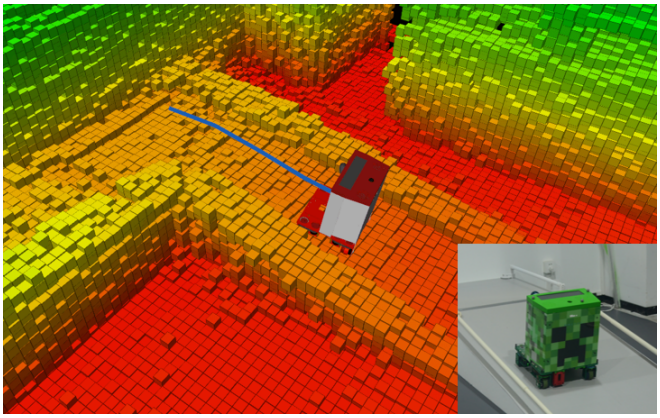


Fig. 1: A wheeled robot navigating on non-planar ground. The local grid map is visualized as elevation-rendered cubes; the blue line indicates the planned local trajectory.

Recent advances have made developing such a framework increasingly attainable. Breakthroughs have emerged in motion planning for ground robots in unstructured 3D environments [7], [8]. Concurrently, research into advanced spatial data structures has demonstrated a critical role in optimizing both computational latency and memory consumption for navigation tasks [9]. Alongside these algorithmic advancements, our long-term practice shows that a dedicated mapping system can dramatically simplify the integration of motion planners into a deployable pipeline. These factors jointly motivate us to develop our versatile framework, FINENAV[1].

FINENAV is a full-stack navigation framework designed specifically for ground robots operating in unstructured 3D environments. It employs a novel hierarchical mapping system inspired by the cache-memory model in computer science. Local perception data is cached in a high-rate voxel grid for rapid access, and is subsequently lazily offloaded into a global octree for memory-efficient storage. Leveraging this hierarchical foundation, FINENAV seamlessly integrates advanced motion planning algorithms, achieving the versatility to navigate scenarios with dynamic obstacles, non-planar ground, and multi-story structures. Furthermore, to maximize practical usability, FINENAV is built upon a highly modular

J. Wang, C. Wang, Y. Cao, W. Xi, and J. He are with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, the Key Laboratory of System Control and Information Processing, Ministry of Education of China, and Shanghai Key Laboratory of Perception and Control in Industrial Network Systems, Shanghai, 200240, China. E-mails: {wangjinghui, minhangwcy, ziyezhishi, bddwyx, jphe}@sjtu.edu.cn

Z. Sun is with the School of Computer Science, China University of Geosciences, Wuhan, China. E-mail: dragon521hope@cug.edu.cn

This work was supported in part by the National Natural Science Foundation of China under Grant 62373247.

[1]Project available at https://github.com/FINS-Fines/FineNav

architecture of reusable components, and maintains seamless compatibility with the Navigation2 (Nav2) ecosystem [4].

The main contributions of this work are:

- We propose a novel hierarchical mapping system with enhanced dynamic performance for unstructured environments and empirically demonstrate its effectiveness.
- Building on this method, we develop the full-stack navigation framework (named FINENAV). It is validated in multiple real-world scenarios and exhibits greater versatility than comparable frameworks.
- We further reorganize FINENAV into modular components to improve usability and flexibility, and release it as open-source to foster community development.

II. RELATED WORK

A. Navigation Frameworks

Navigating ground robots in unstructured 3D environments introduces three primary challenges:

- *Dynamic obstacles:* Moving entities create continuous spatial variations, requiring low-latency perception and rapid map updates to ensure real-time avoidance.
- *Non-planar ground:* Traversing uneven terrain requires systems to evaluate traversability based on physical geometric constraints [10], as depicted in Fig. 1.
- *Multi-story structures:* Environments with vertically overlapping spaces require explicit 3D modeling of inter-floor connectivity to enable continuous global path planning across multiple levels [6].

Several system-level frameworks have been proposed to address these challenges. Wang et al. [11] introduced a method to extract multi-layer 2D maps from 3D representations, enabling reasoning about slopes and staircases. While effective in capturing structural features, this approach requires reasoning over the entire pre-built map, which limits its suitability for real-time adaptation. Jian et al. [12] proposed a plane-fitting method that models the ground to identify traversable regions. However, this procedure extracts only a single surface along the z-axis in a bottom-to-top manner, which restricts its applicability in multi-layer environments such as bridges or multi-story buildings. Atas et al. [13] encountered the same limitation due to its reliance on an elevation map, which cannot represent multiple vertical layers. Cao et al. [14] developed a widely adopted environment for autonomous exploration tasks. Although it demonstrates success in these tasks within multi-story parking garages, its navigation capabilities remain limited to single-story scenarios.

In summary, existing frameworks typically focus on a subset of the above challenges and cannot address them simultaneously. Table I summarizes their capabilities. This gap highlights the need for a versatile framework that can accommodate dynamic obstacles, non-planar ground, and multi-story structures.

TABLE I: Comparison on Capabilities of Navigation Frameworks

	Dynamic Obstacles	Non-planar Ground	Multi-story Structures
Wang et al. [11]	✓	×*	×
Jian et al. [12]	×	✓	×
Atas et al. [13]	✓	✓	×
Cao et al. [14]	✓	✓	×
Proposed	✓	✓	✓

* Relies on pre-built maps rather than online reasoning.

B. Environment Representation

Map representation forms a cornerstone of navigation systems. While some recent approaches explore mapless navigation [15], [16], map construction remains indispensable for global reasoning in navigation tasks.

A straightforward representation for 3D environments is the dense voxel grid, in which space is discretized into fixed-size cubic voxels with occupancy states. This representation supports constant-time access to elements but incurs a substantial memory footprint, particularly in large-scale environments. To address this limitation, Hornung et al. [17] proposed OctoMap, an octree-based representation. By pruning subtrees with homogeneous occupancy, OctoMap achieves significant memory savings compared to dense grids. However, this sparsity introduces a critical computational cost—querying voxel states requires logarithmic time in the depth of the tree, which creates a bottleneck for high-rate perception.

To mitigate the trade-offs of these basic representations, substantial efforts have focused on optimizing monolithic spatial data structures [18], [19], [20]. However, these approaches inherently couple high-rate local perception with global memory management. As the map scales, maintaining memory contiguity within a massive monolithic structure becomes fundamentally difficult. This resulting memory fragmentation degrades CPU cache efficiency, severely bottlenecking practical access speeds [9].

Crucially, robotic navigation naturally exhibits strong spatial locality, as real-time operations primarily query the robot's immediate surroundings. Rather than relying on a monolithic map, a navigation framework can leverage this locality by combining distinct maps through targeted responsibility allocation. This collaborative approach can simultaneously achieve high-speed reactive access and scalable global mapping.

III. SYSTEM ARCHITECTURE

Fig. 2 shows the structure of FINENAV. The framework is organized around a hierarchical mapping system that maintains two complementary representations: a high-rate 3D *local grid* centered on the robot and a persistent *global octree* for long-term storage.

The two maps serve distinct but cooperative roles in navigation. The local grid captures the most recent sensor observations within the robot's reliable sensing range, providing accurate short-term perception. Algorithms such as raycasting and terrain analysis operate directly on the local map, making fast access its primary design goal. In contrast, the global octree stores the representation of the entire environment. It enables long-term reasoning, and

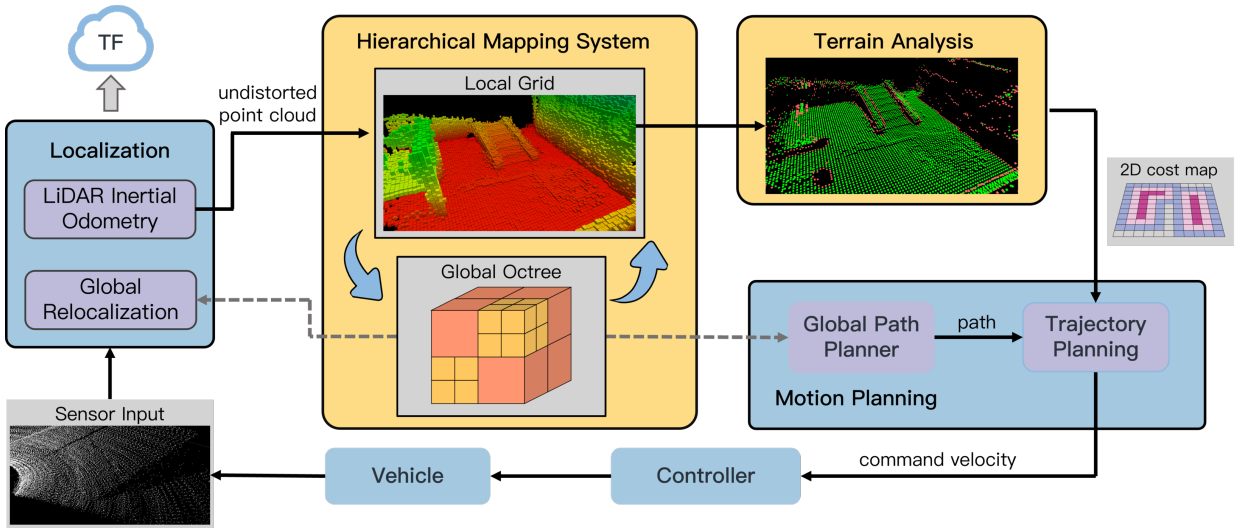


Fig. 2: System overview of the FINENAV framework. LiDAR data is used for localization by fusion with a pre-built map. Undistorted point clouds are integrated into a hierarchical map containing a local grid and a global octree. Terrain analysis produces a 2D cost map for motion planning, after which the controller executes the trajectory. Operations marked with gray dotted lines occur only during initialization.

must be memory-efficient to remain practical in large-scale environments.

These two maps exchange data seamlessly. As the robot moves, regions that leave the local grid are offloaded into the global octree, while newly entered regions are reloaded when required. This continuous exchange mirrors the cache–memory relationship in computer systems. To enable this complementary cooperation, both map structures and their update mechanisms are carefully designed, which will be detailed in Sec. IV.

Peripheral modules are integrated around the hierarchical mapping system to form a complete navigation stack, which will be detailed in Sec. V.

IV. HIERARCHICAL MAPPING SYSTEM

This section details the implementation of FINENAV's hierarchical mapping system. Since the system relies on two complementary map representations with distinct structures and responsibilities: 1) we first describe their map structures, highlighting the differences in their design goals; 2) we then present the dynamic update procedure, which illustrates how the two maps interact through continuous data exchange during navigation; and 3) we conclude with an analysis showing how this hierarchical design absorbs the complexity of unstructured environments, providing a unified foundation for versatile navigation.

A. Map Structures

1) Local Map: The local map must support rapid queries and frequent updates to enable real-time reasoning. FINENAV therefore employs a 3D grid structure, since grid maps provide $O(1)$ access to voxels and their neighbors [21], which is essential for algorithms relying on spatial adjacency.

The local workspace is discretized into a voxel grid of size $\mathbf{S} = (s_x, s_y, s_z)$, where each dimension is chosen as an odd integer to ensure the robot's geometric center coincides with the central voxel (see Fig. 3).

A voxel is addressed by an integer index triplet $\mathbf{i} = (i_x, i_y, i_z)$, obtained by mapping a point in world coordinates, $\mathbf{p} = (p_x, p_y, p_z)$, into grid coordinates:

$$i_d = \left\lfloor \frac{p_d - c_d}{r} + 0.5 \right\rfloor, \quad \forall d \in \{x, y, z\}, \tag{1}$$

where r is the voxel resolution and $\mathbf{c} = (c_x, c_y, c_z)$ denotes the grid center. This operation discretizes a continuous position into the nearest voxel index.

Each voxel stores a floating-point height value. These values are arranged in a one-dimensional array, with the memory address for index $\mathbf{i}$ given by:

$$\text{address}(\mathbf{i}) = i_z + i_y \cdot s_z + i_x \cdot (s_y \cdot s_z). \tag{2}$$

This z–y–x ordering ensures that all voxels along a vertical column at (x, y) are contiguous in memory, which minimizes overhead for terrain analysis that operate on vertical structures.

To maintain the map centered on the moving robot without costly full-grid shifts, FINENAV implements the local map as a *cyclic buffer*. An offset index $\mathbf{O} = (o_x, o_y, o_z)$ marks the displacement between the logical grid origin and the physical memory start. As the robot moves, $\mathbf{O}$ is updated and only the newly entered regions are cleared, eliminating the need for full memory copies.

With the cyclic buffer implementation, Eq. (1) is modified to account for wrapping at grid boundaries. The mapping from a point $\mathbf{p}$ to its voxel index becomes:

$$i_d = \left\lfloor \frac{p_d - c_d}{r} + 0.5 + o_d \right\rfloor \bmod s_d, \quad \forall d \in \{x, y, z\}. \tag{3}$$

This formulation ensures that indices wrap around at the grid boundaries while keeping the robot fixed at the grid center.

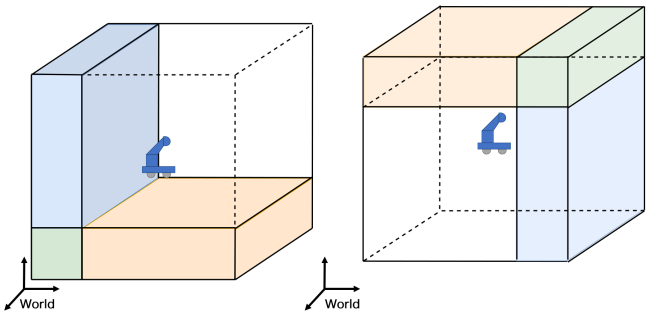


Fig. 3: Illustration of the local grid implemented as a cyclic buffer. The robot remains fixed at the grid center, while the logical grid (dashed box) shifts with robot motion. Colored regions indicate memory blocks that are reused across shifts, demonstrating how cyclic buffering preserves spatial consistency without requiring costly full-grid copies.

2) Global Map: The global map must provide scalable storage of large-scale environments without incurring prohibitive memory costs. To this end, FINENAV employs an octree-based representation, which exploits spatial sparsity and thus greatly reduces memory usage in large-scale environments.

A well-known drawback of octrees is their $O(\log N)$ cell access time [17], which is considerably more expensive than the constant-time lookups offered by contiguous grid structures. Directly coupling an octree with time-critical navigation tasks therefore introduces substantial overhead.

FINENAV's hierarchical design mitigates this limitation. The global octree does not process raw sensor point clouds or participate in local reasoning. Instead, it integrates only the aggregated voxel data exported from the local grid map. This design substantially reduces the update rate and computational overhead placed on the global map. Furthermore, the costly propagation of child-node updates to parent nodes can be deferred to background processing, ensuring that real-time reasoning and planning remain unaffected.

Our implementation builds upon the OctoMap framework [17], but differs in a key aspect: each octant stores a floating-point height value, while still preserving OctoMap's probabilistic occupancy updating mechanism. This modification equips the global map with the vertical accuracy required to represent complex terrain geometry in unstructured 3D environments.

B. Dynamic Update Procedure

The mapping system supports range-sensing modalities such as LiDAR, stereo, and RGB-D cameras. To incorporate dynamic obstacles in real time, FINENAV performs an update cycle in which sensor data is first integrated into the local grid map and subsequently synchronized with the global octree. This design decouples the real-time pipeline with the costly update of the global octree.

The update cycle for each sensor frame consists of three sequential steps, as illustrated in Fig. 4.

a) Local Grid Realignment: Based on the most recent localization estimate, the local grid map is shifted so that its center remains aligned with the robot. During this shift, voxels that move outside the grid are dumped for later insertion into the global octree, while voxel states for newly entered regions are retrieved from the global map.

Complexity. The main overhead arises from synchronizing boundary voxels. The number of affected voxels is proportional to the grid surface area. For a grid of size $\mathbf{S} = (s_x, s_y, s_z)$, we define

$$A_{\text{grid}} = s_x s_y + s_y s_z + s_x s_z. \tag{4}$$

A naive implementation would require $O(d \cdot A_{\text{grid}})$ operations, where d is the octree depth. FINENAV reduces this cost by grouping boundary voxels into bounding boxes and using OctoMap's accelerated leaf access [17], which yields

$$T_{\text{realign}} = O(d + A_{\text{grid}}). \tag{5}$$

b) Sensor Integration: Once the grid is realigned, the incoming point cloud is transformed into the world coordinate frame and inserted into the local grid. For each measurement, a ray is cast from the sensor origin to the measured endpoint, with all traversed voxels marked as free and the terminal voxel updated as occupied. To ensure efficiency, FINENAV employs a 3D variant of the Digital Differential Analyzer (DDA) algorithm [22] for voxel traversal. Following this step, the local grid reflects the latest observations and is immediately available for terrain analysis and motion planning.

Complexity. Let $\mathcal{M}$ denote the input point cloud with $|\mathcal{M}| = m$ points. Each point requires a raycast, and the number of voxels traversed is proportional to the sensing range L divided by voxel resolution r. The total cost is

$$T_{\text{raycast}} = O\left(m \cdot \frac{L}{r}\right). \tag{6}$$

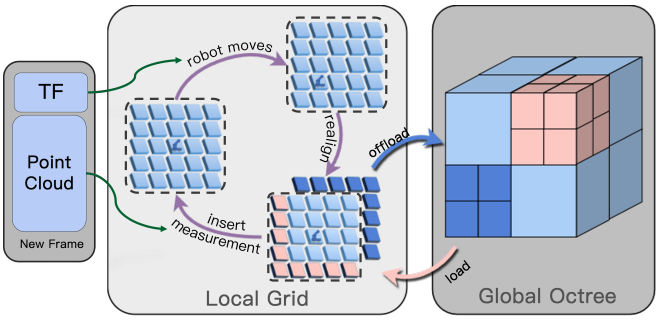


Fig. 4: Map update pipeline in FINENAV. As the robot moves, the local grid is realigned to remain centered on the robot. Newly entered regions are loaded from the global octree, while incoming sensor measurements are inserted into the updated grid. Outdated regions are offloaded to the global octree, where updates can be executed in the background without disrupting real-time performance.

c) Global Octree Update: Finally, the voxels dumped in the first step are inserted into the global octree. Each update applies OctoMap's probabilistic occupancy rule [17], and for occupied voxels, the height value from the local grid is propagated to the corresponding octant. Since this step can be deferred to background execution, it does not impact the latency of the real-time update loop.

C. Analysis

The hierarchical mapping system absorbs much of the complexity inherent in unstructured environments. Its effectiveness can be understood with respect to the three key challenges:

a) Dynamic obstacles: Raycasting and incremental updates are performed directly on the local grid, where fast memory access enables obstacle information to be refreshed in real time. The handling of dynamic obstacles is achieved through the update procedure described in Sec. IV-B.

b) Non-planar ground: By storing explicit height values instead of simple binary occupancy, the local grid provides fine vertical resolution required for accurate terrain analysis. This direct encoding of elevation natively accelerates high-rate geometric calculations, such as slope evaluation and elevation change detection.

c) Multi-story structures: The global octree maintains a persistent, memory-efficient representation of the entire environment. By preserving vertical connectivity, it provides the essential spatial context required to compute continuous paths across multiple levels.

Therefore, the hierarchical mapping system provides a unified foundation that enables FINENAV to operate reliably across the challenges of unstructured environments.

V. PERIPHERAL MODULES

To build a complete navigation framework around the hierarchical mapping system, FINENAV integrates several essential modules. Among these, the terrain analysis module plays a central role in bridging perception and planning and is therefore discussed in detail. Other modules, including localization and global path planner, are introduced more concisely. Finally, we describe the modular design and compatibility features that ensure flexibility and seamless integration with existing navigation ecosystems.

A. Terrain Analysis

Terrain analysis identifies traversable ground by reasoning over the most recent local grid map. Its output is a 2D cost map that can be directly consumed by downstream planning modules.

The default analyzer in FINENAV determines ground traversability according to two simple yet broadly applicable rules:

1) A candidate ground point must have sufficient vertical clearance above it for the robot to pass.
2) The slope around the candidate must lie within the robot's locomotion capability.

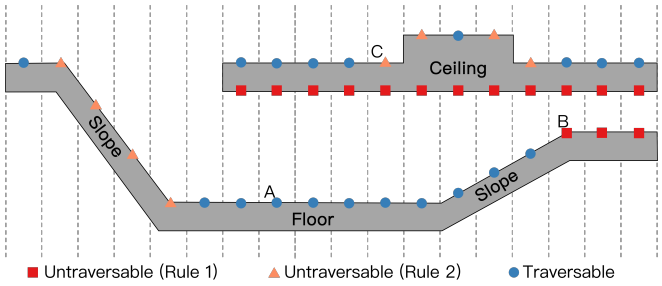


Fig. 5: Illustration of FINENAV's default terrain analysis. At each horizontal grid cell (x, y), candidate ground points are evaluated for traversability according to the two rules in Sec. V-A. Point A is marked traversable. Point B is rejected because the vertical clearance above it is insufficient for the robot to pass. Point C is rejected because the elevation difference with neighboring cells exceeds the robot's maximum admissible slope.

These two rules capture the essential requirements of most ground robots and therefore serve as an out-of-the-box solution.

For a column (x, y) containing a set of sorted height samples

$$\mathcal{Z}(x,y) = \{z_1, z_2, \ldots, z_m\}, \quad z_1 < z_2 < \cdots < z_m, \tag{7}$$

the subset of candidate ground points is defined as

$$\mathcal{Z}_c(x,y) = \left\{ z_i \in \mathcal{Z}(x,y) \;\middle|\; i = m \text{ or } z_{i+1} - z_i \geq H_r \right\}, \tag{8}$$

where H_r is the minimum vertical clearance required for the robot. The case $i = m$ ensures that the topmost surface is considered as a candidate ground point.

Among $\mathcal{Z}_c(x, y)$, the analyzer selects the candidate that is closest to the robot's current body height p_z:

$$z_c(x,y) = \underset{z_i \in \mathcal{Z}_c(x,y)}{\arg\min} \; |z_i - p_z|. \tag{9}$$

Traversability is then determined by evaluating the slope around the candidate ground point. For the four-connected neighborhood $\mathcal{N}(x, y)$, the maximum vertical difference is

$$\Delta(x,y) = \max_{(u,v) \in \mathcal{N}(x,y)} |z_c(x,y) - z_c(u,v)|, \tag{10}$$

and the slope angle is

$$\varphi(x,y) = \arctan\left(\frac{\Delta(x,y)}{r}\right), \tag{11}$$

where r is the horizontal grid resolution. The cell (x, y) is marked as traversable if

$$\varphi(x,y) < \theta_{\max}, \tag{12}$$

with $\theta_{\max}$ denoting the maximum admissible slope.

B. Localization

FINENAV employs an extended version of Fast-LIO2 [23] for LiDAR-inertial odometry. To better integrate it into a navigation framework, we introduce two key extensions. First, velocity estimates from the internal filter are extracted and published as an explicit output, enabling reliable velocity feedback even on systems equipped with only a single LiDAR. Second, pose predictions are propagated using IMU data between LiDAR frames to generate a high-frequency odometry stream, preventing system instability at high robot speeds.

In addition to high-frequency odometry, practical navigation frameworks often require global relocalization against pre-built maps to support lifelong autonomy. FINENAV facilitates this through its global octree, which is continuously updated during operation to maintain an accurate map over long deployments. Volumetric localization methods, like [24], can be deployed directly on the map to perform global relocalization.

C. Global Path Planner

FINENAV integrates an external global planner that leverages the 3D global octree to compute reference paths, a capability critical for navigation in scenarios with multi-story structures. FINENAV adopts the approach in [6], which reduces the computational burden of 3D planning by formulating the problem in a 2.5D configuration space.

We further enhanced the original implementation to better support real robotic systems. First, the planner was re-engineered for CPU-only platforms, accommodating the limited computational resources of many robots. Second, data handling was optimized by minimizing disk I/O and redundant map copying, which significantly improves runtime performance.

D. Usability

To ensure practical usability, FINENAV prioritizes three key design aspects.

First, all components are organized in a modular architecture. Each module can be replaced or reconfigured independently without affecting the rest of the system.

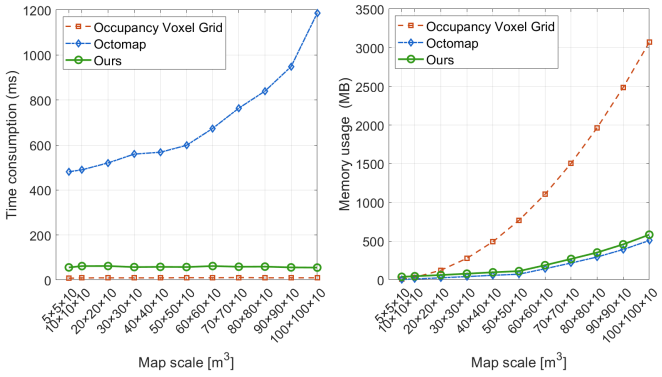
Second, selected modules are implemented through plugin interfaces. In particular, terrain analysis is tightly coupled with locomotion capability, and thus exposed as a plugin. Custom analyzers can be dynamically loaded as long as they implement the required interface, enabling adaptation across platforms ranging from wheeled robots restricted to near-flat ground to legged robots capable of climbing stairs.

Finally, FINENAV maintains compatibility with existing 2D navigation ecosystems. A 2D occupancy-grid interface is exported from the hierarchical mapping system, allowing seamless integration with Nav2 [4] and its well-established controllers.

VI. EXPERIMENTS

To evaluate the performance and versatility of the proposed FINENAV framework, we conducted a series of experiments. First, we quantitatively assessed the efficiency of the hierarchical mapping system against established mapping approaches. Second, we validated FINENAV's effectiveness in a complex, unstructured environment that encompasses dynamic obstacles, non-planar ground, and multi-story structures.

A. Hierarchical Mapping System Performance



(a) (b)

Fig. 6: Performance comparison of FINENAV's hierarchical mapping system with an occupancy voxel grid and OctoMap. (a) Update time per frame under different map scales. (b) Memory consumption under different map scales.

For map evaluation, we focused on two metrics critical for real-time navigation: 1) *update time per frame*, measured as the average time to integrate a LiDAR scan via raycasting, which dictates system responsiveness; and 2) *memory consumption*, which determines scalability in large-scale environments. We benchmarked FINENAV's hierarchical mapping system against two widely used alternatives: an occupancy voxel grid and OctoMap [17]. The voxel grid is known for fast updates due to its contiguous memory layout, whereas OctoMap offers strong memory efficiency through its sparse octree representation.

Experiments were conducted in simulation using Ignition Gazebo and the multi-story parking garage environment from [14]. All maps were configured with an identical voxel resolution of 0.05 m and initialized at different map scales. Each frame contained approximately 30k LiDAR points, incrementally inserted into the map. All experiments were run on a laptop with an Intel i9-13900HX CPU and an NVIDIA GeForce RTX 4060 Laptop GPU.

Results are shown in Fig. 6. The hierarchical mapping system achieves a favorable balance between computational efficiency and memory footprint. Its update time is comparable to that of the occupancy voxel grid, with the slight overhead arising from synchronization between its internal local grid and global octree. At the same time, its memory usage remains close to OctoMap. The additional footprint stems from storing a height value in each node—a necessary tradeoff to avoid z-axis discretization errors that would otherwise compromise terrain analysis.

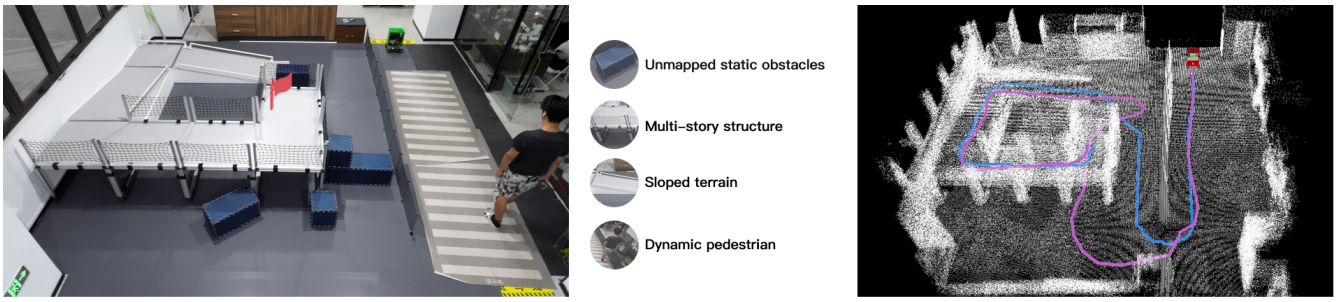
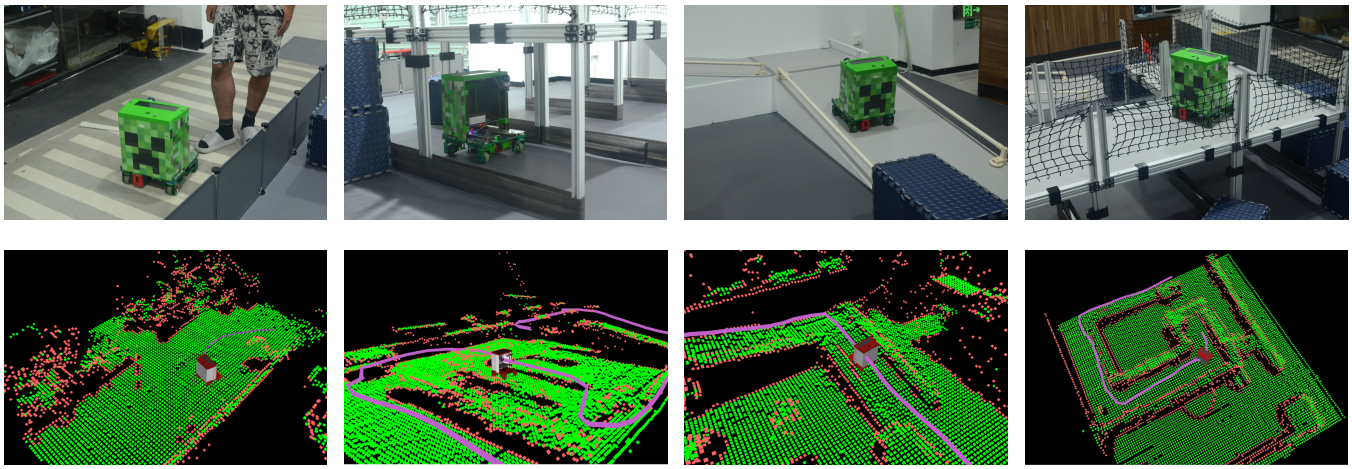


Fig. 7: Experimental setup and results of real-world navigation in an unstructured environment. Left: the constructed testing scenario featuring multi-level terrain and dynamic obstacles, with the robot (green) tasked to reach a goal on the second floor (red flag). Right: the corresponding pre-built map and navigation traces, showing the globally planned path (blue) and the actual trajectory (purple) deviating to avoid unmapped obstacles.



(a) Dynamic Obstacle Avoidance (b) Underpass Traversal (c) Slope Traversal (d) Second-floor Arrival

Fig. 8: Representative experimental frames demonstrating FINENAV's performance across diverse challenges. For each scenario, the top image shows the real-world view and the bottom image shows the corresponding algorithmic visualization, where green indicates traversable areas and red indicates obstacles. The robot's trajectory is shown in purple.

B. Navigating in Real-world Scenarios

FINENAV was deployed on a wheeled robot equipped with a Livox Mid360 LiDAR for perception and a mini-PC with an AMD R7-8845H CPU and 16 GB RAM for onboard computation.

The testing environment (Fig. 7) was deliberately constructed to feature the primary challenges of unstructured environments: dynamic obstacles (pedestrians), non-planar terrain (slopes), and multi-story structures. Only the persistent structural elements were pre-built into the global octree; additional obstacles and moving pedestrians were intentionally left unmapped, requiring online perception and reasoning.

The robot was tasked with autonomously reaching a goal marker on the second floor. Upon receiving the goal, the global planner used the pre-built octree to compute a feasible reference path. Real-time trajectory generation was then carried out by Nav2's MPPI controller [5], guided by traversability estimates from FINENAV's default terrain analyzer.

Across all trials, FINENAV reliably generated accurate traversability estimates in real time, enabling the robot to closely follow the global reference path while dynamically avoiding both static and unexpected moving obstacles. Fig. 8 presents representative snapshots from a successful run. These results demonstrate FINENAV's versatility across the key challenges of unstructured environments.

VII. CONCLUSION

In this paper, we presented FINENAV, a versatile framework designed for reliable ground robot navigation in unstructured 3D environments. Its novel hierarchical mapping system combines a high-rate local voxel grid with a memory-efficient global octree, achieving a balance between real-time responsiveness and scalable storage. Building on this foundation, the framework integrates localization, terrain analysis, and motion planning into a coherent pipeline. Experimental results confirm that FINENAV effectively addresses the core challenges of dynamic obstacles, non-planar terrain, and multi-story structures in real-world scenarios.

In addition, FINENAV emphasizes usability through modular design, plugin-based extensibility, and compatibility with existing 2D navigation ecosystems. By releasing the framework as open-source software, we aim to support further research and encourage its adoption across diverse robotic platforms.

REFERENCES

[1] J. Tordesillas and J. P. How, "FASTER: Fast and safe trajectory planner for navigation in unknown environments," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 922–938, 2021.

[2] M. Zhang, N. Chen, H. Wang, J. Qiu, Z. Han, Q. Ren, C. Xu, F. Gao, and Y. Cao, "Universal trajectory optimization framework for differential drive robot class," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 13 030–13 045, 2025.

[3] R. Han, S. Wang, S. Wang, Z. Zhang, J. Chen, S. Lin, C. Li, C. Xu, Y. C. Eldar, Q. Hao, and J. Pan, "NeuPAN: Direct point robot navigation with end-to-end model-based learning," *IEEE Transactions on Robotics*, vol. 41, pp. 2804–2824, 2025.

[4] S. Macenski, F. Martín, R. White, and J. G. Clavero, "The marathon 2: A navigation system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[5] G. Williams, P. Drews, and e. a. Goldfain, Brian, "Aggressive driving with model predictive path integral control," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016.

[6] B. Yang, J. Cheng, B. Xue, J. Jiao, and M. Liu, "Efficient global navigational planning in 3-D structures based on point cloud tomography," *IEEE/ASME Transactions on Mechatronics*, vol. 30, no. 1, pp. 321–332, 2024.

[7] L. Xu, K. Chai, Z. Han, H. Liu, C. Xu, Y. Cao, and F. Gao, "An efficient trajectory planner for car-like robots on uneven terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023.

[8] J. Wang, L. Xu, H. Fu, Z. Meng, C. Xu, Y. Cao, X. Lyu, and F. Gao, "Towards efficient trajectory generation for ground robots beyond 2D environment," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023.

[9] S. Durvasula, R. Kiguru, S. Mathur, J. Xu, J. Lin, and N. Vijaykumar, "VoxelCache: Accelerating online mapping in robotics and 3D reconstruction tasks," in *International Conference on Parallel Architectures and Compilation Techniques (PACT)*, 2022, pp. 239–251.

[10] Y. Shu, L. Dong, J. Liu, C. Liu, and W. Wei, "Overview of terrain traversability evaluation for autonomous robots," *Journal of Field Robotics*, vol. 42, no. 5, pp. 1724–1765, 2025.

[11] C. Wang, J. Wang, C. Li, D. Ho, J. Cheng, T. Yan, L. Meng, and M. Q.-H. Meng, "Safe and robust mobile robot navigation in uneven indoor environments," *Sensors*, vol. 19, no. 13, p. 2993, 2019.

[12] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.

[13] F. Atas, G. Cielniak, and L. Grimstad, "Elevation state-space: Surfel-based navigation in uneven environments for mobile robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.

[14] C. Cao, H. Zhu, F. Yang, Y. Xia, H. Choset, J. Oh, and J. Zhang, "Autonomous exploration development environment and the planning algorithms," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[15] A. Leininger, M. Ali, H. Jardali, and L. Liu, "Gaussian process-based traversability analysis for terrain mapless navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[16] H. Jardali, M. Ali, and L. Liu, "Autonomous mapless navigation on uneven terrains," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[17] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[18] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D reconstruction at scale using voxel hashing," *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 1–11, 2013.

[19] K. Museth, "VDB: High-resolution sparse volumes with dynamic topology," *ACM Transactions on Graphics*, vol. 32, no. 3, pp. 1–22, 2013.

[20] E. Vespa, N. Nikolov, M. Grimm, L. Nardi, P. H. Kelly, and S. Leutenegger, "Efficient octree-based volumetric SLAM supporting signed-distance and occupancy mapping," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1144–1151, 2018.

[21] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1985.

[22] J. Amanatides, A. Woo *et al.*, "A fast voxel traversal algorithm for ray tracing." in *Annual Conference of the European Association for Computer Graphics (Eurographics)*, 1987.

[23] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[24] F. M. Rico, J. M. G. Hernández, R. Pérez-Rodríguez, J. D. Peña-Narvaez, and A. G. Gómez-Jacinto, "Open source robot localization for nonplanar environments," *Journal of Field Robotics*, vol. 41, no. 6, pp. 1922–1939, 2024.