2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# MPP: Multiscale Path Planning for UGV Navigation in Semi-structured Environments

Rui Cao†, Zhiqiang Yang†, Ran Song*, Ziyu Meng, Ruifeng Wang, Wei Zhang*

*Abstract*— **Autonomous navigation of unmanned ground vehicles (UGVs) in structured road and indoor environments has made significant progress in recent years. However, navigation in outdoor semi-structured environments remains a challenge. This paper presents the multiscale path planning (MPP) method for UGV navigation in semi-structured environments. MPP leverages global, mid-layer and local planners to obtain global path and handle local obstacles of different sizes. First, the global planner provides guidance based on road connection relationships, selecting optimal connections by evaluating the distance between road nodes. Next, the mid-layer planner perceives large-scale obstacles and constructs the costmap, generating a mid-layer path that offers a general direction for the UGV. Finally, a local trajectory planning algorithm, namely terrain-considering timed elastic band (TC-TEB), is used to obtain local trajectory. This algorithm incorporates terrain-velocity constraints into the TEB algorithm to ensure the vehicle's vertical stability. We demonstrate the safety and effectiveness of MPP through experiments in both simulated and real-world environments.**

## I. Introduction

Autonomous navigation of unmanned ground vehicle (UGV) focuses on determining and optimizing the route for the UGV to reach its target safely and effectively. It has made significant progress and has a wide range of applications in structured indoor and road environments [1], [2], [3], [4], [5]. Recently, some researchers further developed techniques for UGV navigation in semi-structured environments. For example, Xie et al. [6] proposed the circular accessible depth (CAD) representation, which is suitable for the perception of semi-structured scenes. Jian et al. [7] introduced the plane-fitting based uneven terrain navigation (PUTN) method, which fits planes to assess scene traversability, enabling commendable navigation in semi-structured environments.

However, semi-structured environments, which typically encompass both structured road scenes and unstructured off-road environments (such as sidewalks, garages, fields and gardens) remain challenging for most existing UGV navigation techniques. This is because unstructured environments usually contain a variety of unknown and unexpected obstacles with irregular shapes and varying sizes. Most existing methods [6], [8], [9], [10] for UGV navigation are tailored primarily for irregular obstacles of conventional size and thus cannot effectively handle large or small scale unstructured obstacles often found in semi-structured environments.

To address the above issue, this paper proposes the multiscale path planning method (MPP) for UGV navigation in semi-structured environments. MPP includes a global planner for directional guidance, a mid-layer planner for large-scale obstacle avoidance and a local planner with a newly designed terrain-considering timed elastic band (TC-TEB) for small-scale obstacle avoidance and velocity optimization.

In the global planner, a target point is initially specified for path planning. It then employs a graph search algorithm, leveraging a road topology map to generate a globally optimal path with minimum cost, thereby providing directional guidance for mid-layer planning. The mid-layer planner integrates CAD [6] into its perception module and constructs a mid-layer costmap for tracking large-scale obstacles. Using the global path from the global planner and this mid-layer costmap, MPP generates a collision-free path for large-scale obstacles. When navigating through uneven terrains, speed control considering terrain traversability is crucial. TEB [11] focuses merely on obstacle information and does not take into account velocity control. To address this issue, the proposed local planner gains insights into the traversability of the environment by analyzing roughness, slope, and edges in the elevation map. Specifically, a TC-TEB algorithm is developed to ensure that the UGV can navigate away from small-scale obstacles and provide speed control in scenarios with passable but high-cost terrains, such as speed bumps or highly uneven road surfaces, which results in smoother and safer navigation.

The contributions of our work are summarized as follows:

- We propose the MPP method integrating global, mid-layer and local planners for UGV navigation in semi-structured environments, and demonstrate its effectiveness in both simulated and real-world environments.
- We develop a new mid-layer planner with a mid-layer costmap including the CAD information and a sub-target selection path planning to handle large-scale obstacles.
- We present a novel terrain-considering TEB (TC-TEB) algorithm which considers the terrain traversability of the environment to optimize velocity control.

† indicates equal contribution.

* Corresponding authors: Ran Song (Email: ransong@sdu.edu.cn) and Wei Zhang (Email: davidzhang@sdu.edu.cn).

Rui Cao, Zhiqiang Yang, Ran Song, Ziyu Meng, Wei Zhang are with the School of Control Science and Engineering, Shandong University, Jinan, China.

Ruifeng Wang is with Shandong Engineering Research Center of Campus Safety Brain (SERCCSB), Jinan, China.

This work was supported in part by the National Key Research and Development Plan of China under Grant 2021ZD0112002, in part by the National Natural Science Foundation of China under Grants 61991411, U22A2057 and 62076148, in part by the Shandong Excellent Young Scientists Fund Program (Overseas) under Grant 2022HWYQ-042, and in part by Project for Self-Developed Innovation Team of Jinan City under Grant 2021GXRC038.

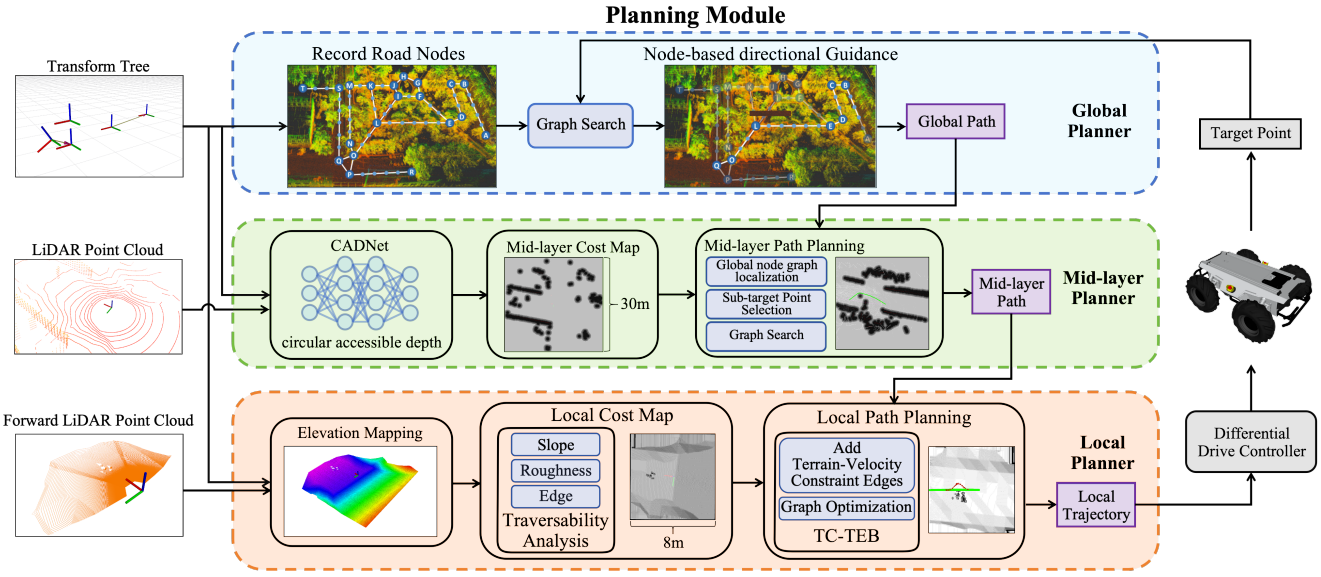


Fig. 1. Overview of the MPP method. MPP takes LiDAR point cloud data as input and obtains the transformation relationships between coordinate systems from the transformation tree. On the right, from top to bottom: The global planner determines the optimal connections between road nodes, where the blue circles represent primary nodes and the blue dots represent secondary nodes. The mid-layer planner generates a mid-layer path based on the global road connections and the cost map constructed from CADNet [6] predictions and transformations. The local planner uses a traversability analysis cost map generated from elevation features and applies the TC-TEB algorithm to derive the optimal trajectory and generate control inputs for the UGV.

## II. Related Work

MPP is a LiDAR based navigation method and highly related to the works on local path planning. Thus this section briefly reviews the literatures in these two areas.

### A. LiDAR-Based Methods for UGV Navigation

According to the dimensions of the perception and planning modules representation, LiDAR-based navigation methods can be categorized into the following three groups.

*1) 2D Representation:* The occupancy grid map (OGM) conveys the traversability of the environment through occupancy probabilities in a 2D map [1], [2], [12], [13]. Marder-Eppstein et al. [1] proposed a method adept at handling laser scan or point cloud to structure global and local costmaps. Lu et al. [2] introduced layered costmaps, which separates costmap processing into layers for efficient tracking different types of obstacles. Ondrúška et al. [12] developed an RNN-based network using 2D laser scan to predict the OGM. Hu et al. [13] employed raycasting to segment nonground LiDAR points and trained a network to predict the occupancy of future states. Bird's Eye View (BEV) semantic maps [14], [15], [16] with rich traversability information have attracted a surge of research interest due to their exceptional performance. Han et al. [14] proposed a semantic segmentation-assisted scene completion network to inpaint sparse semantic LiDAR points into a semantic map. Shaban et al. [15] formulated traversability estimation as a semantic terrain classification problem. Peng et al. [16] proposed a multi-attention mechanism to boost the performance of BEV semantic segmentation. Xie et al. [6] proposed CADNet to generate the CAD representation for UGV navigation that can handle negative obstacles.

*2) 2.5D Representation:* Xie et al. [17] proposed a 2.5D navigation graph that represents both vertical and horizontal obstacles to generate the walkable domain. Gim et al. [18] introduced an autonomous navigation system with a 2.5D map integrating a 2D grid map and 3D geometry information of dynamic objects in a dynamic environment. There are some methods create a 2.5D terrain elevation map [8], [19], [9], [20] for traversability analysis. Fankhauser et al. [8] formulated an elevation mapping method from a robot-centric perspective, which can specifically incorporate the drift in pose estimation. Miki et al. [9] accelerated the process of elevation mapping using GPU. Xie et al. [19] proposed an approach to elevation mapping and traversability analysis using sparse data from a single LiDAR sensor. Semi et al. [20] developed an elevation mapping method that fuses multi-modal information into a configurable environmental representation. However, methods relying on elevation maps are typically used as local perception module due to their restricted perception range.

*3) 3D Representation:* Putz et al. [21] proposed a flexible and map-independent navigation framework for 3D environments by leveraging mesh navigation [22]. Jian et al. [7] utilized a 3D grid map for navigation in unstructured 3D environments. However, these approaches require an initial complete modeling of the outdoor environment and perform poorly with large-scale obstacles. Kato et al [4] proposed Autoware, which can circumvent this problem without full environmental modeling but has limitations in handling unstructured obstacles. Several current approaches [23], [24] represent the environment through 3D occupancy maps derived from LiDAR point clouds for traversability assessment. Nonetheless, these strategies lead to significant memory usage and computational costs, especially in large

outdoor environments, which impair the operational speed and agility of UGV.

### B. Local Path Planning for UGV Navigation

Local path planning methods optimize the planned path conditioned on real-time environmental changes and vehicle's kinematic constraints. Dynamic window approaches [25], [26] introduce selecting velocities within a dynamic window that confines the search space to a set of achievable velocities within a short timeframe, which considers both safety and feasibility. Darweesh et al. [27] introduced OpenPlanner, a sampling-based algorithm that uses a given global path to generate an obstacle-free local trajectory by sampling a set of roll-outs. Quinlan et al. [28] proposed to deform a collision-free path away from obstacles by stretching or elongating the designed elastic bands. This is followed by Rosmann et al. [11], which developed timed elastic bands (TEB) to incorporate time and robot's kinodynamic constraints. These methods disregard the traversability of the terrain around the robot, causing poorer stability on uneven terrain. To address this problem, we analyse the traversability of ground as a new constraint into the path optimization process of TEB algorithm. This approach leads to a smoother and more comfortable local path planning.

## III. Method

In this section, we introduce our framework designed to leverage strength of multiscale information. Fig. 1 displays the structure of our multiscale path planning (MPP) method. The core of our framework is the planning module, which consists of a three-layer planner. Firstly, the global planner provides node-based connectivity information and initial directional guidance, given the road topology. Subsequently, the mid-layer planner generates a safe path that incorporates obstacle data, using the circular accessible depth information provided by the perception module. Finally, our proposed TC-TEB algorithm in the local planner generates a reliable local trajectory that minimizes the vertical oscillation of UGV by integrating results from feature-based local traversability analysis.

### A. Global Planner

*1) Motivation:* We can analogize the outdoor navigation for UGV to the process of people using smartphone navigation apps, such as Google Maps, for road-level planning. Such a top-down solution approach reduces the complexity of navigation issues and significantly enhances the efficiency of the entire planning process. The top-level problem is how to provide UGV with a general direction. Therefore, we opt to first construct a global road topology map to provide initial road connectivity relationships.

*2) Road Topology Map:* As shown in the global planner section of Fig. 1, the topology map of road information for the UGV is an undirected graph. The primary nodes in the graph are located at corners or intersections, and the cost of the edges between nodes is calculated using the Euclidean distance between them. Secondary nodes are inserted at

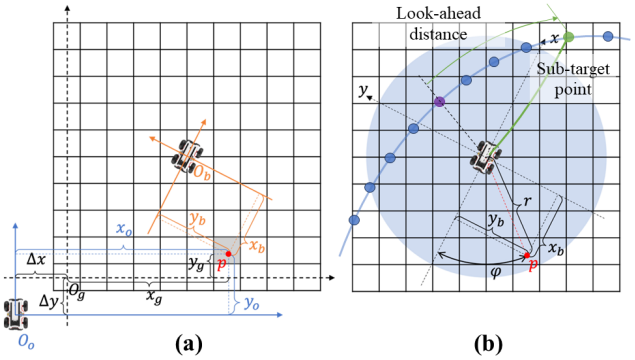


Fig. 2. (a) depicts the transformation of the CADNet prediction results into the cost map. (b) shows the selection of the mid-layer sub-target points. In (b), the blue area represents the circular area of accessible depth perception, the red point indicates the accessible depth prediction result for one of the predicted directions, the green dot signifies the mid-layer sub-target point, and the purple dot maps the position of the UGV on the global topology map.

certain intervals between the primary nodes and are used to position the UGV within the topology map. The primary nodes are designated by users and the secondary nodes are generated using the Map Toolbox [29] in Unity. After specifying a target point, we use graph search algorithms to find the sequence of node connections required for the navigation task, based on the secondary node where the UGV is mapped in the topology graph and the position of the target node.

The global planner provides directional guidance through topological nodes, which assists the mid-layer planner in generating paths.

### B. Mid-layer Planner

*1) Motivation:* Large-scale obstacle avoidance requires global environmental information [30]. In outdoor navigation methods [6], [27], global path relies on pre-recorded waypoints is not optimized. They depend on the local planner for perception and obstacle avoidance, resulting in navigation failures due to a lack of global perception. To solve this problem, we propose a mid-layer planner to perceive and respond to large-scale obstacles. In particular, the mid-layer planner integrating CADNet [6] into its perception module for circular accessible depth prediction, constructs a mid-layer costmap for tracking large-scale obstacles. Given the global path from global planner and this mid-layer costmap, the large-scale obstacle collision-free path can be generated.

*2) Mid-layer Costmap:* To gather information on obstacles within a large-scale range in semi-structured environments, we use CAD [6] as the perception method for mid-layer planning. As shown in the mid-layer planner module of Fig. 1, CAD can predict the border of traversable area using LiDAR point clouds. This results in a probabilistic distribution map of accessible depth based on polar coordinate grids. To enable path planning methods, which typically require graph searches on cost maps based on Cartesian coordinates, it is necessary to transform the perceived accessible depth into a two-dimensional grid costmap. Algorithm 1 outlines this process.

In the algorithm, the input $S$ represents a sequence of LiDAR sensor point clouds, while the output $M$ is a two-dimensional cost map used for path planning. The variable $frame$ denotes a single frame of point cloud data. As shown in Fig. 2 (b), the set $\mathbb{D}$ represents the blue area and $d$ represents the red point. Some functions presented in Algorithm 1 are described as follows:

- $\mathbf{PolarToCartesian}(d)$: Given a point $d = (r, \varphi)$ in polar coordinates, the point $p_b = (x_b, y_b)$ where $x_b = r\cos\varphi, y_b = r\sin\varphi$ in Cartesian coordinates is returned.
- $\mathbf{BaseToOdomTransform}(\mathbf{p_b})$: Given a point $p_b = (x_b, y_b)$ in the base coordinate system of robot, it returns the point $p_o = (x_o, y_o)$ in the odometry coordinate system. The calculation formula is $(x_o, y_o, z_o, 1)^T = T \cdot (x_b, y_b, z_b, 1)^T$, where $T$ represents the transformation matrix from the odometry coordinate system to the base coordinate system.
- $\mathbf{WorldToGrid}(p)$: As shown in Fig. 2 (a), a virtual grid coordinate system is created, with the origin $O_g$ at the centre of the first cell in the bottom left-hand corner, and its orientation is the same as that of the odometry coordinate system. Therefore, the transformation relationship between it and the odometry coordinate system can simply be described by the offsets along the x and y axes, $\Delta x$ and $\Delta y$, respectively. Thus, the coordinates of point p in the grid coordinate system can be obtained as $(x_g, y_g)$, where $x_g = x_o - \Delta x, y_g = y_o - \Delta y$.
  Then, based on the grid resolution $s(m/grid)$, the indices of the cell $(i, j)$ in the cost map corresponding to this position can be obtained, where $i = \lfloor x_g/s \rfloor$ and $j = \lfloor y_g/s \rfloor$.
- $\mathbf{SetCost}(M, i, j)$: Using the given indices $i$ and $j$, mark the cost value of the corresponding cell in the cost map $M$ as occupied.

As shown in Algorithm 1, the process to construct the mid-layer cost map is as follows: Firstly, each $frame$ is input into the CADNet to get the maximum accessible depth set $\mathbb{D}$ in the circular region surrounding the UGV. For $d = (r, \varphi) \in \mathbb{D}$ in every direction, calculate its coordinates $p_b = (x_b, y_b)$ in the Cartesian coordinate system. Apply the transformation matrix $T$ to convert $p_b$ into the odometry coordinate system odom, resulting in coordinates $p_o = (x_o, y_o)$. Finally, based on the grid resolution s and the offset of the cost grid in the odometry coordinate system, $\Delta x$, $\Delta y$, the corresponding grid coordinates values $(i, j)$ are computed, and the cost map grid at $(i, j)$ is marked as occupied.

*3) Path Planning:*

*a) Sub-target Point Selection:* Similar to indoor mobile robots, mid-layer planning involves a graph search task based on a grid cost map. However, there is a key difference: in indoor navigation, target points are typically included within the global cost map but outdoor navigation scenarios often span distances of several hundred meters. Given the perception method CAD has a limited range of about a dozen meters, this results in target points usually not being within the coverage of the mid-layer cost map in outdoor scenarios. Therefore, it is necessary to dynamically select sub-target point before initiating mid-layer path planning. To get a feasible trajectory from the present location of the UGV, it is imperative that the locales designated for sub-target points are devoid of impediments. Consequently, we strategically situate these sub-target points within the range of the cost map, yet deliberately beyond the periphery of the perception range, to guarantee unimpeded accessibility by the graph search algorithm at all times.

**Algorithm 1** PointCloudsToCostMap

```
Input: PointsSeq S
Output: Costmap M
for frame in S do
    𝔻 ← CADNet(frame)
    for d in 𝔻 do
        p_b ← PolarToCartesian(d)
        p_o ← BaseToOdomTransform(p_b)
        i, j ← WorldToGrid(p_o)
        SetCost(M, i, j)
    end for
end for
return M
```

*b) Planning Steps:* As shown in Fig. 2 (b), the mid-layer path planning consists of three steps as follows: Firstly, the mapping position of the UGV on the global topological connectivity node graph is determined. The mapping position refers to the secondary node obtained from the global planner that is closest to the UGV's current position. Subsequent to the ascertainment of the mapping position, the mid-layer planning earmarks a corresponding secondary node situated on the global road as the sub-target point. This selection is predicated upon a pre-established look-ahead distance. Finally, the cost map is used to employ a graph search method to find a path to the mid-layer target. This path is then passed to the downstream local planner to compute the control instructions for the UGV.

### C. Local Planner

*1) Motivation:* While CADNet can detect unstructured obstacles like road shoulders, cones and lampposts, it assumes that smaller obstacles on the longitudinal scale are accessible. However, small-scale obstacles such as bricks, small rocks, or small potholes are often preferable to avoid for safety[6]. Additionally, when UGV navigate through uneven terrains, terrain traversability considering speed control is crucial. To address this issue, we propose a more refined local planner utilizing elevation map as perception module. By analyzing roughness, slope, and edges in elevation map, we can gain insights into the traversability of the environment. Relying on the traversability map, our TC-TEB algorithm can navigate away from small-scale obstacles and provide speed control in scenarios with passable but high-cost terrains, such as speed bumps or highly uneven road surfaces.

*2) Local Traversability Analysis:* Terrain traversability analysis is employed using geometric features as the perception method of mid-layer planner. Slope, roughness,

and edges are selected as terrain attributes to calculate the traversability of grid $g$. The elevation values of the grids within a certain neighborhood centered on grid $g_i$ are recorded as $\mathbb{H} = \{h_1, h_2, ...h_j\}_{j=1:N}$. The corresponding position set is denoted as $\mathbb{P} = \{(x_j, y_j)_{j=1:N}\}$, where N represents the count of elevation cells in the vicinity. Each grid cell $g_j$ is represented by the position vector $\mathbf{x}_j = [x_j, y_j, h_j]^T$.

*a) Slope:* We use the position vector $\mathbf{x}_j$ to calculate the slope feature value $S_i$: First, calculate the normal vector $\mathbf{n}$ of the plane using SVD. Then, calculate the slope feature: $S_i = \arccos(\mathbf{n_z})$.

*b) Roughness:* We use $\mathbb{H}$ and the average value of these elevation values $\bar{h}$ to calculate the roughness value $R_i$:

$$R_i = \sqrt{\frac{1}{N-1}\sum_{j=1}^{N}(h_j - \bar{h})^2}. \tag{1}$$

*c) Edges:* We use the slope feature $S_j$ and the average value of these slope features within the neighborhood $\bar{S}$ to calculate the edge feature $E_i$:

$$E_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(S_j - \bar{S})^2}. \tag{2}$$

*d) Traversability:* $E_i$ mainly reflects the situation where the terrain is at an edge position, $S_i$ focuses on the changing trend of the ground, and $R_i$ shows the difference between the cell and the surrounding terrain. Then we fuse the three features to calculate the traversability value $U_i$ as follows:

$$U_i = 1 - \omega_1 \frac{s}{s_{\text{crit}}} - \omega_2 \frac{r}{r_{\text{crit}}} - \omega_3 \frac{e}{e_{\text{crit}}}. \tag{3}$$

The lower the value of $U$, the higher the potential undulation of the UGV's body. Here, $s$, $r$, and $e$ represent slope, roughness, and edge values, respectively; $\omega_1$, $\omega_2$, and $\omega_3$ are three weights with a sum of 1; values $s_{\text{crit}}$, $r_{\text{crit}}$, and $e_{\text{crit}}$ are the set maximum allowable values for the three geometric factors of slope, roughness, and edge values, respectively. If one of the terrain features $s$, $r$, or $e$ exceeds its critical value, then $U_i$ is set to 0. The traversability value $U_i = 1$ indicates free passage, while $U_i = 0$ denotes impassable terrain.

*3) Terrain-Considering TEB Algorithm:* The TC-TEB algorithm was proposed to address the local trajectory planning problem in semi-structured scenes. Its aim is to enhance the body stability of UGV in uneven terrain. The algorithm's framework is based on the TEB [11] algorithm, which constructs various problems in path planning as constraint edges within a hypergraph structure. The TEB algorithm obtains the optimal trajectory through graph optimization.

Our algorithm retains the original constraint items of the TEB algorithm as shown in Fig. 3 (a). Based on the terrain traversability value mentioned previously, we have added a terrain-velocity constraint cost function $f_{v\_terr}$ into the TEB algorithm, as illustrated in Fig. 3 (b). Terr is depicted for the sake of visualization only and it is not subject to optimization. It connects adjacent pose nodes $p_i$ and $p_{i+1}$

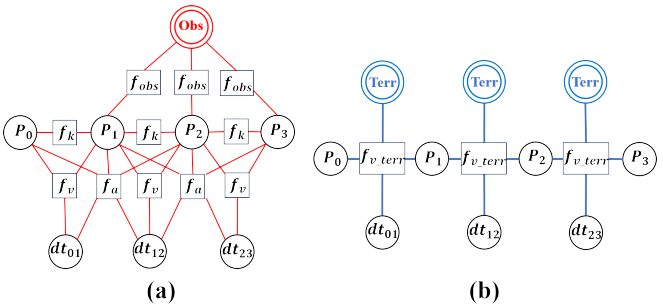


Fig. 3. Comparison of the hypergraphs of the objective functions used in the TEB and the TC-TEB algorithms. (a) displays the complete constraint edges used in the TEB algorithm, and (b) shows only the newly added constraint edges in the TC-TEB algorithm. The blue double circles in the figure represent terrain factors related to the current pose, denoted by Terr.

along with a node representing the time difference $dt_{i,i+1}$ between them, and the terrain related to the pose node $p_i$. The computation method of the $f_{v\_terr}$ cost function is as follows:

$$f_{v\_terr}(v, v_{\text{max\_}}) = \begin{cases} -v - v_{\text{max\_}}, & \text{if } v < -v_{\text{max\_}} \\ 0, & \text{otherwise} \\ v - v_{\text{max\_}}, & \text{if } v > v_{\text{max\_}}. \end{cases} \tag{4}$$

The cost function $f_{v\_terr}$ is composed of two parts: one for linear velocity $v$ and its proportionally scaled maximum $v_{\text{max\_}}$, and another for angular velocity $\omega$ and its scaled maximum $\omega_{\text{max\_}}$. Both parts follow a similar calculation approach but are distinct in their maximum boundary values, $v_{\text{max}}$ and $\omega_{\text{max}}$. The specific formulas for these calculations are detailed below:

$$v_{max\_} = ratio * v_{max} \tag{5}$$

where $ratio$ is the reduction ratio for the maximum linear velocity $v_{\text{max}}$ and the maximum angular velocity $\omega_{\text{max}}$, calculated based on the terrain cost value $\text{cost}_p$ near the pose $p_i$. The calculation of $ratio$ is as follows:

$$ratio = \begin{cases} r_h, & \text{if } cost_p < cost_l \\ (1 - \frac{cost_p - cost_l}{cost_h - cost_l})(r_h - r_l) + r_l, & \text{otherwise} \\ r_l, & \text{if } cost_p > cost_h. \end{cases} \tag{6}$$

In Eq. (6), $cost_{\text{h}}$, $cost_{\text{l}}$, $r_h$, and $r_l$ are parameters we have set, where $cost_{\text{h}}$ and $cost_{\text{l}}$ represent the upper and lower boundaries of the terrain cost value $cost_p$ in the terrain-velocity constraint calculation, $r_h$ and $r_l$ are the upper and lower limits of $ratio$.

The calculation of $cost_p$ is as follows: Given the UGV's pose $p_i$, wheelbase $L$, track width $S$, and orientation $\theta$, the poses of the four wheels of the UGV at pose $p_i$ can be calculated. Subsequently, the grid coordinates in the cost map at these wheel poses can be determined, obtaining their respective cost values, denoted as $\text{cost}_{fl}$, $\text{cost}_{fr}$, $\text{cost}_{rl}$, and $\text{cost}_{rr}$. We use the following formula to calculate the terrain cost value $\text{cost}_{p_i}$ at pose $p_i$:

$$\text{cost}_{p_i} = (\text{cost}_{fl} + \text{cost}_{fr} + \text{cost}_{rl} + \text{cost}_{rr})/4. \tag{7}$$

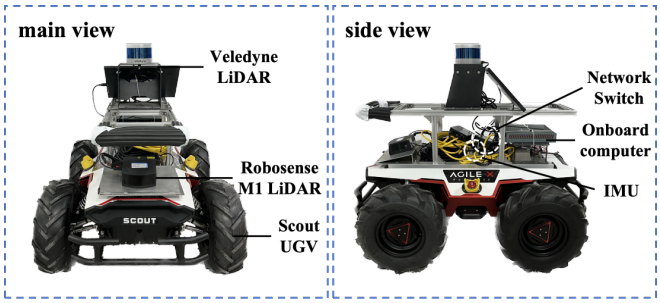


Fig. 4. Our experimental platform, Scout 2.0 is a four-wheeled differential mobile robot equipped with WT-GAHRS1 IMU, RS-LIDAR-M1, and Velodyne VLP-16 LiDAR. The onboard computer used for executing computational processes features an i5-9400H CPU and a GTX-1650 GPU. The sensor configuration in both the simulated and real environments is consistent. The VLP-16 LiDAR is mounted at the center of the UGV, 0.8 meters from the ground, while the RS-LiDAR-M1 is positioned at the front of the UGV, angled downward by 15 degrees.

## IV. Experiments

In this section, to demonstrate the feasibility of our MPP method in practical applications, we conducted comprehensive navigation experiments in both simulated and real-world scenarios, as shown in the supplemental video. Furthermore, through ablation experiments, we have shown the necessity of incorporating a mid-layer planner and the effectiveness of the local planner in optimizing speed.

### A. Comprehensive Navigation Experiment

We deployed the MPP method on the experimental platform as shown in Fig. 4 and conducted comprehensive navigation experiments across multiple scenarios.

*1) Setups:*

*a) Scenarios:* We set up four scenarios, including driveway with speed bumps, sidewalk, underground garage with ramp and uneven dirt road. Among them, driveway, sidewalk, and garage are real-world scenarios, whereas dirt-road is a simulated scenario. The sidewalk contains negative obstacles such as curbs, along with small-scale obstacles like bricks and small wooden blocks, while the garage ramp and simulated dirt road have undulating surfaces. Each method was experimented 15 times within each scenario.

*b) Metrics:* We evaluate the navigation performance using four metrics: collision, success rate, average speed and vertical oscillation. Collisions are categorized into three categories based on the scale of the obstacles: large, small, and other. Success rate is defined as the ratio of the number of times the UGV reaches its navigation target without any collisions. Different obstacles arising in various sections of each test scenario introduce uncertainty in the success rate even within the same scenario. Average speed is calculated as the mean velocity of the UGV across all successful navigation tests. Vertical oscillation is measured by the standard deviation of the z-axis acceleration recorded by the IMU sensor throughout all successful navigation tests.

*c) Baselines:* Two effective autonomous navigation methods, Autoware [4] and CAD [6], were selected as baselines for comparison. In the navigation experiments of CAD, the TEB [11] algorithm was employed as local planner. Autoware is a navigation framework that relies on a pre-constructed high-definition map for autonomous driving positioning and global planning, with its perception module based on PointPillars.

TABLE I

Quantitative Comparison of Different Navigation Methods through Real-World and Simulated Experiments.

| Scenarios | Methods | Obstacle Collisions | | | Success Rate | Avg. Speed (m/s) | Vertical Oscillation |
|---|---|---|---|---|---|---|---|
| | | Large | Small | Other | | | |
| Driveway | Autoware [4] | 1 | 2 | 4 | 53.3% | 0.95 | 0.98 |
| | CAD [6] | 3 | 2 | 0 | 66.7% | 0.98 | 1.03 |
| | **MPP** (Ours) | 0 | 0 | 0 | 100% | 0.91 | 0.73 |
| Sidewalk | Autoware [4] | - | 6 | 9 | 0% | - | - |
| | CAD [6] | - | 5 | 0 | 66.7% | 0.93 | 0.24 |
| | **MPP** (Ours) | - | 0 | 0 | 100% | 0.88 | 0.18 |
| Garage | Autoware [4] | 2 | - | 4 | 60% | 0.91 | 2.38 |
| | CAD [6] | 4 | - | 0 | 73.3% | 0.95 | 2.43 |
| | **MPP** (Ours) | 0 | - | 0 | 100% | 0.83 | 1.78 |
| Dirt-Road | Autoware [4] | 4 | 8 | 3 | 0% | - | - |
| | CAD [6] | 5 | 8 | 0 | 13.3% | 0.89 | 2.89 |
| | **MPP** (Ours) | 0 | 0 | 0 | 100% | 0.77 | 1.41 |

*2) Results and Analysis:* Table I presents the quantitative results of the navigation experiment. In Autoware, the positions of curbs in the high-definition map are predefined, and its perception module cannot detect negative obstacles such as curbs. Therefore, any deviation in positioning could cause Autoware to fall off the curbs. Furthermore, the 3D detection based Autoware is unable to detect small-scale obstacles such as bricks and wooden blocks, as well as terrain undulations. This leads to frequent collisions with small-scale obstacles, some of which lead to navigation failures, and maintaining a relatively high speed in undulating terrain, which increases the value of vertical oscillation.

CAD uses the prediction results from CADNet to avoid collisions with other obstacles. However, CAD experiences can get trapped in corners of large-scale obstacles due to the limitations of the traditional global-local path planning navigation framework. Similar to Autoware, it exhibits an inability to detect smaller obstacles, including bricks and small wooden blocks, as well as a failure to detect terrain undulations. Consequently, its behavior in navigating scenarios characterized by small obstacles and terrain undulations mirrors that observed with Autoware.

In contrast, our multiscale navigation framework, MPP, integrates a mid-layer planner to avoid large-scale and negative obstacles in advance, preventing the system from getting stuck. The local planner, based on terrain traversability analysis, enhances the reliability of the mid-layer path, navigating around small-scale obstacles. Furthermore, the TC-TEB algorithm enables the UGV to reduce speed appropriately while navigating undulating slopes, dirt roads, and speed bumps, thereby enhancing vehicle stability during travel. The method was tested and did not collide with any obstacles, achieving the highest navigation success rate and the minimum degree of oscillation.

### B. Mid-layer Path Planning Experiment

As shown in the Fig. 5, we tested the necessity of adding a mid-layer planner in an outdoor scenario. In the experiments

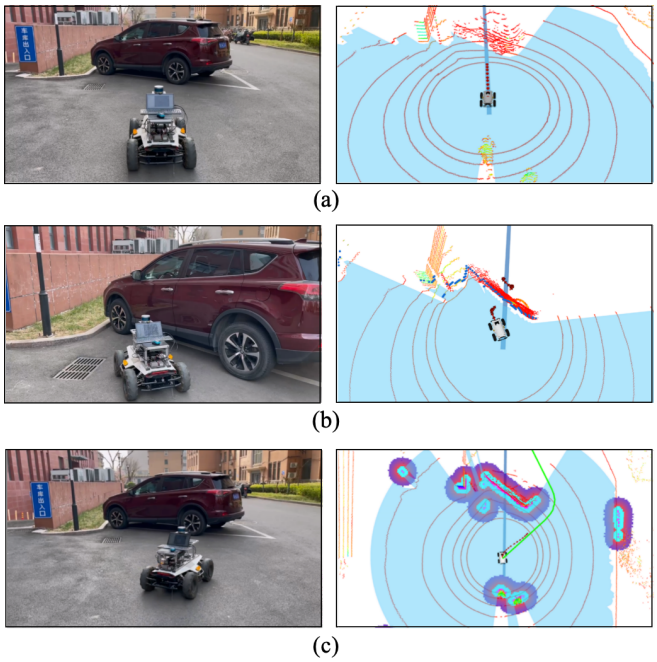


Fig. 5. Mid-layer planner experiment. The point clouds are represented by colourful dots, while the light blue areas indicate the border of traversable area predicted by CADNet [6]. The global path is represented by a blue line, while the local trajectory planned by the TEB [11] method is illustrated by a sequence of red arrows. The mid-layer path is shown with a green curve.

shown in Fig. 5 (a), the SUV blocks the way. As shown in Fig. 5 (b), the local trajectory planned by the TEB algorithm is interrupted by the large-scale obstacle. Fig. 5 (c) shows experiments using our proposed MPP framework, which includes a mid-layer planner. The planned path of the mid-layer enables the UGV to navigate around the large-scale obstacle in advance and successfully complete its navigation mission, guiding the UGV to successfully navigate around the obstacle.

The experiments above demonstrate that incorporating mid-layer planning significantly enhances the traditional global-local path planning framework for handling large-scale obstacles.

### C. Local Path Planning Experiment

We also tested the effectiveness of the local planner in real-world scenarios. Fig. 6 shows the perception results generated at the garage ramp and the sidewalk. At the garage ramp, CAD [6] is able to perceive slender bollards and surrounding walls as obstacles, but it fails to detect changes in slope gradient and terrain undulations. However, the local passability map reveals a significant difference in traversability values between undulating slopes and flat surfaces. The CAD and local traversability map both successfully detect curb edges and street lamps at the sidewalk, preventing the UGV from falling off. However, the CAD's perception results fail to identify small-scale obstacles on the sidewalk, such as bricks and small wooden blocks, which are clearly marked in the local traversability map. Local traversability analysis can compensate for some of the perception results missed by the mid-layer, making the perception more reliable.

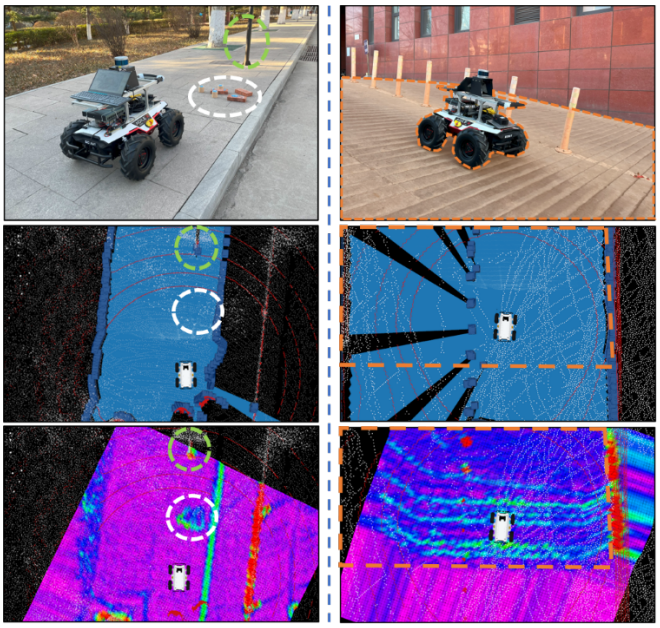

Fig. 6. Mid-layer perception and local perception effects in two unstructured scenarios, with the left side depicting a sidewalk and the right side showing an unevenly undulating garage ramp. The dashed lines in the picture outline the utility pole, bricks, and ramp respectively.

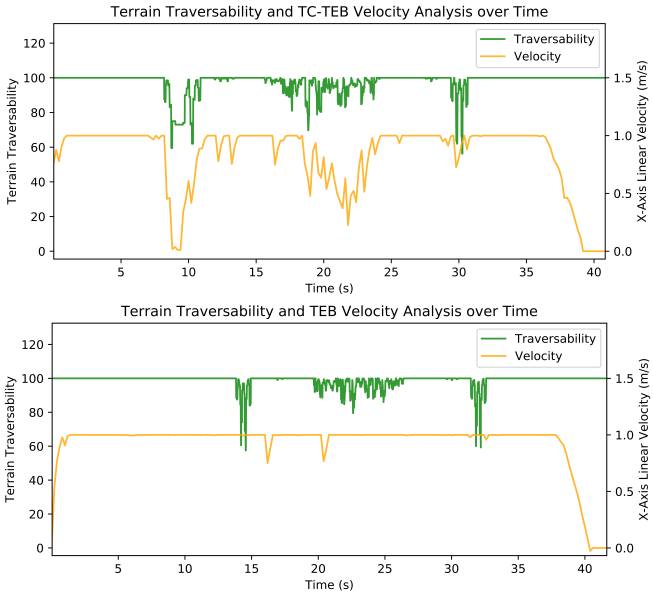


Fig. 7. Comparison of velocity and traversability curves between TC-TEB and TEB Algorithms when traversing uneven terrain. Fluctuations in the traversability curve indicate the presence of terrain undulations.

Fig. 7 shows the navigation performance of an UGV traversing an irregular terrain, facilitated by the TC-TEB local path planning algorithm. It is discernible that the algorithm modulates the vehicle's linear velocity in response to varying levels of traversability. We can observe a reduction in speed as the traversability diminishes.

## V. Conclusion

In this paper, we introduce a novel multiscale path planning framework (MPP) for UGV navigation in semi-

structured environments. By incorporating a mid-layer planner into the conventional structure of global-local indoor navigation framework, we effectively solved the challenges posed by large-scale obstacles in semi-structured settings. We enhance the ability of navigation framework to perceive small-scale obstacles by utilizing the result of local traversability analysis generated from a combination of geometric features of elevation maps. Additionally, by integrating this with the TEB algorithm, we proposed a new local path planning algorithm, the TC-TEB algorithm, which links terrain undulation with velocity to improve the stability and safety of the UGV on uneven terrain. Experiments conducted in various semi-structured scenarios validate the advantages of our navigation framework.

## REFERENCES


[1] Open Source Robotics Foundation, "move_base." http://wiki.ros.org/move_base, 2024. visit date: Aug. 2, 2024.

[2] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 709–715, 2014.

[3] X. Huang, W. Chen, W. Zhang, R. Song, J. Cheng, and Y. Li, "Autonomous multi-view navigation via deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, pp. 13798–13804, 2021.

[4] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, pp. 60–68, Dec. 2015.

[5] K. Zhu, W. Chen, W. Zhang, R. Song, and Y. Li, "Autonomous robot navigation based on multi-camera perception," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 5879–5885, 2020.

[6] S. Xie, R. Song, Y. Zhao, X. Huang, Y. Li, and W. Zhang, "Circular accessible depth: A robust traversability representation for ugv navigation," *IEEE Trans. Robot.*, vol. 39, pp. 4875–4891, Sept. 2023.

[7] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 7160–7166, 2022.

[8] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *Int. Conf. Climbing and Walking Robots (CLAWAR)*, pp. 433–440, 2014.

[9] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, "Elevation mapping for locomotion and navigation using gpu," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 2273–2280, 2022.

[10] X. Huang, H. Deng, W. Zhang, R. Song, and Y. Li, "Towards multi-modal perception-based navigation: A deep reinforcement learning method," *IEEE Robot. Autom. Lett.*, vol. 6, pp. 4986–4993, Mar. 2021.

[11] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *ROBOTIK 2012*, pp. 1–6, 2012.

[12] P. Ondruska and I. Posner, "Deep tracking: Seeing beyond seeing using recurrent neural networks," in *Proc. AAAI Conf. Artif. Intell.*, pp. 3316–3367, 2016.

[13] P. Hu, A. Huang, J. Dolan, D. Held, and D. Ramanan, "Safe local motion planning with self-supervised freespace forecasting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 12732–12741, 2021.

[14] Y. Han, J. Banfi, and M. Campbell, "Planning paths through unknown space by imagining what lies therein," in *Proc. CORL*, pp. 905–914, 2021.

[15] A. Shaban, X. Meng, J. Lee, B. Boots, and D. Fox, "Semantic terrain classification for off-road autonomous driving," in *Proc. CORL*, pp. 619–629, 2022.

[16] K. Peng, J. Fei, K. Yang, A. Roitberg, J. Zhang, F. Bieder, P. Heidenreich, C. Stiller, and R. Stiefelhagen, "Mass: Multi-attentional semantic segmentation of lidar data for dense top-view understanding," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, pp. 15824–15840, Feb. 2022.

[17] W. Xie, X. Fang, and S. Wu, "2.5 d navigation graph and improved a-star algorithm for path planning in ship inside virtual environment," in *PHM-Besançon 2020*, pp. 295–299, 2020.

[18] H. Gim, M. Jeong, and S. Han, "Autonomous navigation system with obstacle avoidance using 2.5 d map generated by point cloud," in *ICCAS 2021*, pp. 749–752, 2021.

[19] H. Xie, X. Zhong, B. Chen, P. Peng, H. Hu, and Q. Liu, "Real-time elevation mapping with bayesian ground filling and traversability analysis for ugv navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 7218–7225, 2023.

[20] G. Erni, J. Frey, T. Miki, M. Mattamala, and M. Hutter, "Mem: Multi-modal elevation mapping for robotics and learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 11011–11018, 2023.

[21] S. Pütz, J. S. Simón, and J. Hertzberg, "Move base flex a highly flexible navigation framework for mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 3416–3421, 2018.

[22] S. Pütz, T. Wiemann, M. Kleine Piening, and J. Hertzberg, "Continuous shortest path vector field navigation on 3d triangular meshes for mobile robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[23] S. Macenski, D. Tsai, and M. Feinberg, "Spatio-temporal voxel layer: A view on robot perception for the dynamic world," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, 2020.

[24] T. Hines, K. Stepanas, F. Talbot, I. Sa, J. Lewis, E. Hernandez, N. Kottege, and N. Hudson, "Virtual surfaces and attitude aware planning and behaviours for negative obstacle navigation," *IEEE Robot. Autom. Lett.*, vol. 6, pp. 4048–4055, Mar. 2021.

[25] B. P. Gerkey and K. Konolige, "Planning and control in unstructured terrain," in *ICRA Workshop on Path Planning on Costmaps*, 2008.

[26] D. Chen, M. Zhuang, X. Zhong, W. Wu, and Q. Liu, "Rspmp: Real-time semantic perception and motion planning for autonomous navigation of unmanned ground vehicle in off-road environments," *Appl. Intell.*, vol. 53, pp. 4979–4995, Jun. 2023.

[27] H. Darweesh, E. Takeuchi, K. Takeda, Y. Ninomiya, A. Sujiwo, L. Y. Morales, N. Akai, T. Tomizawa, and S. Kato, "Open source integrated planner for autonomous navigation in highly dynamic environments," *J. Robot. Mechatron.*, vol. 29, pp. 668–684, Jun. 2017.

[28] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 802–807, 1993.

[29] "Maptoolbox: plugins to make autoware vector maps in unity." https://github.com/autocore-ai/MapToolbox, 2024. visit date: Aug. 20, 2024.

[30] G. Han, X. Qi, Y. Peng, C. Lin, Y. Zhang, and Q. Lu, "Early warning obstacle avoidance-enabled path planning for multi-auv-based maritime transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, pp. 2656–2667, Mar. 2022.