

Full length article



Constraint-aware motion planning for vehicles with terrain traversability assessment and optimization in construction scenarios

Chenlong Feng ^{a,b}, Jixin Wang ^b, Qi Wang ^c, Yuying Shen ^{a,*}, Yunda Wei ^b, Shaokai Zhang ^b, Shuyi Zhang ^b, Quan Zhang ^b

^a School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, 610031, China

^b Key Laboratory of CNC Equipment Reliability, Ministry of Education, School of Mechanical and Aerospace Engineering, Jilin University, Changchun 130025, China

^c State Key Laboratory of Fluid Power and Mechanical Systems, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Keywords:

Constraint-aware motion planning
Terrain traversability assessment
Construction scenarios with uneven terrain
Trajectory optimization
Safety constraints
Wheeled vehicle

ABSTRACT

Reliable motion planning for autonomous wheeled vehicles is crucial for improving construction automation and operational efficiency. Construction environments are characterized by complex obstacle distributions and uneven terrain. These factors present major challenges to traditional motion planning methods, which struggle to account for terrain-induced instability, handle complex terrain geometry efficiently, and generate dynamically feasible trajectories. A constraint-aware motion planning method with terrain traversability assessment is proposed to generate feasible, safe, and efficient motion trajectories. The method employs a Terrain Traversability Assessment (TTA) model that integrates vehicle dynamics and terrain geometry to compute continuous terrain traversability costs. Based on the TTA model, a terrain-aware Patch-RRT* algorithm is introduced to efficiently search feasible paths. The resulting paths are then used to construct safety constraints in complex environments. A constrained trajectory optimization framework is further designed using Bézier curves, incorporating safe, waypoint, continuous, and dynamic constraints while considering the coupling between the vehicle and terrain. Both simulation and real-world experiments demonstrate that the proposed method significantly improves trajectory quality thus achieving smoother paths, enhanced vehicle stability, and greater planning efficiency. The method provides a viable and effective solution for motion planning in complex construction scenarios involving wheeled vehicles.

1. Introduction

With the expansion of the modern construction industry, challenges such as the demand for construction efficiency and labor shortages has intensified [1,2]. Autonomous wheeled vehicles have become essential for enhancing construction efficiency and reducing labor costs [3]. The vehicles are extensively utilized for tasks such as site inspections, construction mapping, material transport, and earthwork operations, as shown in Fig. 1. However, construction environments are inherently unstructured, often characterized by uneven terrain, loose materials with inclined slopes, densely distributed equipment, rocks, and trees [4]. These factors increase the risk of vehicle rollover and loss of control, posing substantial challenges to autonomous and safe vehicle navigation. This instability adversely affects both safety and efficiency in construction operations [5].

Motion planning is a critical component of autonomous navigation systems for vehicles, enabling the generation of safe and smooth trajectories from start to goal [6]. While motion planning methods for 2D

environments are relatively well-developed, adapting these methods to construction environments presents unique challenges [7]. In particular, these methods fail to account for the effects of uneven terrain on the safe operation of vehicles. Addressing complex environments with uneven terrain conditions introduces three primary challenges: (1) assessing terrain traversability [8]; (2) developing motion planning methods that enable high computational efficiency when faced with complex environments [9]; (3) addressing the strong coupling between terrain and vehicle dynamics [10]. In response to these challenges, extensive research has focused on advance motion planning methods for vehicles navigating in complex environments with uneven terrain.

Accurately assessing the traversability of uneven terrain is essential to ensuring the safety of the vehicle. Point cloud data is often employed to represent obstacle distributions, allowing the workspace of vehicles to be classified into traversable and ultraversable areas [11–13]. However, point clouds may also be representative of terrain that is traversable by vehicles in complex environment, which makes it

* Corresponding author.

E-mail address: yshen@sjtu.edu.cn (Y. Shen).



Fig. 1. Application scenarios for wheeled vehicles.

impossible to apply simple methods for identifying traversable areas [14]. Furthermore, varying terrain slopes influence vehicle stability differently. Researchers apply statistical methods to estimate terrain geometry information from point clouds and convert the result into continuous terrain traversability scores to support motion planning [15]. Recently, a more comprehensive estimate of terrain has been achieved by regressing the terrain traversability scores with the help of deep learning without terrain geometry information [16]. Nevertheless, most of these methods assign terrain traversability scores based on geometry, ignoring the real effects of terrain on vehicles, including its impact on vehicle dynamics. The pose of the vehicle varies when traversing the same terrain at different speeds. Therefore, it is crucial to explore comprehensive terrain assessment models that consider the coupled relationship between vehicle dynamics and terrain features.

Motion planning methods for uneven terrain typically convert the terrain traversability score into large-scale cost maps integrated with other cost factors to guide the planning process [17,18]. Existing motion planning methods are categorized into five types: (1) reactive-based methods; (2) graph search-based methods; (3) sampling search-based methods; (4) optimization-based methods and (5) learning-based methods. Reactive-based methods utilize the dynamic model of the vehicle to compute a cost-optimal trajectory within a limited time horizon at each step. Despite the high computational efficiency of this method, the lack of prior knowledge about the environment results in lower-quality trajectories. Therefore, researchers have developed graph search-based and sampling search-based methods to obtain globally optimal trajectories [19]. Graph search-based methods tend to have low success rates in environments with complex obstacle distributions [20]. In contrast, sampling search-based methods utilize dense random sampling to thoroughly explore the environment, ensuring the asymptotic optimality of the trajectory [21,22]. Research indicates that sampling search-based methods offer distinct advantages for motion planning in complex environments with uneven terrain. In general, the methods struggle to balance the extensive computational cost with the need for planning efficiency. Additionally, random sampling often disregards vehicle dynamic constraints [23]. Learning-based methods attempt to learn navigation policies or trajectory mappings from data, enabling autonomous systems to adapt in partially known or dynamic environments [24]. Although learning-based methods such as reinforcement learning or end-to-end prediction have shown promise, their real-world application in safety-critical and highly unstructured construction environments remains limited due to data inefficiency, interpretability, and safety concerns [25].

In recent years, motion planning has increasingly been modeled as an optimal control problem, enabling trajectory generation that adheres to vehicle dynamics constraints [26,27]. This method reformulates the planning problem as a constrained nonlinear optimization

problem, efficiently solving for vehicle motion trajectories through numerical methods [28]. However, defining effective safety constraints for trajectory optimization remains challenging in complex construction environments, particularly on uneven terrain [29]. This results leads to trajectories that are not safely executable by the vehicle, thus affecting construction efficiency.

Significant progress has been made in motion planning methods for complex environments with uneven terrain. However, most existing methods overlook the interaction between terrain features and vehicle dynamics [30]. For instance, due to gravitational effects, the force needed to achieve a given acceleration differs when the vehicle is moving uphill versus downhill. Neglecting the factor often results in trajectories that are impossible for the vehicle to follow [31]. Therefore, motion planning methods that account for the coupled relationship between terrain features and vehicle dynamics are essential for generating feasible, safe, and efficient motion trajectories.

This study proposes a constraint-aware motion planning method that integrates vehicle-terrain traversability assessment, comprising four main components: traversable area identification and extraction, comprehensive terrain traversability assessment, efficient safe path search, and constraint-aware trajectory optimization. Valid Area Function (VAF) is introduced to achieve effective identification of traversable areas based on the geometric features of the terrain as well as the dynamical constraints of the vehicle. The inertial measurement unit (IMU) signals are leveraged to generate a continuous terrain traversability score. A time-sensitive terrain traversability assessment (TTA) model is constructed based on a 2D ConvLSTM neural network to learn IMU-represented terrain traversability scores from point cloud data. Based on the framework of Rapidly-exploring Random Trees Star (RRT*), Patch-RRT* is developed for adaptive terrain assessment to reduce computational costs and improve path search efficiency. The Patch-RRT* incorporates a 2D flow shape into 3D space, enabling adaptive terrain assessment and facilitating efficient path searching. The optimization problem is formulated using segmented Bézier curves. The path generated by Patch-RRT* is used to construct safety constraints. Waypoints, continuity, and dynamic constraints are also defined using Bézier curves. The objective is to minimize jerk while accounting for terrain features and vehicle dynamics, thereby ensuring safe vehicle operation. The contributions of this study can be summarized as follows:

(1) A comprehensive terrain traversability assessment method is proposed, which considers the coupled relationship between vehicle dynamics and terrain. The method enables effective identification of traversable areas and produces continuous terrain traversability scores to support terrain-aware planning.

(2) A path search method, Patch-RRT*, is introduced for adaptive terrain assessment, aiming to reduce computational cost and improve search efficiency.

(3) A constraint-aware trajectory optimization method is proposed that incorporates safety, waypoints, continuity, and dynamic constraints. It is capable of generating feasible, safe and efficient motion trajectories in complex environments with uneven terrain.

(4) Multi-scenario simulations and real-world experiments demonstrate the effectiveness of the proposed motion planning method.

2. Literature review

The study of vehicle motion planning methods for complex environments with uneven terrain is crucial for advancing automation in construction. In the past, researchers have conducted extensive studies on this problem and explored a variety of solutions. Motion planning methods for vehicles can be broadly classified into the following five categories: (1) reactive-based; (2) graph search-based; (3) sampling search-based; (4) optimization-based and (5) learning-based methods. Table 1 provides a detailed summary and comparison of several types of motion planning methods. This section provides an overview of the above methods.

Table 1
Summary of motion planning methods.

Methods	Description	Advantages	Challenges
Reactive-based	Real-time generation of local trajectories based on the current motion state of the vehicle.	Low computational cost; adaptable to dynamic environments.	Susceptible to local minima; strong dependence on static parameters.
Graph search-based	Global path generation using cost maps via graph traversal.	Provides globally optimal paths; suitable for complex obstacle environments.	Limited dynamic feasibility; inefficient path search.
Sampling-based	Searches for feasible paths through random sampling in high-dimensional spaces.	Flexible in unstructured terrain; ensures asymptotic optimality.	Weak in handling vehicle dynamics; high computational cost.
Optimization-based	Transforms planning into a constrained optimization problem incorporating smoothness and safety constraints.	Generates dynamically feasible, smooth trajectories.	Often neglects terrain-vehicle coupling.
Learning-based	Learns navigation policies from data via deep, reinforcement, or imitation learning methods.	Capable of adapting to unseen environments; enables end-to-end trajectory generation.	Requires large datasets; generalizes poorly; and lacks safety and interpretability.

2.1. Terrain traversability assessment methods

The terrain traversability assessment is based on the fact that normally vehicle movements should avoid challenging terrain such as steepness for safety. Existing methods can be broadly categorized into two groups, statistical based methods and learning based methods. Statistical based methods rely on recognizing the geometry of the terrain and mapping it to the effect on vehicle state. Yonghoon et al. [32] mapped the inclination characteristics of the terrain to the attitude angle of the vehicle to assess the traversability of the terrain. Jian et al. [15] fitted the point cloud to a plane and evaluated the traversability based on the degree of inclination as well as the roughness. Guizilini et al. [33] use kernel calculations to approximate the point cloud as an arbitrarily complex function, thus regressing the traversability of the terrain. However, most of the statistical based methods default to the vehicle being stationary on the corresponding terrain, ignoring the dynamic features of the vehicle. Learning-based methods focus on obtaining a semantic understanding of the environment, with the most common application being segmentation or categorization of the environment. Seo et al. [34] used a self-supervised learning approach to segment traversable terrain from visual sensors. Li et al. [35] constructed a terrain segmentation network to recognize the access environment. However, the limited categorization hardly reflects the specific challenges that terrain poses to vehicles.

2.2. Reactive-based motion planning methods

Reactive-based methods incorporate the current motion state of the vehicle to solve in real time for a customized cost-optimal motion trajectory over a finite time horizon [36]. The advantage of this method is the low computational costs. It can well deal with the dynamic obstacles appearing in the environment [37]. The most representative methods include Dynamic Window Approach (DWA) [38], Timed Elastic Band (TEB) [28]. DWA solves the trajectory planning problem by searching for speed that maximize a cost function in a collision-free region of the vehicle-accessible speed space [39]. The objective function generally includes goal distance, obstacle avoidance capability, and dynamic feasibility [40,41]. Traditional reactive-based methods are often applied to planning in 2D environments, and by further fusing 3D perceptual information, they are able to be applied to environments with complex terrain conditions [42]. For example, Chen et al. [43] proposed a real-time semantic perception and motion planning (RSPMP) for off-road environments by integrating DWA with a backward horizon planner to achieve safe autonomous navigation under complex terrain conditions. Lee et al. [44] proposed a novel Finite Memory Filtering (FMF)-based trajectory planning algorithm in unknown dynamic environments, called the Finite Distribution Estimation-based Dynamic Window Approach (FDEDWA), to generate more optimal paths in complex dynamic environments.

Although reactive-based methods are widely applied, their limitations remain evident. Due to the lack of the priori knowledge of the environment, reactive-based methods are susceptible to local minima [45] that leads to the generation of lower-quality trajectories. In addition, such methods rely heavily on static settings of the parameters in their cost functions and are unable to adapt to environmental features that may change significantly [46].

2.3. Graph search-based motion planning methods

Graph search-based methods are widely used for global path search in complex environments. It can find optimal paths based on non-uniform cost maps [47,48], typical methods are Hybrid A* [49], Field D* [50]. Calculating the terrain traversability scores by traversal and adding it to the cost map can solve motion planning problems in 3D environments [51]. For example, Thoresen et al. [52] developed the Traversability-Hybrid A*, which incorporates terrain traversability estimates to find paths in rough terrain that satisfy the feasibility of the vehicle dynamics and are near-optimal. Lee et al. [53] learn vehicle uncertainty induced by terrain from operator driving data, encode the learned uncertainty distribution into the traversability cost of the terrain, and finally combine Hybrid A* and Gaussian Processes (GP) to generate motion trajectories. Toscano et al. [54] proposed a DEM-based Asymmetric Inclination-Aware (DEM-AIA) vehicle motion planner for calculating estimates of the pitch and roll angles of the vehicle traveling through terrain segments to enable the consideration of the motion planning for terrain slope. Liu et al. [55] proposed a T-Hybrid A* motion planner that takes into account changes in vehicle position, establishes a new hybrid map representation that fuses 2D Grid map and 2.5D Elevation map, and explicitly takes safety as a constraint for motion planning, enabling autonomous navigation in unstructured environments.

However, graph search-based methods tend to design optimal trajectories at the regional global level, which results in ignoring the current vehicle state constraints and failing to satisfy the need of vehicles for stable driving postures in detail [56].

2.4. Sampling search-based motion planning methods

Sampling search-based motion planning methods have been shown to be an effective strategy for motion planning in complex inhomogeneous cost environments [57]. Jeong et al. [58] developed an optimal obstacle avoidance path planner for pose stabilization, the Rapidly Exploring Random Trees Star (Q-RRT*), for construction vehicles, which is able to minimize the distance to the target area and stabilize the pose of the vehicle. Sampling-based methods always need to maintain a global cost map, which is constantly searched in the map during the planning process. In order to speed up the search, Wang et al. [59] use RRT with variable step size based on octree maps to achieve

efficient path search. Krusi et al. [60] propose a global motion planner and terrain assessment method in a generic 3D environment based on Informed-RRT*. Optimized six-dimensional trajectories conforming to curvature constraints and continuity constraints are computed on a disordered point cloud map to improve computational efficiency. On this basis, Jian et al. [15] proposed an improved sample-based plane-fitting RRT*, PUTN-RRT*, for obtaining sparse trajectories in uneven terrain, using Gaussian Process Regression (GPR) to generate the traversability of dense trajectories interpolated from the sparse trajectories and using nonlinear model prediction control for local optimization to achieve trajectory planning for complex environment with uneven terrain.

Sampling search-based motion planning methods have fully demonstrated its advantages for planning in uneven terrain, but efficient sampling methods are shown to be strongly randomized, and their generated trajectories are difficult to satisfy the dynamical feasibility of the vehicle. In addition, high computationally cost estimates pose a significant challenge for efficient motion planning.

2.5. Optimization-based motion planning methods

Optimization-based methods usually construct the planning problem as a mathematical optimization problem and solve it using numerical optimization algorithms [12]. In recent years, optimization-based methods have been rapidly developed due to the feasibility of generating trajectories that match the dynamics of the vehicle and the efficient solution. Zheng et al. [61] proposed a multi-objective evolutionary trajectory optimization method based on Gaussian adaptive strategy. Spline interpolation is used to generate trajectory solutions for digital elevation topographic maps, followed by the construction of a multi-objective optimization function, which takes into account the parameters of trajectory length, uniformity, slope and undulation to solve the trajectory planning problem for uneven terrain. Considering the safety of vehicles in complex terrain, Wang et al. [17] constructed a penalty domain that responds to the terrain features and built a gradient-based spatio-temporal trajectory optimizer to generate safe driving trajectories for vehicles in uneven terrain. Han et al. [20] proposed a real-time trajectory optimization method that can generate high-quality motion trajectories under arbitrary environmental constraints, and further simplified the trajectory representation by exploiting the differential flatness property of the vehicle to achieve a more efficient solution. Zhou et al. [62] decoupled the collision-free motion planning problem of vehicles into a Dual-Loop Iterative Anchored Path Smoothing (DL-IAPS) problem and a Piecewise-Jerk Speed Optimization (PJSO) problem, which achieved significant driving performance improvements, including more robust and accurate obstacle avoidance, higher control feasibility and higher computational efficiency.

On the other hand, optimization-based methods are often combined with other motion planning methods to generate high-quality trajectories. Shen et al. [63] proposed a hierarchical trajectory planning method consisting of A*, RRT, and Local Trajectory optimization (A*-RRT-LTR) to address dynamic and incomplete constraints on uneven terrain. Xu et al. [10] built a terrain pose mapping and integrated it into trajectory optimization, which is used to optimize the paths searched by Hybrid A* to generate trajectories that satisfy the feasibility of vehicle dynamics.

Optimization-based methods have been shown to generate trajectories with higher feasibility. However, most trajectory optimization methods ignore the effect of terrain on vehicle dynamics, and the generated trajectories are often too conservative to be tracked well by the controller. Therefore, it is necessary to deal with the feasibility of vehicle dynamics coupled with terrain to generate safer and more reliable motion trajectories.

2.6. Learning-based motion planning methods

In recent years, learning-based motion planning methods have attracted increasing attention due to their ability to handle complex, high-dimensional environments. These methods can be broadly categorized into Deep Learning (DL), Reinforcement Learning (RL), and Imitation Learning (IL) methods [24].

DL-based methods have demonstrated notable potential in extracting high-level semantic features from complex environments and enabling end-to-end or hybrid trajectory generation strategies [25]. Li et al. [64] proposed a DL-based and quadratic programming method for trajectory planning in irregular terrains. In mining and construction, researchers have used CNN-based models for reverse trajectory generation of autonomous bulldozers [65] and data-driven excavation trajectory planning for unmanned excavators [66]. Recent advancements include the integration of vision-language-action (VLA) models for cooperative autonomous driving in complex traffic scenarios. V2X-VLM [67] and OpenDriveVLA [68] exploit large-scale pre-trained vision-language models to achieve interpretable, end-to-end motion planning via multimodal reasoning.

RL models the planning problem as a sequential decision-making process, where agents learn optimal policies by maximizing cumulative rewards through interaction with the environment [69]. Park et al. [70] proposed an iterative reward prediction method for autonomous vehicle trajectory planning, which estimates future rewards based on predicted states and integrates them into the learning process to enhance stability. Zhang et al. [71] proposed a predictive-prescriptive trajectory planning framework that utilizes a partially observable Markov decision process to model sequential decision problems and a deep RL method to learn high-quality strategies. RL relies on high-quality training data and suffers from poor robustness. For this reason, Huang et al. [72] proposed a smart vehicle trajectory planning method based on risk reachable sets. To accurately assess and avoid potential risks in the driveable area, they construct a reachable set representing driving risk regions. The initial trajectory is then generated by safety-based RL and projected into the reachable set. However, RL models typically require large training datasets, careful reward design, and face generalization difficulties when applied to diverse and rugged terrains.

Imitation learning seeks to learn motion behaviors by mimicking expert demonstrations, thereby reducing the need for extensive environment interaction [73,74]. This class of methods is particularly attractive for applications where expert demonstrations are readily available or where exploration is costly. Li et al. [75] integrated generative adversarial imitation learning and virtual reality for controlling construction robots to perform long-term collaborative construction tasks involving multiple subtasks. Gao et al. [76] proposed a keypoint-based visual imitation learning (K-VIL) method that automatically extracts sparse, object-centered, implementation-independent task representations from a small number of human demonstration videos. However, their generalization to unseen tasks and adaptation to new terrain features remain open challenges.

3. Methodology

3.1. Overall framework

The vehicle motion planning method for complex environments with uneven terrain is shown in Fig. 2 and consists of the following three main parts.

- Assessment of terrain traversability.** IMU, GPS, and LiDAR data are collected during vehicle motion in uneven terrain. Simultaneous Localization and Mapping (SLAM) methods such as Advanced Lidar Odometry and Mapping (A-LOAM) are used to construct Point-Cloud map [77]. Firstly, the VAF is constructed using the dynamics of the vehicle to identify traversable areas from PointCloud map. The disordered point cloud data in the traversable areas are mapped

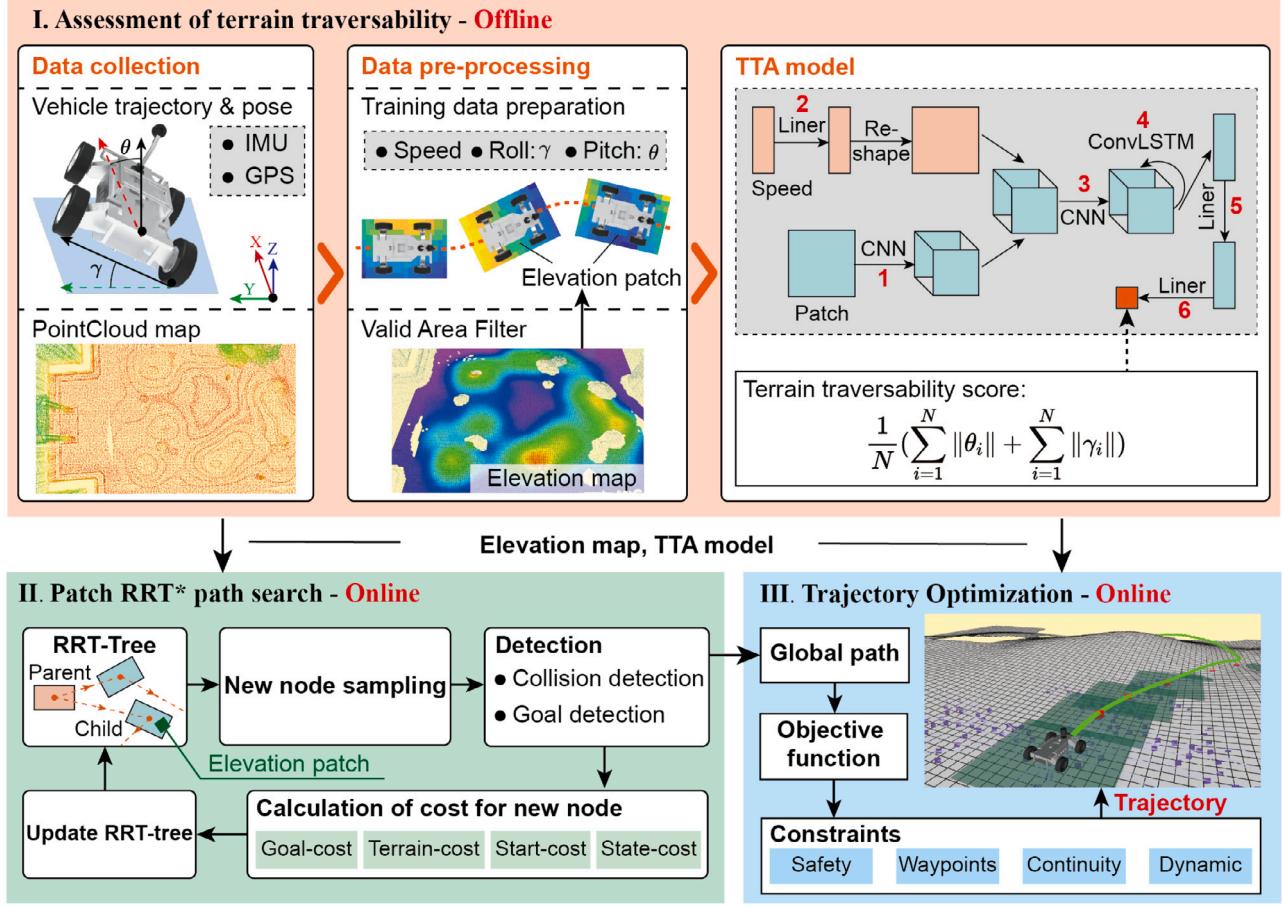


Fig. 2. Overall architecture of the proposed constraint-aware motion planning framework integrating terrain assessment, path search, and trajectory optimization.

into an indexable elevation map [78]. Secondly, elevation patches are extracted along the vehicle trajectory, and the vehicle's pose is used to assign traversability scores to the corresponding patches. Finally, the TTA model is developed using Convolutional Neural Network (CNN) and Convolutional Long Short Term Memory Network (ConvLSTM) that integrates vehicle dynamics.

2. Patch RRT* path search. Based on the framework of RRT*, Patch-RRT* is constructed by integrating the path search process with elevation patches. Combined with the trained TTA model offline, the terrain is evaluated on demand to achieve efficient path search.

3. Trajectory optimization. A constraint-aware trajectory optimization method is constructed. The trajectory of the vehicle is formulated as a segment Bézier curve. Safety constraints are constructed based on the paths generated by Patch-RRT*, which define the feasible space for trajectory optimization. Continuity, dynamical feasibility, and waypoint constraints are constructed through segmented Bernstein basis curve and its higher-order derivatives. Minimizing jerk is set as the objective function of the optimization, while the traversability cost of the terrain is included to generate safe, smooth, and feasible motion trajectories.

In order to express the proposed method more clearly, the key mathematical notations involved in the method are summarized in Table 2 for reference.

3.2. Assessment of terrain traversability

3.2.1. Valid area filter

Accurate environment perception is the basis for safe and efficient motion planning. For traditional planning, obstacles are represented by inflated point clouds. However, for complex environment with uneven

terrain, the point cloud may also represent traversable areas. The VAF based on the dynamic of vehicle is constructed to identify the traversable areas from original PointCloud map. The points located in the traversable areas in the original PointCloud map are defined as:

Definition 1. Given point $p = [p_x, p_y, p_z]^\top$ in the original PointCloud map $M = [p_x, p_y, p_z]$, and $D = [p_x^D, p_y^D, p_z^D]$ is the neighborhood of p with radius r . The value of r must exceed the smallest sphere radius that can enclose the vehicle, for $\forall p^D = [p_x^D, p_y^D, p_z^D]^\top \in D$ satisfying:

$$\left| \arctan \left(\frac{p_z - p_z^D}{\sqrt{(p_x - p_x^D)^2 + (p_y - p_y^D)^2}} \right) \right| \leq \alpha \quad (1)$$

where α is the maximum climbing gradient of the vehicle. Then p is the point in the traversable areas.

Based on Definition 1, the PointCloud map M^s containing only traversable areas can be extracted. Due to the lack of sensor accuracy, there are a large number of noise points in the original PointCloud map. The relaxation factor ω is introduced while performing the traversable areas extraction. The algorithm of VAF is shown in Algorithm 1.

The determination of the parameters in the VAF module is based on the physical features and dynamics of the vehicle. The value of the radius r is slightly larger than the smallest sphere that can completely cover the vehicle body to ensure that all relevant terrain around the vehicle is taken into account. The maximum gradient α is set based on the rated maximum climbing capacity of the vehicle, which is determined from the specifications provided by the manufacturer or experimentally by means of an incline test. The relaxation factor ω

Table 2
Notation summary.

Notation	Description
$p = [p_x, p_y, p_z]^\top$	Point in original PointCloud map \mathbf{M} .
$\mathbf{D} = [p_1^D, p_2^D, \dots, p_d^D]^\top$	Neighborhood of the point p with radius r .
α	Maximum climbing gradient of vehicle.
ω	Relaxation factor for filtering noisy points.
\mathbf{M}^s	Extracted PointCloud map of traversable areas.
$\mathbf{x} = [x, y, \psi]^\top$	Vehicle state vector.
$\mathbf{R} = [x_b, y_b, z_b] \in \text{SO}(3)$	Vehicle pose matrix.
γ, θ, ψ	Roll, pitch and yaw angles of vehicle.
\mathbf{H}	Elevation map.
$\mathbf{H}_i^p \in \mathbf{H}$	Elevation patch.
$\sigma = [x, y]^\top$	Differential flat output space of vehicle.
$\mathcal{N} = \{\sigma, \psi, \mathbf{H}_{\text{node}}, J_{\text{node}}, \mathcal{N}_{\text{parent}}\}$	Node structure in Patch-RRT*.
$J_{\text{node}}, J_{\text{terrain}}, J_{\text{goal}}, J_{\text{start}}$	Cost components in Patch-RRT* node.
$\tau \in [0, 1]$	Normalized time parameter.
t	Actual time parameter.
$b_{n,i}(\tau)$	i th Bernstein basis polynomial of degree n .
n	Degree of the Bernstein polynomial basis.
$\mathbf{b}_n(\tau) = [b_{n,0}, b_{n,1}, \dots, b_{n,n}]^\top$	Vector of Bernstein polynomial basis.
$\mathbf{c}_j = [c_{j,1}, c_{j,2}, \dots, c_{j,n}]^\top$	Control points of Bézier trajectory for j th segment.
$B_j(\tau)$	Bézier trajectory in j th segment.
$\mathbf{s} = [s_1, s_2, \dots, s_m]^\top$	Time scaling factors for each segment.
$\mathbf{T} = [T_0, T_1, \dots, T_m]^\top$	Start/end time of each trajectory segment.
m	Number of segments of Bézier trajectory.
T^a	Total duration of the entire trajectory.
ϵ	Objective function for trajectory optimization.
$f(t)$	Segmented Bézier trajectory.
$\mathbf{j}(t) = [f_x^{(3)}(t), f_y^{(3)}(t)]^\top$	3-rd order derivative of Bézier trajectory for x, y axis.
ρ_{ter}	Weight for terrain cost in objective function.
$\rho_T = [\rho_{T,1}, \rho_{T,2}, \dots, \rho_{T,m}]^\top$	Weights for segment time duration.
$\mathcal{S} = \{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_k\}$	Final path returned by Patch-RRT*.
ϱ	Safety constraints rectangle threshold.
$\mathbf{A} = [r_1^\top, r_2^\top, \dots, r_k^\top]$	Safety constraints.
$\mathbf{r} = [r_x^-, r_x^+, r_y^-, r_y^+]^\top$	Rectangular bounds of safe region.
$a_{j,i}^{(l)}$	Difference coefficient of l th derivative at control point i of segment j .
\mathcal{T}	Patch-RRT* search tree.
$\mathcal{Q} = \{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_g\}$	Neighboring nodes of current node \mathcal{N} .
\mathbb{P}	Vehicle trajectory dataset.
\mathbb{I}	Vehicle IMU dataset.
Vel_{\max}, Vel_{\min}	Maximum and minimum linear speed of the vehicle.
Acc_{\max}, Acc_{\min}	Maximum and minimum linear acceleration of the vehicle.

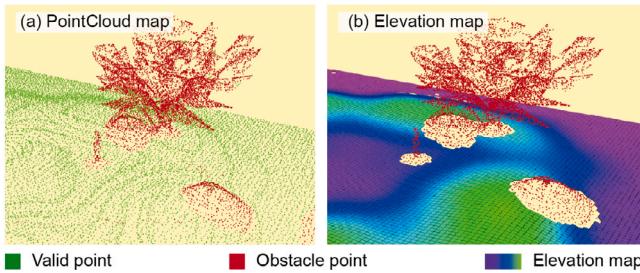


Fig. 3. (a) The PointCloud map after VAF, in which the red point cloud represent the untraversable areas, and (b) the Elevation map \mathbf{H} based on the PointCloud map \mathbf{M}^s , in which the blank areas are the untraversable areas.

serves to mitigate the effects of sensor noise and is tuned empirically by balancing sensitivity to slope irregularities and robustness to outliers.

Fig. 3(a) gives the PointCloud map of demonstration scenario and the \mathbf{M}^s of the traversable areas after VAF. The disordered point cloud in \mathbf{M}^s is mapped into Elevation map using the method in Ref. [78], as shown in Fig. 3(b). In Elevation map, the terrain elevation value can be easily retrieved from the grid. For untraversable areas, the retrieved elevation values are NULL. Subsequent Elevation map will be used to train TTA model and Patch-RRT* path search.

3.2.2. Terrain traversability assessment model

The traversability of the terrain is strongly coupled with vehicle motion. The motion state can be represented using the state $\mathbf{x} =$

$[x, y, \psi]^\top \in \mathbb{R}^3$ as well as the pose $\mathbf{R} = [x_b, y_b, z_b] \in \text{SO}(3)$. The most direct effect of terrain on the vehicle is reflected in the pose \mathbf{R} , especially the roll angle γ and pitch angle θ . This study introduces the TTA model, which integrates the geometric features of terrain and the vehicle motion information.

The training data for the TTA model include the vehicle motion trajectory and the corresponding elevation patches \mathbf{H}_i^p , each slightly larger than the vehicle's footprint. The output of the model is defined as the average of the absolute values of roll (γ) and pitch (θ) angles while traversing the elevation patch \mathbf{H}_i^p :

$$J_i = \frac{1}{N} \sum_{i=1}^N (\|\theta_i\| + \|\gamma_i\|) \quad (2)$$

where N denotes the number of trajectory points in the patch \mathbf{H}_i^p ; the motion trajectory is obtained from the GPS. The patch is extracted from the Elevation map \mathbf{M}^s . The pose of the vehicle is obtained from the IMU. The GPS and IMU data are time-aligned using signal timestamps. The data processing flow is shown in Fig. 4.

The structure of TTA model is shown in Fig. 2, and the hyperparameters are shown in Table 3. The TTA model employs a deep neural network architecture that combines 2D convolution and ConvLSTM, which is capable of capturing the features of terrain changes in both spatial and temporal dimensions. The input to the TTA model is a time series of elevation patches extracted along the continuous trajectory of the vehicle. Each elevation patch is a 2D grid of 32×32 pixels, representing the terrain elevation information within the localized areas of the vehicle. In order to reflect the trend of terrain change during continuous motion, the TTA model uses a sequence input with

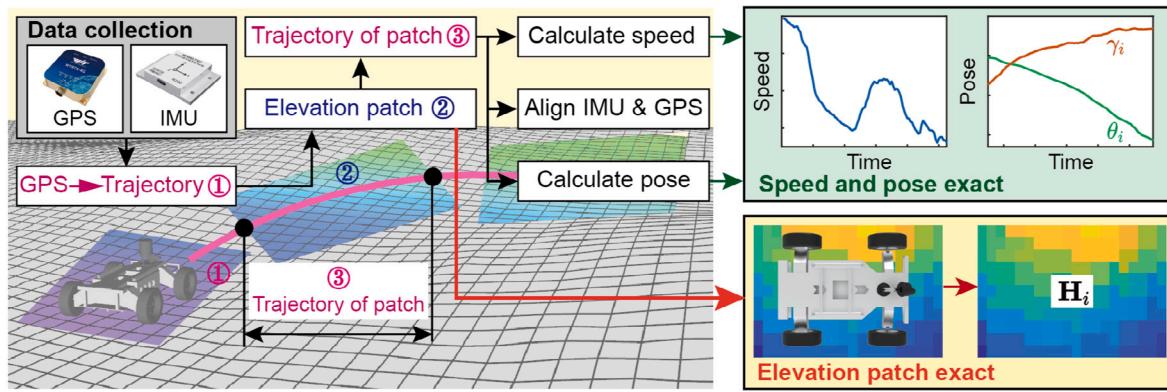


Fig. 4. Generation of training data by aligning GPS-IMU trajectory data with elevation patches. Vehicle trajectories (①) are collected using GPS and IMU, then projected onto a global elevation map to extract elevation patches (②). Within each patch, the corresponding sub-trajectory (③) is used to compute speed and pose. The resulting data includes speed, roll and pitch, and the elevation patch H_i .

Algorithm 1 Valid Area Filter

```

Input: Original PointCloud map M, Radius r, Maximum climbing gradient  $\alpha$ .
Output: PointCloud map only contain traversable areas  $M^s$ .
1:  $M^s \leftarrow \emptyset$ ;
2: > Iterating over all the points in M.
3: for  $p \in M$  do
4:   > Compute neighborhood of p.
5:   D = GetNeighbors( $p, r$ );
6:   count  $\leftarrow 0$ ;
7:   len  $\leftarrow \text{Len}(D)$ ;
8:   > Compute number of points in D that satisfy Eq. (1).
9:   for  $p^D \in D$  do
10:     $\alpha_{p^D} = |\arctan\left(\frac{p_z - p_z^D}{\sqrt{(p_x - p_x^D)^2 + (p_y - p_y^D)^2}}\right)|$ ;
11:    if  $\alpha_{p^D} \geq \alpha$  then
12:      count  $\leftarrow$  count +;
13:    end if
14:   end for
15:   > Determine if p is in traversable areas
16:   if ( $count / len \leq \omega$ ) then
17:      $M^s \leftarrow M^s \cup p$ ;
18:   end if
19: end for
20: return  $M^s$ 

```

a time window length of 10. Each input to the TTA model is a tensor of shape $(B, T = 5, C = 1, H = 32, W = 32)$, where B is the batch size, T is the number of temporal steps, C is the number of channels, and H and W are the height and width of the patch, respectively.

The TTA model structure is designed as a multi-stage feature extraction and fusion architecture. First, the input elevation patch is passed through a 2D convolutional layer with 3×3 convolutional kernels to extract spatial features. Then, Batch Normalization (BN) with ReLU activation function is used to enhance the stability and nonlinear expression of the TTA model. In order to take into account the effect of vehicle dynamic states on terrain traversability, the model introduces vehicle speed as an auxiliary channel input. The speed is normalized to the interval $[0, 1]$ to improve the numerical stability and training efficiency of the network. This normalization process scales linearly based on the upper and lower limits of the actual vehicle speed. The normalized speed input is transformed into a 2D activation map of the same size as the convolutional feature map by a fully connected layer. This activation map is then fused by channel-by-channel multiplication

with the convolutional feature map extracted by convolution of the elevation patch, allowing dynamic adjustment of different regions at the feature level. This allows the model to capture the real effects of terrain on vehicle states under different speed conditions. This mechanism effectively enhances the ability of the TTA model to model the terrain-vehicle coupling relationship.

The fused multi-frame feature sequences are fed into a ConvLSTM layer with a kernel size of 3×3 , which is used to model the evolutionary properties of the elevation patch in the time dimension. The output of ConvLSTM is further compressed into feature vectors through a fully connected layer after dimensionality reduction via a 2D average pooling layer, and overfitting is suppressed by the Dropout mechanism. Finally, the TTA model outputs a scalar traversability score that is used to regress the average absolute pose response (the sum of roll angle $\|\gamma\|$ and pitch angle $\|\theta\|$) of vehicle. Each output node of the TTA model is activated with ReLU to maintain sparsity and facilitate network training convergence.

3.3. Patch-RRT* path search

Conventional RRT* finds an asymptotically optimal path by randomly sampling within the differential flat output space $\sigma = [x, y]^\top$ of the vehicle. However, this method neglects the influence of terrain on vehicle motion during the sampling process. Therefore, the Patch-RRT* for on-demand evaluation of terrain traversability is constructed. Patch-RRT* utilizes elevation patches to guide the search process and improve efficiency.

Construct the node of Patch-RRT* as $\mathcal{N} := \{\sigma, \psi, \mathbf{H}_{node}, J_{node}, \mathcal{N}_{Parent}\}$, where \mathcal{N}_{Parent} is the parent node connected to the current node \mathcal{N}_{now} . ψ represents the vehicle yaw angle, defined as the heading angle from \mathcal{N}_{Parent} to the current node. \mathbf{H}_{node} is the elevation patch obtained associated with the current differential flat output σ of vehicle; J_{node} is the cost of the current node, described as:

$$J_{node} = \lambda_t J_{terrain} + \lambda_g J_{goal} + \lambda_{start} J_{start} + \lambda_{state} J_{state} \quad (3)$$

where $J_{terrain}$ is the terrain traversability scores calculated from the TTA model. J_{goal} is the Euclidean distance from the current node to the goal. J_{start} is the cumulative Euclidean distance of the path from the current node to the start. J_{state} is the change in ψ from \mathcal{N}_{Parent} to the current node $\|\mathcal{N}_{Parent} \cdot \psi - \mathcal{N}_{now} \cdot \psi\|$. Each λ_i is a weight parameter used to balance the cost components. The Patch-RRT* is shown in Algorithm 2.

3.4. Trajectory optimization

3.4.1. Trajectory optimization problem formulation

The trajectory of vehicle in complex environment with uneven terrain can be parameterized in terms of its differential flat output σ as

Table 3
Hyperparameters and layer configurations of TTA model ('No.' corresponding to Fig. 2).

No.	Layers	Filter hidden units	Kernel size	Stride/Rate	Padding	Activation
1	2D-Conv	[16, 256]	3	1	2	–
	2D-AvgPool	–	3	1	–	ReLU
	BN	–	–	–	–	–
2	Fully connected	[16, 256]	–	–	–	–
	Dropout	–	–	[0.1, 0.5]	–	–
	2D-Conv	[16, 256]	3	1	–	–
3	2D-AvgPool	–	3	1	–	ReLU
	BN	–	–	–	–	–
	ConvLSTM	[16, 256]	3	1	2	–
4	2D-AvgPool	–	3	1	–	ReLU
	Fully connected	[16, 256]	–	–	–	–
	Dropout	–	–	[0.1, 0.5]	–	–
5	Fully connected	1	–	–	–	–
6	Fully connected	1	–	–	–	–

a function of time t . The trajectory is parameterized as a segment Bézier curve using the Bernstein polynomial basis. The Bernstein polynomial basis is defined as [79]:

$$b_{n,i}(\tau) = C_n^i \cdot \tau^i \cdot (1 - \tau)^{n-i}, C_n^i = \frac{n!}{i!(n-i)!} \quad (4)$$

where n denotes the degree of freedom of the basis; C_n^i is the binomial coefficient. $\tau \in [0, 1]$ is the normalized time parameter. The vector $\mathbf{b}_n(\tau) = [b_{n,0}, b_{n,1}, \dots, b_{n,n}]^\top \in \mathbb{R}^{(n+1) \times 1}$ of Bernstein polynomial basis is obtained. Then the Bézier trajectory can be expressed as:

$$B_j(c_j, \tau) = c_j^\top \mathbf{b}_n(\tau) = \sum_{i=0}^n c_{j,i} b_{n,i}(\tau) \quad (5)$$

where $c_j = [c_{j,0}, c_{j,1}, \dots, c_{j,n}]^\top \in \mathbb{R}^{(n+1) \times 1}$, c_j is the set of control points of the j th segment of the Bézier trajectory. The Bézier trajectory lies within the convex hull formed by all control points, with the first and last control points lying on the curve. For an m -segment Bézier trajectory with time variable t , since the time of each segment trajectory is on fixed interval $[0, 1]$, scaling factor $s = [s_1, s_2, \dots, s_m]^\top$ is introduced to scale the time t . Then the m -segment Bézier trajectory can be expressed as:

$$f(t) = \begin{cases} s_1 \cdot B_1(c_1, \frac{t - T_0}{s_1}), & t \in [T_0, T_1] \\ s_2 \cdot B_2(c_2, \frac{t - T_1}{s_2}), & t \in [T_1, T_2] \\ \vdots & \vdots \\ s_m \cdot B_m(c_m, \frac{t - T_{m-1}}{s_m}), & t \in [T_{m-1}, T_m] \end{cases} \quad (6)$$

where $T = [T_0, T_1, \dots, T_m]^\top$ denotes the time of each segment for trajectory. The total time of segmented Bézier trajectory is $T^a = T_m - T_0$.

Based on the constructed m -segment Bézier trajectory, the objective function of the optimization problem is proposed in this study:

$$\epsilon = j(t)^\top j(t) + \rho_{ter} \cdot \mathcal{L}(f_x(t), f_y(t)) \quad (7)$$

where $f_x(t)$ and $f_y(t)$ denote the segmented Bézier trajectories on the x and y axis, respectively. $j(t) = [f_x^{(3)}(t), f_y^{(3)}(t)]^\top$ which denote 3rd order derivative of Bézier trajectory. The 3rd order derivative of the trajectory is denoted as jerk, the smaller the jerk the smoother the trajectory. The objective function takes the minimum jerk of the trajectory as the optimization objective. ρ_{ter} is the constant. $\mathcal{L}(f_x(t), f_y(t))$ is the terrain traversable scores at time t obtained from TTA model. The objective function for trajectory optimization can be formulated as:

$$\min_{c_x, c_y, T} \int_0^{T^a} (j(t)^\top j(t) + \rho_{ter} \cdot \mathcal{L}(f_x(t), f_y(t))) dt + \rho_T^\top T \quad (8)$$

where c_x , c_y denote the trajectory control points on the x and y axis, respectively, $\rho_T = [\rho_{T,1}, \rho_{T,2}, \dots, \rho_{T,m}]^\top$ is cost weights for the length of time of each segment Bézier trajectory.

3.4.2. Safety constraints

Constructing reliable safety constraints for trajectory optimization in complex environments with uneven terrain is crucial. After obtaining the asymptotically optimal path $S = \{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_u\}$, which contains u nodes and is generated by Patch-RRT*, the safe space in the environment can be extracted to form safety constraints for back-end optimization. This ensures that the optimized trajectories allow for safe vehicle navigation.

For each node of Patch-RRT* path $\mathcal{N}_i \in S$, a rectangular safety region is constructed based on the nearest-obstacle distance retrieved from the Elevation map \mathbf{H} . The initial region is defined as the largest axis-aligned rectangle that fits inside a safe circle centered at the node, with its sides aligned with the global x and y axis for ease of constraint formulation. This rectangle is then iteratively expanded in the x and y directions by querying neighboring grid cells, until the largest feasible safety rectangle is identified. To avoid excessively large solution spaces that may degrade optimization efficiency, a threshold φ is imposed on the maximum rectangle size. The overall process is shown in Fig. 5, and the safety constraints found using this mechanism are able to cover the topological safety space of the path nodes.

Due to the dense distribution of path nodes, the safety regions of adjacent nodes may overlap, introducing redundant constraints and increasing computational complexity. To address this, a redundancy elimination mechanism is designed:

- If the current node lies outside the most recent rectangle but within the second-to-last one, the most recent rectangle is deemed redundant and removed;
- If the node lies outside all previous rectangles, a new rectangle is generated and added to the safety constraints set.

This process is detailed in Algorithm 3, and the final safety constraints is denoted as:

$$\mathbf{A} = [\mathbf{r}_1^\top, \mathbf{r}_2^\top, \dots, \mathbf{r}_k^\top] \quad (9)$$

where $\mathbf{r}_j = [r_{x,j}^-, r_{x,j}^+, r_{y,j}^-, r_{y,j}^+]^\top$, $r_{x,j}^-$, $r_{x,j}^+$ denotes the left and right boundaries of the j th rectangle on the x axis, $r_{y,j}^-$, $r_{y,j}^+$ denotes the down and up boundaries of the j th rectangle on the y axis.

Owing to the convex envelope property of the Bézier curve, the safety constraints for trajectory optimization can be formulated as a set of constraints on the control points of the j th segment of for segmented Bézier trajectory:

$$\begin{cases} \min(c_{x,j}) \geq r_{x,j}^-, & \max(c_{x,j}) \leq r_{x,j}^+ \\ \min(c_{y,j}) \geq r_{y,j}^-, & \max(c_{y,j}) \leq r_{y,j}^+ \end{cases} \quad (10)$$

where $c_{x,j}$ and $c_{y,j}$ denote the control points of the j th segment of the Bézier trajectory on the x and y axes, respectively.

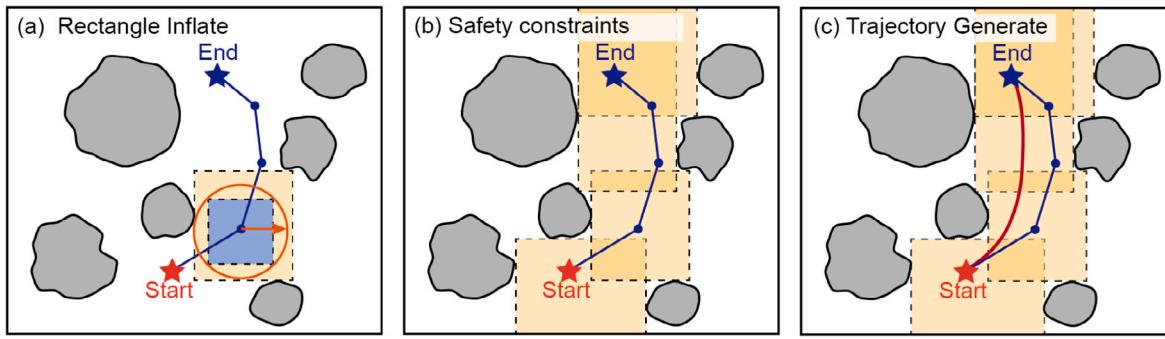


Fig. 5. (a) The process of inflating the safety constraints rectangle; (b) the process of using the safety constraints to cover path nodes and (c) the results of trajectory optimization using the generated safety constraints.

3.4.3. Other constraints

In addition to safety constraints, this study introduces several other constraints to ensure the continuity and dynamic feasibility of the generated trajectories. An auxiliary variable $a_{j,i}^{(l)}$ is introduced and defined as follows:

$$a_{j,i}^{(0)} = c_{j,i}, \quad a_{j,i}^{(l)} = \frac{n!}{(n-l)!} \left(a_{j,i+1}^{(l-1)} - a_{j,i}^{(l-1)} \right) \quad (11)$$

where l is the order of the derivative and n is the degree of freedom of the Bernstein basis. $a_{j,i}^{(l)}$ denotes the coefficient of the l th order difference quotient at the i th control point of the j th segment of the Bézier trajectory.

Waypoints constraints: equation constraints on trajectories at the start and goal with n th order derivatives. Given the l th order derivative $d_j^{(l)}$ at the start or goal of the j th trajectory segment, where $l \leq n$, the constraint is defined as:

$$a_{\mu j,0}^{(l)} \cdot s_j^{1-l} = d_j^{(l)} \quad (12)$$

Continuity Constraints: equation constraints where l th ($0 \leq l \leq n-1$) order derivatives of the connecting points of the two segment trajectories are all equal. For j th and $(j+1)$ th segment trajectories, the equation constraint can be formulated as:

$$a_{j,n}^{(l)} \cdot s_j^{1-l} = a_{j+1,0}^{(l)} \cdot s_{j+1}^{1-l}, \quad a_{j,i}^{(0)} = c_{j,i} \quad (13)$$

Dynamical Feasibility Constraints, inequality constraints imposed on the higher-order derivatives of the trajectories to satisfy the dynamical feasibility of the vehicle. The speed and acceleration of the vehicle are constrained to be within the intervals $[Vel_{\min}, Vel_{\max}]$ and $[Acc_{\min}, Acc_{\max}]$, respectively. For the control points of the j th segment of the trajectory, the inequality constraint can be formulated as:

$$\begin{aligned} Vel_{\min} &\leq n \cdot (c_{j,i} - c_{j,i-1}) \leq Vel_{\max} \\ Acc_{\min} &\leq n \cdot (n-1) \cdot \frac{(c_{j,i} - 2c_{j,i-1} + c_{j,i-2})}{s_j} \leq Acc_{\max} \end{aligned} \quad (14)$$

In summary, the objective function (Eq. (8)), the equality constraints (Eqs. (12)–(13)), and the inequality constraints (Eqs. (11) and (14)) for the established nonlinear constrained optimization problem are presented, respectively. The trajectory optimization framework assumes that the linear speed and acceleration of vehicle remain within fixed upper and lower bounds. While this simplified modeling approach is broadly applicable across various vehicle operation scenarios, it demonstrates enhanced validity and robustness within the following operational envelope: (1) the vehicle operates at relatively low speeds, typically within the range of 0.2–5.0 m/s; (2) the center of gravity for vehicle change is limited, and the dynamics is relatively stable; (3) the vehicle control system has a high execution frequency (≥ 10 Hz). This modeling approach is particularly suitable for mobile robotic tasks with high control frequency requirements, such as low-speed inspection, material handling, and fixed-point operations. Its main advantage lies in significantly reducing the computational complexity of trajectory optimization, while maintaining control accuracy and feasibility. This

facilitates efficient deployment on embedded platforms with limited computing resources.

This nonlinear constrained optimization problem is solved using the Powell-Hestenes-Rockafellar Augmented Lagrangian Method (PHR-ALM) combined with the Limited Memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm. PHR-ALM transforms the original constrained optimization problem into a series of unconstrained subproblems that can be solved iteratively by introducing penalty parameter to the constraints in the Lagrangian function. The initial penalty parameter for the constraint is set to $\rho_0 = 1.0$, which is dynamically adjusted during the solution process according to the degree of constraint violation. The constraint violation tolerance is set to 10^{-4} , and when a constraint violates this threshold, its penalty parameter is updated according to $\rho_{k+1} = \min(10\rho_k, 10^6)$. The initial value of the Lagrange multiplier is set to 0 and updated by gradient descent at the end of each external iteration to improve the accuracy of the multiplier estimation.

The unconstrained subproblems formed in each iteration are solved efficiently by the L-BFGS algorithm. L-BFGS approximates the Hessian matrix by finite-time historical gradient information, which has the advantages of high efficiency, stability and low memory consumption. The L-BFGS uses information from the last 5 iterations to update the approximate value of the Hessian matrix. The line search strategy employs methods that satisfy the Strong Wolfe condition to ensure that the objective function has sufficient descent and suitable curvature. For each unconstrained subproblem, convergence is considered when the relative change in the objective function is less than 10^{-6} or the gradient parameter is less than 10^{-5} . For the outer main iteration, the optimization process is terminated when the maximum constraint violation is less than the feasibility tolerance, and the relative improvement of the objective function is less than 10^{-5} .

The optimization algorithm is implemented based on C++, relying on the Eigen and LBFGS++ libraries to complete matrix operations and optimization solutions. All calculations were performed on a computing platform equipped with an Intel Core i5-10400F processor (2.9 GHz) and 16 GB of RAM.

3.5. Method integration design

The described method accomplishes the task of vehicle motion planning for complex environments through the collaborative interaction of multiple components, As shown in Algorithms 4 and 5.

3.6. Trajectory quality metrics definition

To comprehensively assess the quality of the trajectories generated by the motion planning method proposed in this study, the following four categories of trajectory quality metrics are defined:

Algorithm 2 Patch-RRT*

Input: Elevation map \mathbf{H} , Node at start of trajectory $\mathcal{N}_{\text{start}}$, Node at goal of trajectory $\mathcal{N}_{\text{goal}}$, Maximum number of samples k .

Output: Asymptotically optimal path S .

- 1: \triangleright **Construction of the RRT tree \mathcal{T} .**
- 2: $\mathcal{T} \leftarrow \{\mathcal{N}_{\text{start}}\}$;
- 3: **while** $i \in k$ **do**
- 4: \triangleright **Randomly sampling vehicle state σ .**
- 5: $\sigma \leftarrow \text{RandomSample}()$
- 6: \triangleright **Iterate through the tree \mathcal{T} to find the nearest node to σ .**
- 7: $\mathcal{N}_{\text{nearest}} \leftarrow \text{FindNearest}(\mathcal{T}, \sigma)$
- 8: \triangleright **Growing \mathcal{T} from node $\mathcal{N}_{\text{nearest}}$ to σ to generate new nodes.**
- 9: $\mathcal{N}_{\text{new}} \leftarrow \text{Steer}(\mathcal{N}_{\text{nearest}}, \sigma)$
- 10: **if** $\text{CollisionDetect}(\mathcal{N}_{\text{new}}, \mathcal{N}_{\text{nearest}}) == \text{false}$ **then**
- 11: \triangleright **Insert node \mathcal{N}_{new} into \mathcal{T} .**
- 12: $\text{InsertTree}(\mathcal{T}, \mathcal{N}_{\text{new}})$
- 13: \triangleright **Get the elevation patch of \mathcal{N}_{new}**
- 14: $\mathcal{N}_{\text{new}}.\mathbf{H}_{\text{node}} \leftarrow \text{PatchGet}(\mathcal{N}_{\text{nearest}}, \mathcal{N}_{\text{new}}, \mathbf{H})$
- 15: \triangleright **Calculate the cost of terrain, goal, etc. of \mathcal{N}_{new}**
- 16: $\mathcal{N}_{\text{new}}.\mathbf{J}_{\text{node}} \leftarrow \text{ComputeCost}(\mathcal{N}_{\text{nearest}}, \mathcal{N}_{\text{new}})$
- 17: \triangleright **Constructing neighborhood Ω_{near} of \mathcal{N}_{new}**
- 18: $\Omega_{\text{near}} \leftarrow \text{FindNear}(\mathcal{T}, \mathcal{N}_{\text{new}})$
- 19: \triangleright **Re-select the $\mathcal{N}_{\text{parent}}$ for \mathcal{N}_{new}**
- 20: **for** $\mathcal{N}_{\Omega} \in \Omega_{\text{near}}$ **do**
- 21: $\mathcal{N}_{\text{temp}} \leftarrow \mathcal{N}_{\text{new}}$
- 22: $\mathcal{N}_{\text{temp}}.\mathbf{H}_{\text{node}} \leftarrow \text{PatchGet}(\mathcal{N}_{\Omega}, \mathcal{N}_{\text{temp}}, \mathbf{H})$
- 23: \triangleright **Calculate the node cost according to Eq. (3).**
- 24: $\mathcal{N}_{\text{temp}}.\mathbf{J}_{\text{node}} \leftarrow \text{ComputeCost}(\mathcal{N}_{\Omega}, \mathcal{N}_{\text{temp}})$
- 25: **if** $\mathcal{N}_{\text{temp}}.\mathbf{J}_{\text{node}} < \mathcal{N}_{\text{new}}.\mathbf{J}_{\text{node}}$ **then**
- 26: $\mathcal{N}_{\text{new}} \leftarrow \mathcal{N}_{\text{temp}}$
- 27: $\mathcal{N}_{\text{new}}.\mathcal{N}_{\text{Parent}} \leftarrow \mathcal{N}_{\Omega}$
- 28: **end if**
- 29: **end for**
- 30: \triangleright **Reconnect tree \mathcal{T} in neighborhood Ω_{near}**
- 31: **for** $\mathcal{N}_{\Omega} \in \Omega_{\text{near}}$ **do**
- 32: $\mathcal{N}_{\text{temp}} \leftarrow \mathcal{N}_{\Omega}$
- 33: $\mathcal{N}_{\text{temp}}.\mathbf{H}_{\text{node}} \leftarrow \text{PatchGet}(\mathcal{N}_{\text{new}}, \mathcal{N}_{\text{temp}}, \mathbf{H})$
- 34: $\mathcal{N}_{\text{temp}}.\mathbf{J}_{\text{node}} \leftarrow \text{ComputeCost}(\mathcal{N}_{\text{new}}, \mathcal{N}_{\text{temp}})$
- 35: **if** $\mathcal{N}_{\text{temp}}.\mathbf{J}_{\text{node}} < \mathcal{N}_{\Omega}.\mathbf{J}_{\text{node}}$ **then**
- 36: $\mathcal{N}_{\Omega} \leftarrow \mathcal{N}_{\text{temp}}$
- 37: $\mathcal{N}_{\Omega}.\mathcal{N}_{\text{Parent}} \leftarrow \mathcal{N}_{\text{new}}$
- 38: **end if**
- 39: \triangleright **Determine whether \mathcal{N}_{new} is near goal $\mathcal{N}_{\text{goal}}$**
- 40: **if** $\text{InGoalRegion}(\mathcal{N}_{\text{goal}}, \mathcal{N}_{\text{new}}) == \text{true}$ **then**
- 41: $\Omega_{\text{goal}} \leftarrow \Omega_{\text{goal}} \cup \{\mathcal{N}_{\text{new}}\}$
- 42: **end if**
- 43: **end for**
- 44: \triangleright **Backtrack path from Ω_{goal} to $\mathcal{N}_{\text{start}}$.**
- 45: $S \leftarrow \text{GeneratePath}(\Omega_{\text{goal}}, \mathcal{T})$
- 46: **return** S

(1) **Trajectory smoothness:** This metric is quantified using the average magnitude of the vehicle's acceleration [29,60]. Lower average acceleration typically indicates smoother trajectories with reduced control aggressiveness, contributing to improved control feasibility and vehicle safety.

(2) **Pose stability:** This metric is evaluated using the mean absolute pose angle and the maximum absolute pose angle (in both pitch and roll orientations) [11,12]. Lower pose angles indicate greater stability, which enhances operational safety on rough terrain by reducing the risk of rollover and minimizing bumpiness.

Algorithm 3 Safety constraints generation

Input: Path S , Elevation map \mathbf{H} , Rectangle threshold ρ

Output: Safety constraints \mathbf{A}

- 1: \triangleright **Iterate from the start node.**
- 2: $\mathcal{N} \leftarrow S.\text{pop_front}()$
- 3: \triangleright **Expanding node \mathcal{N} gets the boundary $r = [r_x^-, r_x^+, r_y^-, r_y^+]^\top$.**
- 4: $r = \text{Inflate}(\mathcal{N}, \sigma, \rho)$
- 5: \triangleright **Insert r into \mathbf{A} .**
- 6: $\mathbf{A}.\text{push_back}(r)$
- 7: **while** $\mathcal{N} \in S$ **do**
- 8: \triangleright **Check relationship between current node and expansion boundary of previous two nodes.**
- 9: **if** $\text{Outside}(\mathcal{N}, \sigma, \mathbf{A}[-1])$ **then**
- 10: **if** $\text{Inside}(\mathcal{N}, \sigma, \mathbf{A}[-2])$ **then**
- 11: \triangleright **The \mathbf{A} deletes the boundary obtained from the previous node.**
- 12: $\mathbf{A}.\text{pop_back}()$
- 13: **else**
- 14: $r = \text{Inflate}(\mathcal{N}, \sigma, \rho)$
- 15: $\mathbf{A}.\text{push_back}(r)$
- 16: **end if**
- 17: **end if**
- 18: $\mathcal{N} \leftarrow S.\text{pop_front}()$
- 19: **end while**
- 20: **return** \mathbf{A}

Algorithm 4 VAF and TTA model train - Offline

Input: Original PointCloud map M , Neighborhood radius r , Maximum climbing gradient α , Motion trajectory dataset \mathbb{P} , Imu dataset \mathbb{I} .

Output: Elevation map \mathbf{H} , TTA model \mathcal{F} .

- 1: $M^s \leftarrow \text{ValidAreaFilter}(M, r, \alpha)$ [Algorithm 1]
- 2: \triangleright **PointCloud map converted to Elevation map.**
- 3: $\mathbf{H} \leftarrow \text{ElevationMapCons}(M^s)$
- 4: \triangleright **TTA model train.**
- 5: $\mathcal{F} \leftarrow \text{ModelTrain}(\mathbf{H}, \mathbb{P}, \mathbb{I})$ [Section 3.2.2]
- 6: **return** \mathbf{H}, \mathcal{F}

Algorithm 5 Motion planning - Online

Input: Elevation map \mathbf{H} , TTA model \mathcal{F} , Start $\mathcal{N}_{\text{start}}$, Goal $\mathcal{N}_{\text{goal}}$, Maximum Iterations k , Rectangular thresholds for safety constraints ρ .

Output: Motion trajectory $\mathbf{U} = [t, x, y]$.

- 1: $S \leftarrow \text{PatchRRT_Star}(\mathbf{H}, \mathcal{N}_{\text{start}}, \mathcal{N}_{\text{end}}, k)$ [Algorithm 2]
- 2: \triangleright **Safety constraints construction.**
- 3: $\mathbf{A} \leftarrow \text{SafetyConstraintsGen}(S, \mathbf{H}, \rho)$ [Algorithm 3]
- 4: \triangleright **Safety constraints construction.**
- 5: $\mathbf{U} \leftarrow \text{TrajectoryOptimization}(\mathbf{A}, \mathcal{F}, \mathbf{H})$ [Section 3.4.1]
- 6: **return** \mathbf{U}

(3) **Motion efficiency:** This metric is assessed based on trajectory length, traversal time, and average speed [15]. Shorter lengths, reduced traversal times, and higher average speeds indicate more efficient motion planning.

(4) **Computational cost:** This metric is measured by the total computation time required for trajectory generation, reflecting the method's real-time performance and computational efficiency in complex terrain environments [10].

These metrics collectively evaluate key aspects of motion planning quality, encompassing smoothness, safety, efficiency, and real-time performance. The corresponding quantitative results are detailed in

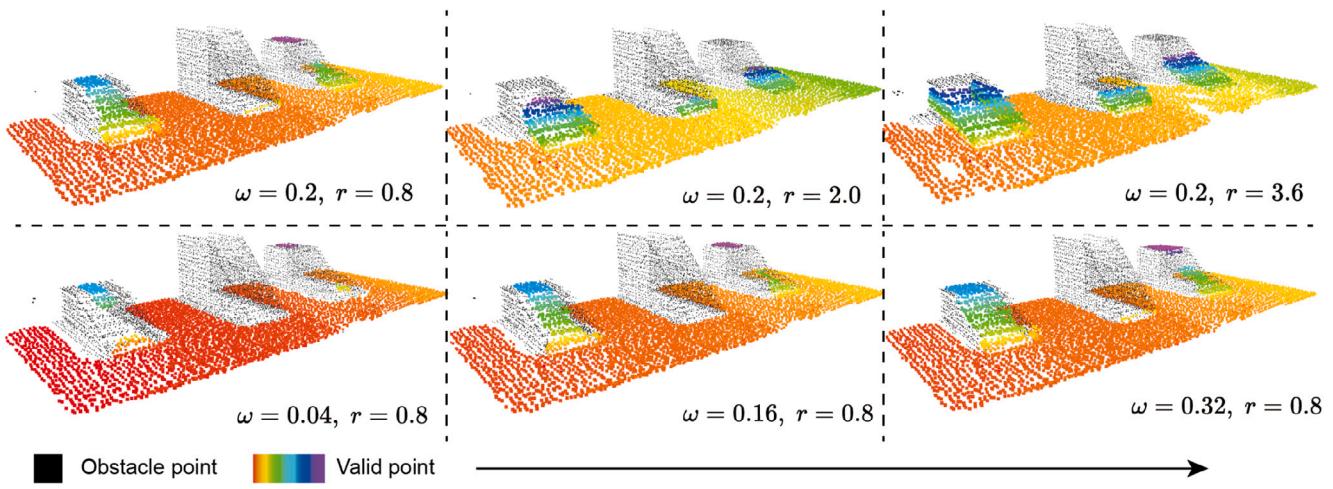


Fig. 6. Visualization of traversable areas extraction under different VAF parameter settings.

the experimental Section 4 and are utilized for comparative validation against existing methods.

4. Simulation and experiment verification

To validate the proposed motion planning method for complex environments with uneven terrain, both simulation and real-world experiments were conducted. Two scenarios, a construction site and a natural forest, were used for each validation to thoroughly evaluate the environmental applicability and reliability. Additionally, comparative experiments with state-of-the-art motion planning methods were performed, demonstrating that the proposed method achieves superior overall performance.

4.1. Effect of VAF parameters on traversable area extraction

To validate the effect of parameters in the VAF on the extraction of traversable areas, three typical slope scenarios were designed: a 20° gentle slope, a 50° steep slope, and a mixed slope consisting of a 20° front section and a 50° rear section. The maximum gradient climb of the vehicle was set to 40°. Point cloud data were collected from these scenarios, and traversable and non-traversable areas were manually labeled to serve as ground truth. We conducted two sets of experiments:

(1) With ω fixed at 0.2, r was varied from 0.8 to 3.6 in steps of 0.4, resulting in 8 parameter combinations.

(2) With r fixed at 0.8, ω was varied from 0.04 to 0.32 in steps of 0.04, resulting in another 8 parameter combinations.

The point clouds were processed using the VAF method with these parameter combinations. Representative results are visualized in Fig. 6, where black points indicate non-traversable areas and colored points indicate traversable areas.

For fixed r , a smaller ω leads to a conservative strategy for non-traversable area extraction. For example, when $\omega = 0.04$, even the 20° gentle slope was misidentified as non-traversable. As ω increases, the accuracy in identifying non-traversable areas improves.

To further quantitatively evaluate the impact of r and ω on the performance of VAF, the following two evaluation metrics are introduced: (1) Precision: the ratio of actual non-traversable points among those predicted as non-traversable; and (2) Recall: the ratio of actual non-traversable points that were correctly identified. The results are presented in Fig. 7.

The results indicate that a smaller neighborhood radius r achieves a comprehensive extraction of non-traversable areas while maintaining high accuracy, yielding a better balance between Precision and Recall. Therefore, setting r to the minimum radius of the vehicle's inscribed circle is practically feasible. On the other hand, a smaller ω yields higher Recall, covering more non-traversable areas, but at the cost of lower Precision and increased false alarms.

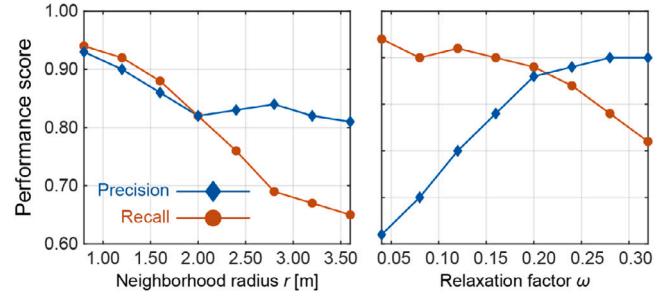


Fig. 7. Precision and recall scores under varying VAF parameters: (a) Neighborhood radius r ; (b) Relaxation factor ω .

4.2. Terrain traversable assessment model

The TTA model considers the coupling between terrain geometry and vehicle dynamics to achieve a more comprehensive terrain traversability assessment. The experiments will be centered on the following aspects to validate the TTA model:

1. Predictive performance of the TTA model on the training dataset.
2. The effect of the introduction of vehicle dynamics on the TTA model performance.
3. Effectiveness of TTA models trained on simulation data in real-world environments.

4.2.1. Training and validation of TTA model

This study evaluates the predictive performance of the TTA model, examining its effectiveness when trained on simulated data and deployed in real-world environments. All simulations were performed on a PC equipped with an Intel Core i5-10400F processor, an NVIDIA GeForce GTX 1050Ti GPU, and running the Ubuntu operating system. Blender modeled the terrain geometry, while Gazebo constructed the simulation environment. The TTA model was developed in PyTorch and implemented within Robot Operating System (ROS). Motion planning algorithms were deployed in ROS using C++, with vehicle control implemented via Model Predictive Control (MPC).

A representative 30 m × 30 m training terrain was procedurally generated using Blender's ANT Landscape plugin. Parameters were calibrated to mimic natural topographic characteristics. The geometric profiles of the terrain were generated using two independent Gaussian distributions. The first group is used to control the slope variation of the terrain with values obeying a Gaussian distribution of $N(0^\circ, 10^\circ)$, and

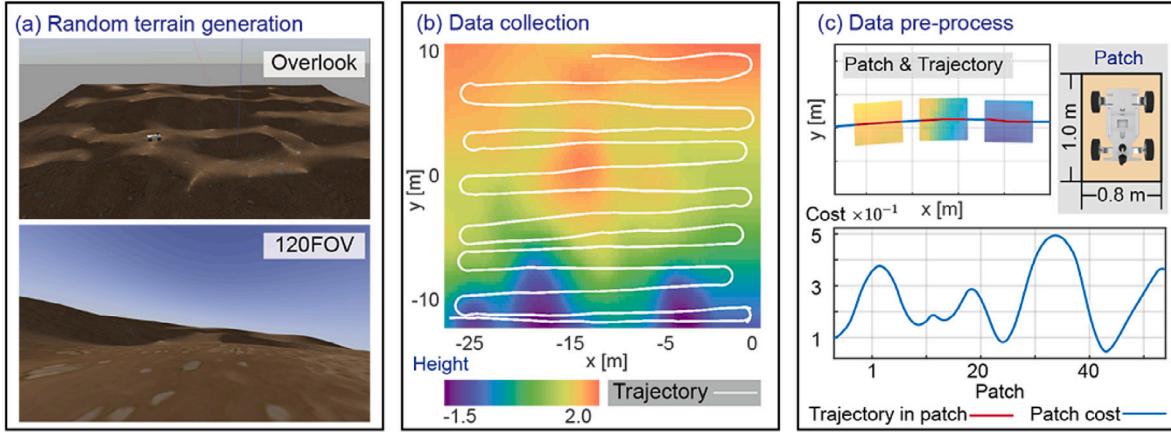


Fig. 8. Processing of Training Data for the TTA Model. (a) Simulation environment created using Blender and Gazebo; (b) Elevation map with vehicle trajectory and (c) elevation patch with vehicle pose data.

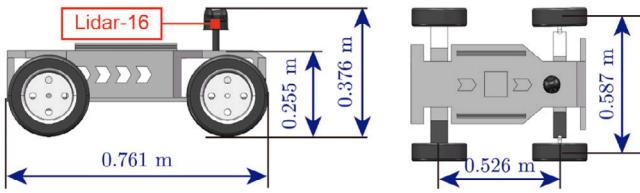


Fig. 9. Vehicle dimensions for simulation.

the maximum slope threshold is set to 0.6 rad to avoid extreme steep slopes that cannot be traversed by vehicles. The second group is used to control the elevation rise and fall of the terrain with values obeying a $N(0\text{m}, 0.5\text{m})$ Gaussian distribution and limiting the maximum local elevation difference to 1.0 m. The generated profile underwent smoothing to eliminate discontinuities, ensuring stable vehicle traversal. High-frequency sinusoidal perturbations (0.01-0.03 m amplitude) were superimposed to simulate microstructured terrain features like rubble. The resulting terrain model was imported into the Gazebo simulation platform.

The simulation environment created is shown in Fig. 8(a), where the terrain is designed to allow vehicle traversal. The simulated vehicle, shown in Fig. 9, has dimensions of $0.761\text{m} \times 0.587\text{m}$. A 16-line LiDAR captured point cloud data, while the 'p3d_base_controller' Gazebo plugin provided pose estimation. Vehicle motion was controlled manually, and point clouds along with pose data were collected, as shown in Fig. 8(b).

The A-LOAM is employed to generate a PointCloud map of the terrain, which is then mapped to an Elevation map. Based on the dimensions of the vehicle, the patch size is set to $1.0\text{m} \times 0.8\text{m}$. The training data obtained is shown in Fig. 8(c). A total of 23,000 training data were prepared and split into training, validation, and test sets in a 7:2:1 ratio for model training.

The structure and parameters of the TTA model are detailed in Section 3.2.2. The training parameters of epoch were set to 100, the Adam optimizer is used, the batch size is set to 128, and the rest of the hyperparameter were optimized using the Bayesian Optimization algorithm. The resulting values are presented in Table 4.

The validation results of the model on the test set are shown in Fig. 10. The evaluation metrics used include Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and the Coefficient of Determination (R^2). Detailed evaluation results are provided in Table 5. The model achieved an R^2 of 0.980 and an MAE of 0.028 rad, indicating high prediction accuracy on the simulation data test set.

4.2.2. Validation of the interaction with vehicle dynamics

Vehicle dynamics play a critical role in determining terrain traversability. The proposed TTA model effectively captures the intrinsic coupling between vehicle dynamics and terrain geometry, thereby generating a comprehensive terrain traversability score. To validate this capability, a random terrain with dimensions of $20\text{m} \times 5\text{m}$ was created using Blender. The terrain profile was constructed by applying Gaussian noise to create random slope variations, followed by a smoothing process to ensure physical plausibility. The maximum elevation difference was set to 0.4 m, with a maximum slope angle of 0.5 rad. The simulation environment was developed using the Gazebo platform, as shown in Fig. 11(a) and (b).

A 3D point cloud of the terrain was generated using A-LOAM and subsequently converted into an elevation map, as shown in Fig. 11(c). The vehicle was tasked with traversing from the start point (0 m, 0 m) to the goal point (30 m, 0 m) at constant linear speeds of 0.5 m/s, 1.0 m/s, 1.5 m/s, 2.0 m/s, and 3.0 m/s, respectively, to construct a validation dataset, as shown in Fig. 12(a). Results indicate a strong positive correlation between vehicle kinematic response and its speed under identical terrain conditions.

To assess model performance under varying dynamic conditions, the prediction accuracy of the TTA model was evaluated across different speed profiles. In addition, ablation experiments were conducted by removing the speed input channel to investigate the impact of vehicle dynamics on model performance. Comparative results are presented in Fig. 12, and the corresponding quantitative metrics are summarized in Table 6.

The No-speed network demonstrates satisfactory prediction performance at low speed ($v = 0.5\text{ m/s}$), achieving MAE of 3.677, RMSE of 4.574, and R^2 of 0.859. However, due to the absence of speed information, its predictive capability significantly deteriorates as the vehicle speed increases, indicating limited generalization across varying dynamic conditions. In contrast, the TTA model effectively captures the coupling relationship between terrain geometry and vehicle dynamic responses, thereby enabling robust pose prediction across a wide range of speeds. On average, the TTA model achieves a MAE of 1.813, RMSE of 2.272, and R^2 of 0.968, consistent across different motion states. Notably, even at high speeds ($v = 3.0\text{ m/s}$), the TTA model maintains high prediction accuracy, demonstrating strong adaptability to dynamic variations. These experimental findings underscore the advantage of incorporating coupled terrain-vehicle dynamics features in enhancing the generalization and prediction accuracy of learning-based models.

4.2.3. Validation of the TTA model for application in real-world environments

To further validate the effectiveness of the model trained on simulation data in real-world scenarios, experiments were conducted using

Table 4
Hyperparameter and layer configurations of TTA model.

No.	Layers	Filter hidden units	Kernel size	Stride/Rate	Padding	Activation
1	2D-Conv	128	3	1	2	–
2	Fully connected	64	–	–	–	–
	Dropout	–	–	0.3	–	–
3	2D-Conv	256	3	1	–	–
4	ConvLSTM	32	3	1	2	–
5	Fully connected	256	–	–	–	–
	Dropout	–	–	0.3	–	–

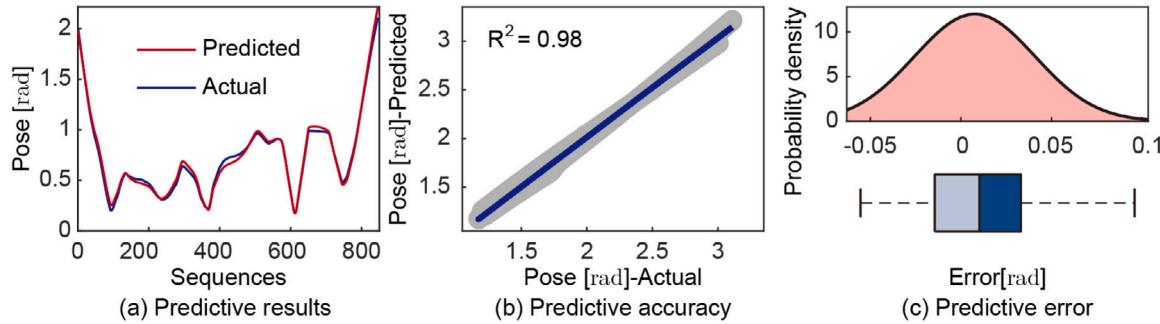


Fig. 10. Predictive Performance of TTA model. (a) The comparison between predicted and actual values; (b) prediction accuracy and (c) distribution of prediction error.

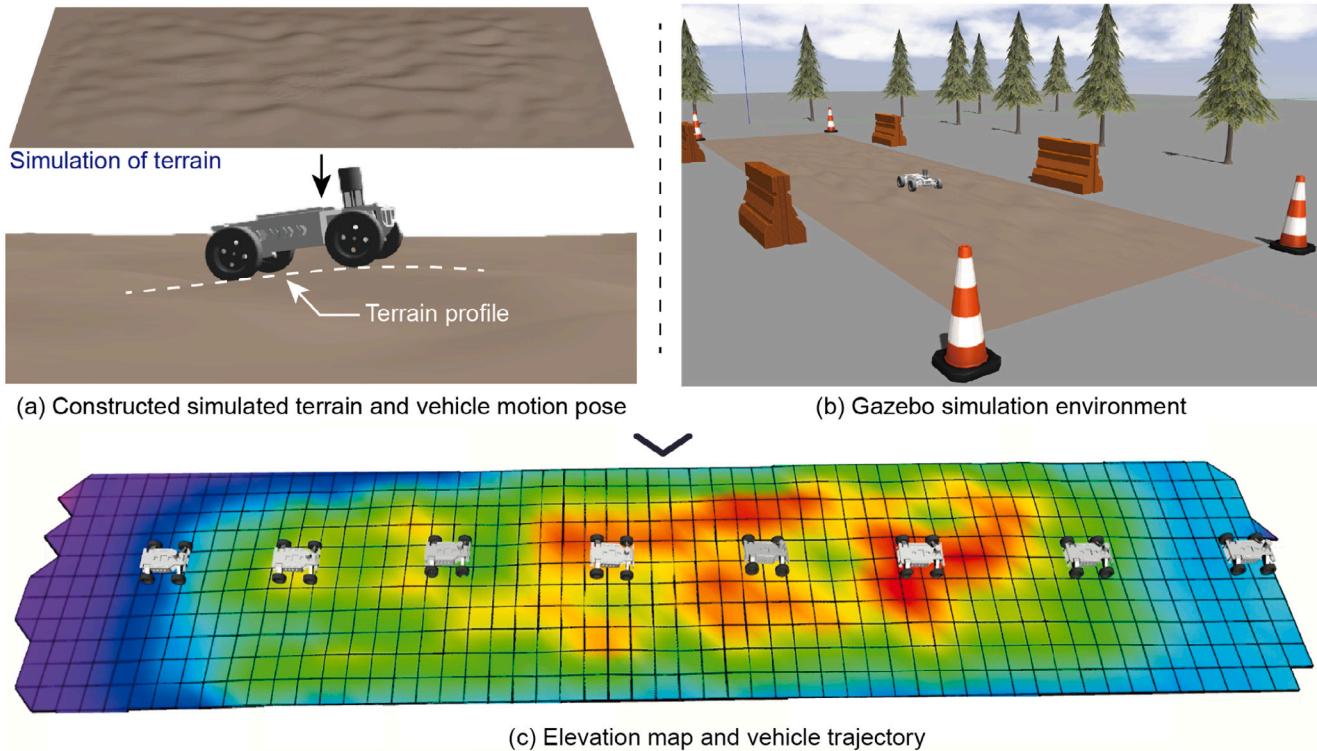


Fig. 11. Simulation environment for validating the interaction of the TTA model with vehicle dynamics.

Table 5
Model prediction accuracy in simulation data set.

Type	MAE	MAPE	RMSE	R ²
Value	0.028	1.679%	0.034	0.980

actual vehicles. The experimental vehicle is shown in Fig. 13(a), with detailed specifications provided in Table 7. The position and pose of the vehicle are measured by RTK/IMU sensors, with antennas mounted on the front and rear of vehicle. A Unitree LiDAR is employed to

Table 6
The quantitative validation of the interaction of TTA model with vehicle dynamics.

Type	No-speed networks			TTA model		
	MAE	RMSE	R ²	MAE	RMSE	R ²
0.5 m/s	3.677	4.574	0.859	0.805	1.167	0.976
1.0 m/s	4.430	5.389	0.696	1.558	2.005	0.966
1.5 m/s	5.475	6.869	0.620	1.798	2.222	0.960
2.0 m/s	8.292	10.846	0.533	2.405	2.986	0.965
3.0 m/s	11.243	15.591	0.396	2.499	2.980	0.972

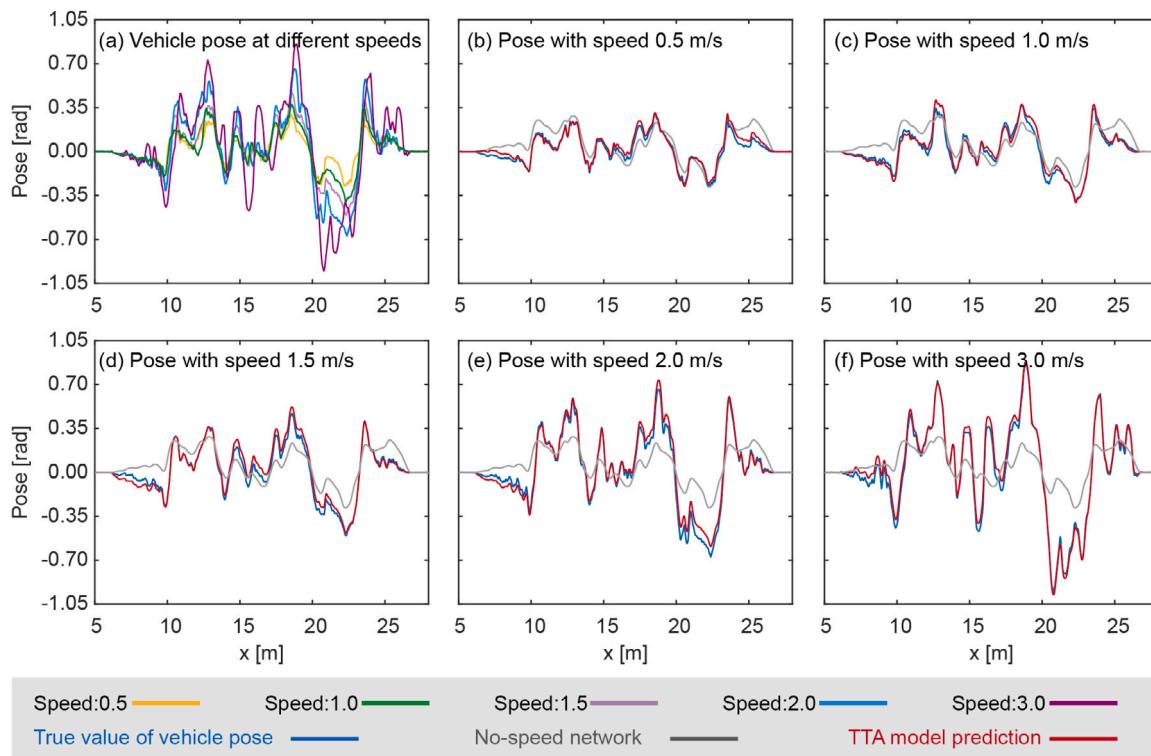


Fig. 12. The validation of the interaction of TTA model with vehicle dynamics.

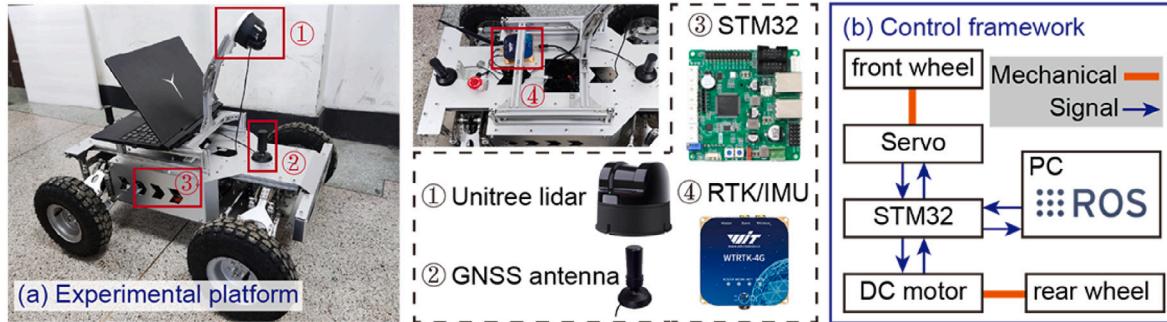


Fig. 13. Experimental platform.

Table 7
Experimental platform parameters.

No.	Parameters	Value	Unit
1	Length × Width × Height	0.761 × 0.587 × 0.376	m
2	Maximum speed	1.650	m/s
3	Weight of vehicles	23.000	kg
4	Tire diameter	0.180	m
5	Minimum turning radius	1.020	m
6	Maximum climbing gradient	0.800	rad

gather environmental point clouds data to construct the PointCloud map. The drive system of vehicle utilizes a rear-wheel drive and front-wheel steering configuration, as illustrated in Fig. 13(b). An STM32 microcontroller is used as the core controller, operating the drive motors and steering servos via a CAN bus, which also communicated between the STM32 and the PC.

The data collection environments are illustrated in Fig. 14. Three distinct environments were selected to represent varied terrain types: Environment 1, with scattered construction materials; Environment 2,

Table 8
Model prediction accuracy in real-world environments.

Type	MAE	MAPE	RMSE	R ²
Environment 1	0.026	8.976%	0.030	0.957
Environment 2	0.021	9.174%	0.026	0.942
Environment 3	0.023	3.404%	0.029	0.945

featuring sloped earthen materials and Environment 3, a natural forest. A total of 5475 data sets were collected.

The results of the terrain traversability assessment in real-world environments using the model trained on simulation data are shown in Fig. 15. The model performance is evaluated with MAE, MAPE, RMSE, and R², as summarized in Table 8. These results indicate that the TTA model trained on simulation data can be effectively applied in real-world environments. In the three experimental environments, R² values reached 0.957, 0.942, and 0.945, respectively, demonstrating high predictive accuracy.



Fig. 14. Validation data collection for TTA model in three real-world environments.

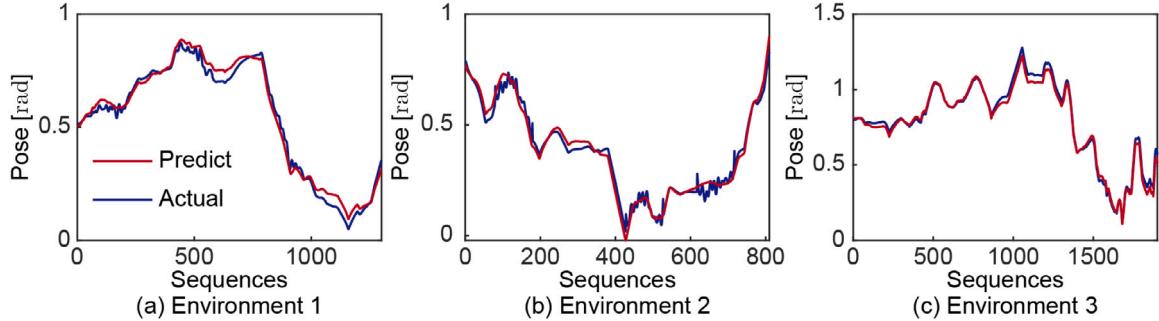


Fig. 15. Prediction accuracy of models in real-world environments.



Fig. 16. Simulated construction site environment. (a) Top view of the entire scene, showing a sloped material pile, scattered stones, guardrails, and construction equipment; (b) field of view (FOV) representation of the environment; (c) Elevation map of extracted traversable areas.

4.3. Simulation

The simulation primarily addresses the following questions:

1. **Environmental adaptability.** A construction site scenario with uneven terrain is created.
2. **Reliability.** A natural forest—were environment with highly rugged terrain and complex, scattered obstacles is created.
3. **Advancement.** A comparative analysis is conducted with current advanced trajectory planning methods for complex environment with uneven terrain.

4.3.1. Case 1

A simulated construction site environment, illustrated in Fig. 16, features uneven terrain, construction machinery, and material piles with inclined gradients. A 16-line LiDAR is employed to generate a PointCloud map, while the VAF is used to delineate effective traversable areas. Based on the size and dynamics of the vehicle, set the neighborhood radius of the VAF to $r = 0.8$ m, the maximum climbing gradient of the vehicle to $\alpha = 0.8$ rad, and the relaxation factor to $\omega = 0.16$. Subsequently, the Elevation map H of traversable areas is extracted and constructed with a spatial resolution of $0.2\text{ m} \times 0.2\text{ m}$, as shown in Fig. 16(c).

The starting point is set at $(-5.917\text{ m}, 12.728\text{ m})$, and the goal point at $(17.764\text{ m}, 4.944\text{ m})$. Patch-RRT* is employed for path planning within

the Elevation map, with a maximum iteration count of $k = 1500$. Sampling targets the goal point with a probability of 0.2, and the step length for tree growth is 0.4 m. After tuning, the weights are set as follows: terrain traversability cost $\lambda_t = 0.3$, goal distance cost $\lambda_g = 0.3$, start distance cost $\lambda_{\text{star}} = 0.3$, and state cost $\lambda_{\text{state}} = 0.1$. After obtaining the path searched by Patch-RRT*, the safety constraints are constructed with rectangular thresholds $\rho = 1.0\text{ m}$, as shown in Fig. 17(a). Patch-RRT* demonstrates high efficiency, achieving pathfinding from the starting point to the goal in 10.284 ms. The constraints on the dynamics of the vehicle are $Vel_{\max} = 1.6\text{ m/s}$, $Acc_{\max} = 1.5\text{ m/s}^2$. With these constraints, the segment Bézier trajectories are constructed and optimized. The optimized trajectories are shown in Fig. 17(b) (c). And Table 9 provides the parameters used in the planning process. This motion planning method exhibits high computational efficiency, producing optimized trajectories in 2.431 ms.

MPC is employed to control the vehicle tracking trajectory. The parameters of the vehicle are shown in Fig. 18, and quantitative evaluation metrics are detailed in Table 10. Within the simulated construction site, the vehicle achieves stable high-speed operation, reaching a peak speed of 1.6 m/s in localized areas and maintaining an average speed of 1.169 m/s. The vehicle experiences a maximum pitch angle of 19.058° and a maximum roll angle of 17.350°, contributing to a smooth motion. Experimental results indicate that the proposed motion planning method effectively manages terrain variations and adaptation

Table 9

Parameters of the motion planning process (Simulation case 1).

Type	Time cost of Patch-RRT* (ms)	Number of rectangles in safety constraints	Time cost of optimization (ms)	Trajectory length (m)
Value	10.284	12	2.431	29.050

Table 10

Motion parameters of the vehicle (Simulation case 1).

Type	Trajectory time (s)	Average speed (m/s)	Average acceleration (m/s^2)	Maximum pitch (deg)	Maximum roll (deg)	Average pitch (deg)	Average roll (deg)
Value	24.035	1.169	0.381	19.058	17.350	7.636	4.664

Table 11

Parameters of the motion planning process (Simulation Case 2).

Type	Time cost of Patch-RRT* (ms)	Number of rectangles in safety constraints	Computational cost of optimization (ms)	Trajectory length (m)
Value	12.433	15	2.629	31.465

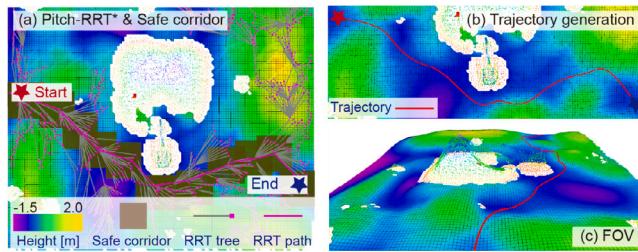


Fig. 17. Motion planning results (Simulation Case 1). (a) Patch-RRT* search process and the generated safety constraints; (b) the final trajectory obtained from optimization; (c) the FOV view of the trajectory.

to complex construction site environments. The method provides safe, reliable, and efficient motion trajectories to guarantee the smooth operation of the vehicle.

4.3.2. Case 2

A simulated natural forest environment, as illustrated in Fig. 16, features uneven terrain, complex and scattered obstacles (Trees, shrubs and stones). Complex forest environments with narrow and dispersed areas that can be safely traversed by vehicles pose significant challenges for motion planning methods. A PointCloud map of the environment is generated using 16-line LiDAR, and the traversable areas of the vehicle is extracted using the VAF with the same parameter settings as Case 1 (Section 4.3.1). The generated Elevation map is presented in Fig. 19(c). Regions with steep slopes that do not meet vehicle traversability requirements are classified as obstacles by the VAF, requiring the motion planning method to avoid these areas.

The starting point of vehicle are set at (4.827 m, -0.030 m), with the goal point at (-9.071 m, 26.266 m). Patch-RRT* uses the same parameter settings as in Case 1 (Section 4.3.1). The result of searched paths and the constructed safety constraints are shown in Fig. 20(a). By assigning a specific probability for direct sampling toward the goal point, the RRT tree converges more quickly and continues iterating to improve path quality. The safety constraints identifies traversable areas within the complex distribution of obstacles. The optimized trajectory is illustrated in Fig. 20(b) (c), with planning parameters provided in Table 11. In this challenging environment, Patch-RRT* successfully identified a path in 12.433 ms, while trajectory optimization is completed in 2.629 ms, demonstrating high efficiency in motion planning. Additionally, this motion planning method reliably finds narrow traversable areas amid scattered obstacles and optimizes a smooth, feasible trajectory along gentle terrain, verifying the realizability of the method.

4.3.3. Comparative verification

To validate the effectiveness of the proposed motion planning method, comparative experiments were conducted against other advanced motion planning methods for complex environments with uneven terrain. The primary methods for comparison include the following:

- (1) Reactive-based motion planning methods, such as RSPMP [43].
- (2) Graph search-based motion planning methods, such as T-Hybrid A* [55].
- (3) Sampling-based motion planning methods, such as PUTN-RRT* [15].
- (4) Optimization-based motion planning methods, such as A*-RRT-LTR [63].

In the established construction site scene, given the same start point (0.583 m, 1.737 m) and goal point (8.331 m, 21.832 m). The motion trajectories are generated using each of the motion planning methods described above. For each method, five motion trajectories are produced, and the optimal trajectory from each set is selected for comparison. Fig. 21 presents the resulting motion trajectories, along with analyses of trajectory speed, acceleration, vehicle absolute pose (indicated by $\|\theta\| + \|\gamma\|$), the distribution of absolute pose value and the distribution of the lengths of the five trajectories produced by each method. The quantitative evaluation results for each method are provided in Table 12, where the blue bolded font indicates that optimal performance is achieved.

The proposed motion planning method demonstrates optimal performance with high computational efficiency, requiring only 14.588 ms to compute the optimized trajectory. This trajectory achieves a shorter path length of 26.310 m, enabling the vehicle to reach the goal in just 11.943 s, which is significantly more efficient than other methods. In contrast, the reactive-based method lacks global path guidance, resulting in a longer travel time of 23.234 s, as its planning and control operate concurrently while motion of the vehicle. During vehicle motion, the trajectory generated by the proposed method allows for a higher average speed of 1.033 m/s, with an average acceleration of $0.204 m/s^2$ and a maximum absolute pose angle of 8.986° , providing a smoother and more stable travel experience than other methods. Overall, the proposed motion planning method for complex environment with uneven terrain excels in computational efficiency, operational effectiveness, and travel stability, reflecting the advancement.

4.4. Experiment

The vehicle used for the real-world environment experiment is shown in Fig. 13(a). The experiment addresses the following critical questions:

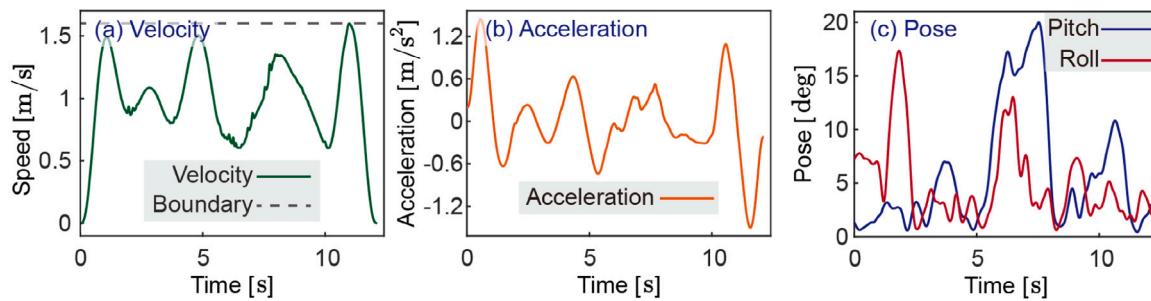


Fig. 18. Motion process of the vehicle (Simulation case 1).

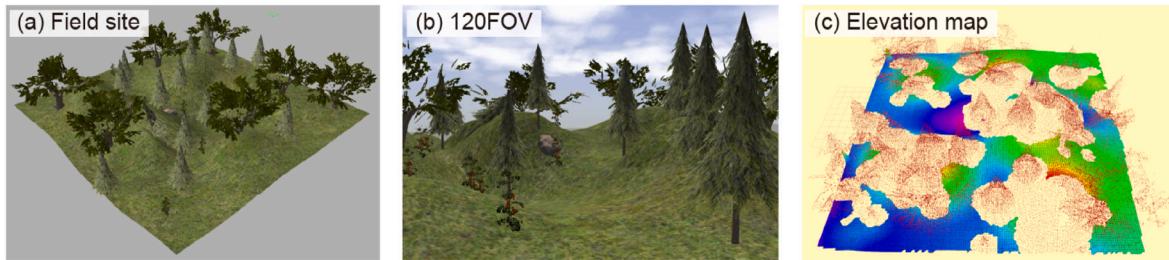


Fig. 19. Simulated natural forest environment. (a) Top view of the entire environment, including complex scattered trees, stones, and shrubs; (b) FOV view of the environment; and (c) the Elevation map of traversable areas.

Table 12
Performance comparison of different motion planning methods (Simulation).

Type	Trajectory time (s)	Trajectory length (m)	Average speed (m/s)	Average acceleration (m/s^2)	Maximum absolute pose (deg)	Average absolute pose (deg)	Computational cost (ms)
RSPMP	23.234	35.071	0.898	0.474	32.809	11.630	—
T-Hybrid A*	17.415	34.894	1.003	0.484	21.420	13.977	28.967
PUTN-RRT*	18.046	30.666	0.845	0.415	28.413	12.465	30.564
A*-RRT-LTR	14.810	32.911	0.872	0.363	29.889	12.811	25.436
Proposed	11.943	26.310	1.033	0.204	8.986	10.519	14.588

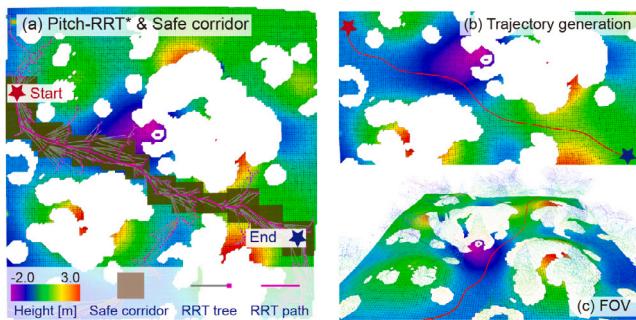


Fig. 20. Results of motion planning (Simulation case 2). (a) The search process of Patch-RRT* and the generated safety constraints; (b) the optimized trajectory; (c) the FOV view of the trajectory.

(1) Environmental applicability of the method in a real-world environment. The proposed motion planning method is deployed in a real-world construction site environment to generate a motion trajectory from the starting point to the goal point, which realizes the smooth motion of the vehicle.

(2) Reliability of the method in a real-world environment. The proposed motion planning method is deployed in a real-world natural forest environment. The method copes well with environments with complex and scattered obstacles and generates a safe and smooth motion trajectory from the starting point to the goal point.

(3) Advancement of the method in a real-world environment. Adequate comparative experiments have been carried out in real-world environments.

4.4.1. Case 1

A construction site at the Nanling campus of Jilin University is selected for the experiments, where piled construction materials and earth, along with the uneven terrain, as illustrated in Fig. 22(a). Using the A-LOAM, a PointCloud map of the environment is constructed, and dense point cloud data were downsampled through voxel filtering [77]. The traversable areas of vehicle is extracted using the VAF. Based on the performance of vehicle, detailed in Table 7, the neighborhood radius of VAF is set to $r = 0.80$ m, and the maximum climbing gradient of the vehicle to $\alpha = 0.80$ rad. To address noise in the original PointCloud map, a relaxation factor is set to $\omega = 0.20$. Subsequently, the Elevation map of the environment is constructed with the grid resolution of $0.25\text{ m} \times 0.25\text{ m}$, as shown in Fig. 22(b).

With the starting point set at $(1.447\text{ m}, -1.878\text{ m})$ and goal point at $(41.669\text{ m}, 2.698\text{ m})$, the maximum iteration for Patch-RRT* is defined as $k = 2000$, with a 0.2 probability of sampling directly at the goal and step length for the tree growth is set at 0.4 m. The terrain traversability cost weight is set to $\lambda_t = 0.4$, the goal distance cost weight to $\lambda_g = 0.3$, the start distance cost weight to $\lambda_{\text{start}} = 0.2$, and the state cost weight to $\lambda_{\text{state}} = 0.1$. After Patch-RRT* searched feasible paths, safety constraints are generated, as shown in Fig. 22(c). Experimental results demonstrate that Patch-RRT* quickly search paths from the starting point to the goal in the real-world environment and continuously optimizes the path with each iteration. Subsequently, segment Bézier trajectories

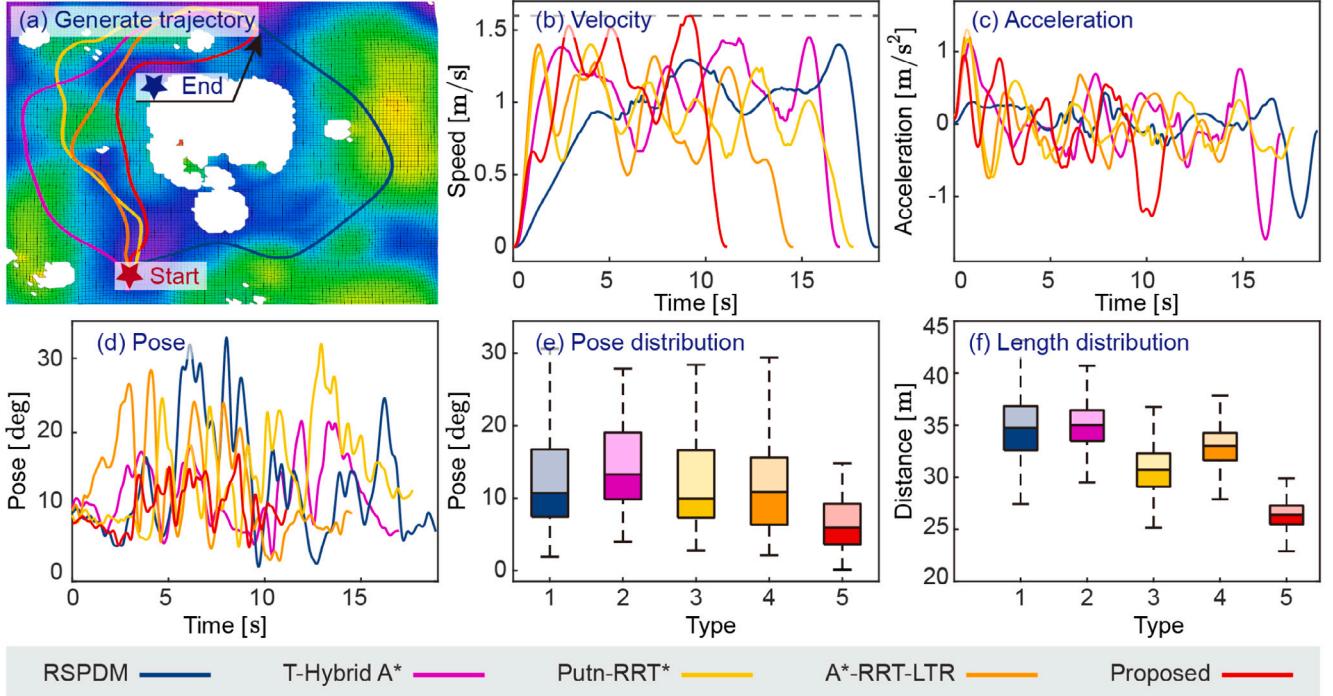


Fig. 21. Comparative validation of motion planning methods (Simulation).

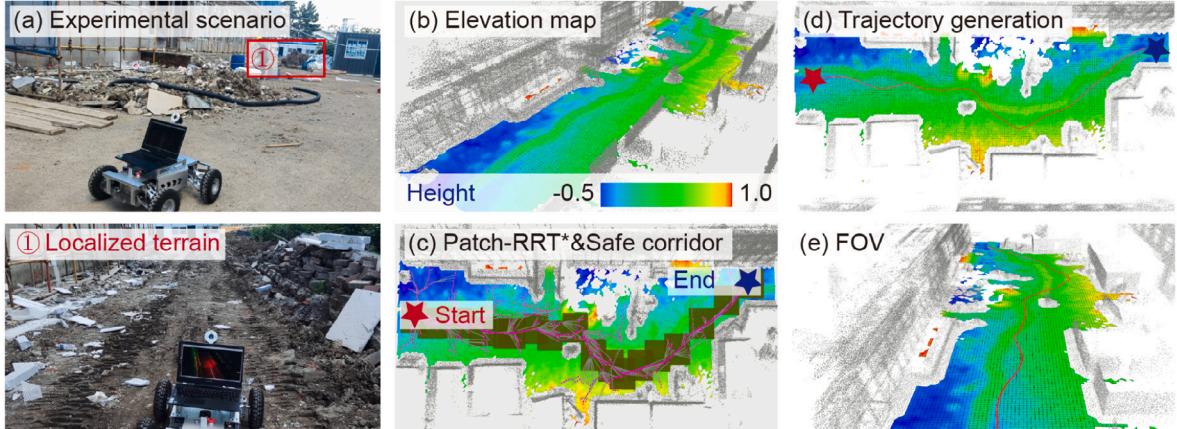


Fig. 22. Real-world construction site experiment. (a) the general view of the environment; (b) the Elevation map of extracted traversable areas; (c) the Patch-RRT* sampling process and the generated safety constraints, (d) the motion trajectory after optimization, and (e) the FOV view of the trajectory.

Table 13
Parameters of the motion planning process (Experiment case 1).

Type	Time cost of Patch-RRT* (ms)	Number of rectangles in safety constraints	Time cost of optimization (ms)	Trajectory length (m)
Value	18.370	17	5.610	44.266

are constructed and optimized under vehicle dynamics constraints, with maximum speed $V_{cc\max} = 1.6 \text{ m/s}$ and acceleration $Acc_{\max} = 1.5 \text{ m/s}^2$. The optimized trajectory is depicted in Fig. 22(d), with quantitative metrics provided in Table 13. Results indicate that, within the real-world construction site environment, the generated trajectory is smooth, navigates through the traversable areas, and maintains a safe distance from obstacles, thereby ensuring the safe and smooth operation of the vehicle.

The MPC controller is implemented to track the desired trajectory of the vehicle. The speed, acceleration, pitch angle θ and roll angle γ , as illustrated in Fig. 23 and detailed in Table 14. Compared to the terrain at real-world construction site is more uneven.

The vehicle maintained relatively low speeds, with an average speed of 0.821 m/s , and an average acceleration of 0.457 m/s^2 . Due to the rough terrain, changes in vehicle pose occurred frequently, although these changes were of low magnitude; the maximum observed pitch and roll angles were 8.823° and 6.137° , respectively. Experimental results demonstrate that the proposed motion planning method demonstrates strong adaptability to real-world construction site environments.

4.4.2. Case 2

A natural forest located in Nanling campus of Jilin University is selected, which is characterized by uneven terrain, large ground height

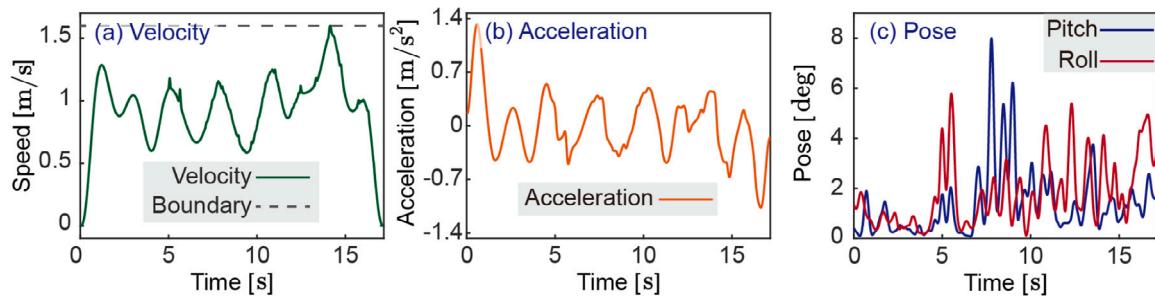


Fig. 23. Motion process of the vehicle (Experiment case 1).

Table 14
Motion parameters of the vehicle (Experiment case 1).

Type	Trajectory time (s)	Average speed (m/s)	Average acceleration (m/s^2)	Maximum pitch (deg)	Maximum roll (deg)	Average pitch (deg)	Average roll (deg)
Value	17.413	0.821	0.457	8.823	6.137	3.268	3.615

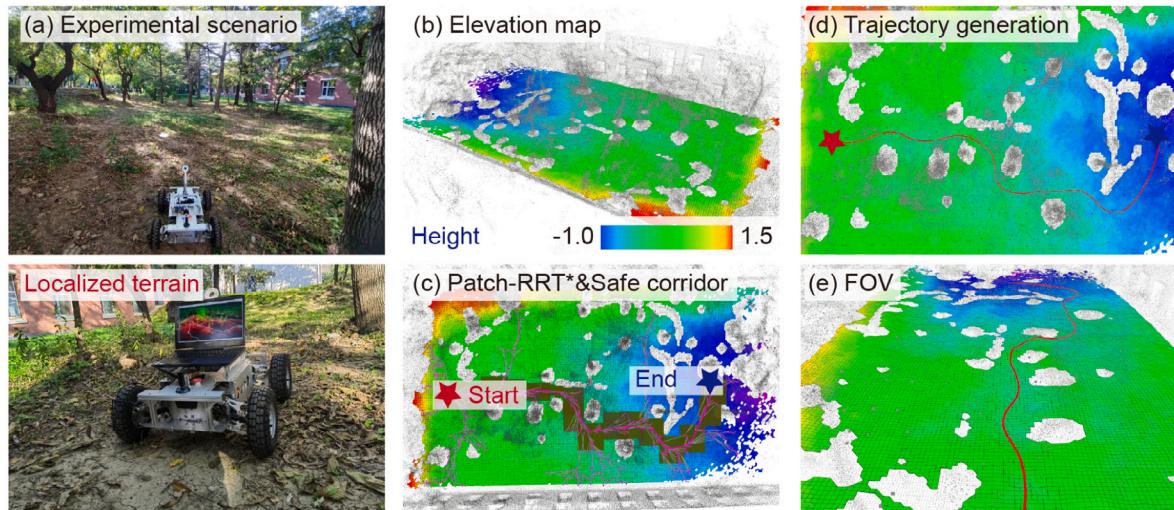


Fig. 24. Real-world natural forest experiment. (a) the general view of the environment; (b) the Elevation map of traversable areas; (c) the Patch-RRT* sampling process and the generated safety constraints; (d) the motion trajectory generated after optimization; (e) the FOV view of the trajectory.

undulations, and the presence of a large number of scattered obstacles (e.g., shrubs, trees), as shown in Fig. 24. A-LOAM is used to construct a PointCloud map, and the collected original PointCloud map is down-sampled by voxel filtering, and then the vehicle traversable area is extracted using VAF. The parameters of VAF are the same as Case 1 (Section 4.4.1). Construct the Elevation map and set the grid with a resolution of $0.25\text{ m} \times 0.25\text{ m}$, and the results are shown in Fig. 24(b).

The generated Elevation map contains numerous scattered obstacles. The starting point for the motion of vehicle are set to $(-2.034\text{ m}, 29.939\text{ m})$, with the goal point at $(-6.504\text{ m}, -6.398\text{ m})$. Patch-RRT* parameters remain consistent with Case 1 (Section 4.4.1), and the resulting paths and safety constraints are illustrated in Fig. 22(c). The results indicate that the safety constraints can identify regions suitable for safe traversal within the complex environment containing dispersed obstacles. The planning parameters are detailed in Table 15, and the optimized trajectory is shown in Fig. 20(d). In this complex environment with uneven terrain, the safety constraints includes a higher number of rectangles, which increases the trajectory optimization time but remains within acceptable bounds. Experimental results confirm that the proposed motion planning method effectively handles scattered obstacles, enabling the planning of feasible, smooth trajectories.

The speed, acceleration, pitch angle γ and roll angle θ of the vehicle during motion are shown in Fig. 25 and Table 16. Compared to

the construction site, the natural forest environment presents greater terrain unevenness and scattered obstacles. The average speed of the vehicle during the movement is 0.973 m/s , which shows a high traversal efficiency. High accelerations were observed at the start and goal points, with a peak acceleration of 1.213 m/s^2 and an average absolute acceleration of 0.453 m/s^2 , resulting in a smoother travel experience. Due to the uneven terrain, vehicle pose changes occurred frequently, though with limited magnitude, as indicated by maximum pitch and roll angles of 9.486° and 9.413° , respectively, reflecting a relatively stable vehicle state. Experimental results demonstrate that the proposed motion planning method effectively adapts to complex environments and well realizability.

4.4.3. Comparative verification

In the experiments above, the potential of the method for real-world applications is validated in two typical unstructured work environments: construction sites and natural forests. To further assess the advancement of method in real-world applications, the forest environment in Section 4.4.2 is compared with several advanced motion planning methods discussed in Section 4.3.3. This comparative experiment uses the starting point, $(-3.673\text{ m}, 11.274\text{ m})$, and goal point $(15.347\text{ m}, 11.070\text{ m})$, to generate the motion trajectory of vehicle, with results presented in Fig. 26.

Table 15
Parameters of the motion planning process (Experiment case 2).

Type	Time cost of Patch-RRT* (ms)	Number of rectangles in safety constraints	Time cost of optimization (ms)	Trajectory length
Value	17.330	20	2.908	39.440

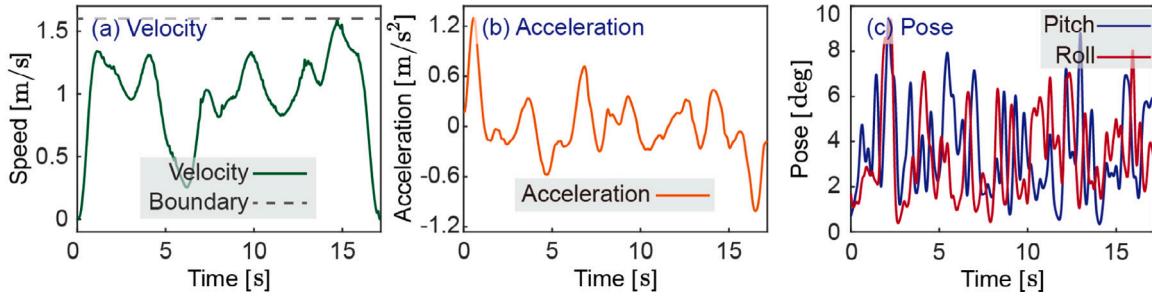


Fig. 25. Motion process of the vehicle (Experiment case 2).

Table 16
Motion parameters of the vehicle (Experiment case 2).

Type	Trajectory time (s)	Average speed (m/s)	Average acceleration (m/s ²)	Maximum pitch (deg)	Maximum roll (deg)	Average pitch (deg)	Average roll (deg)
Value	17.320	0.973	0.453	9.486	9.413	3.652	3.394

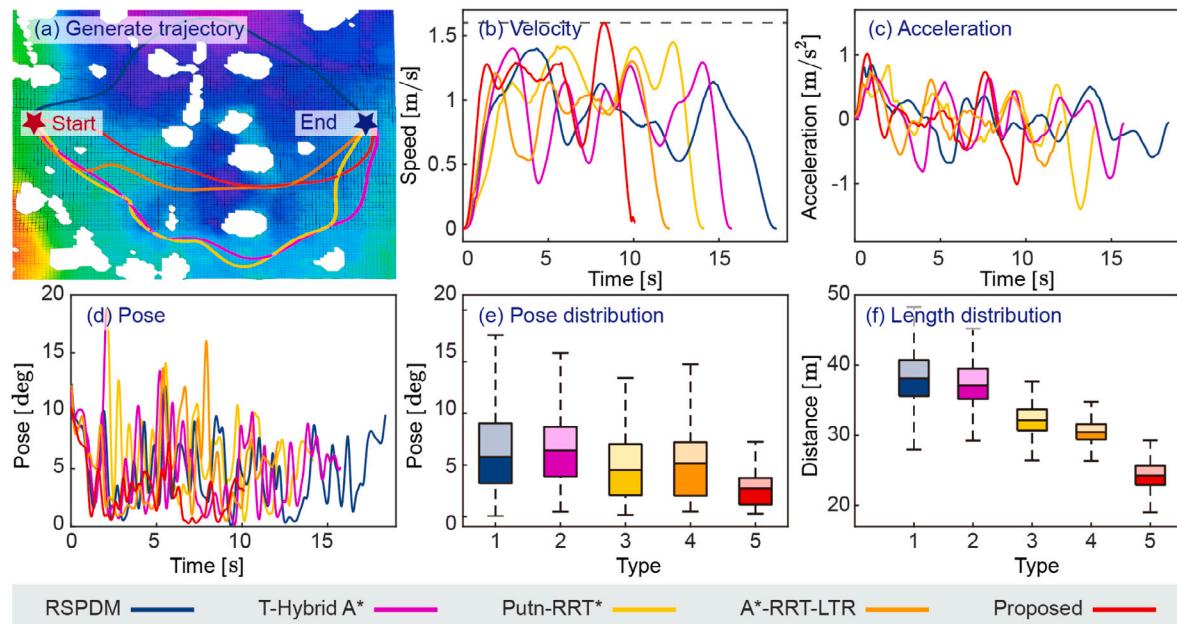


Fig. 26. Comparative validation of motion planning methods (Experiment).

The Table 17 provides quantitative evaluation results for each method, indicating that the proposed motion planning method offers optimal overall performance. The blue bolded font indicates that optimal performance is achieved. And Fig. 26 presents the trajectory, speed, acceleration, absolute pose (indicated as $\|\theta\| + \|\gamma\|$), absolute pose distribution, and trajectory length distribution generated by different methods. In the real-world environment, the proposed method demonstrates high computational efficiency, completing path search and trajectory optimization within 25.634 ms. The generated trajectory is efficient, with a length of 24.248 m, allowing the vehicle to reach the goal point from the starting point in 10.028 s, that outperforming other

methods in operational efficiency. Additionally, the proposed method produces a trajectory with a high average speed of 1.023 m/s and an average acceleration of 0.274 m/s², while maintaining a maximum absolute pose angle of 10.430°, indicating a smoother motion than other methods. In summary, the proposed motion planning method shows superior performance in computational efficiency, operational effectiveness, and vehicle pose smoothness in real-world environments, representing advancement.

In summary, the proposed VAF effectively identifies vehicle traversable areas from the original PointCloud map. The TTA model combined vehicle dynamic and terrain features, trained on simulation

Table 17
Performance comparison of different motion planning methods (Experiment).

Type	Trajectory time (s)	Trajectory length (m)	Average speed (m/s)	Average acceleration (m/s^2)	Maximum absolute pose (deg)	Average absolute pose (deg)	Time cost (ms)
RSPMP	18.876	37.091	0.983	0.345	13.517	6.593	–
T-Hybrid A*	17.042	35.762	0.825	0.392	19.462	7.188	32.487
PUTN-RRT*	14.012	32.367	0.851	0.374	14.484	6.7978	35.947
A*-RRT-LTR	12.987	30.635	0.834	0.405	15.028	5.8304	30.273
Proposed	10.028	24.248	1.023	0.274	10.430	4.029	25.634

data, generalizes well to real-world environments, providing a comprehensive terrain evaluation. The proposed vehicle motion planning method for complex environment with uneven terrain has been validated through multi-scenario experiments in both simulated and real-world experiment. The method demonstrates strong adaptability and generalizability across diverse, complex environments, effectively managing complex and dispersed obstacles. Compared to other advanced methods, the proposed approach offers a substantial improvement in performance.

5. Conclusion

Achieving feasible, safe and efficient motion planning for vehicles in complex environment with uneven terrain is important for advancing automation in construction. However, the uneven terrain poses a number of challenges: the difficult assessment of terrain traversability, large scale cost maps lead to time-consuming computations, and the complexity of highly coupling vehicle dynamics with terrain features.

This study proposed a constraint-aware motion planning methods with integrated vehicle-terrain traversability assessment for feasible, safe and efficient vehicle motion in construction scenario. The main contributions of this study include the following:

(1) The coupling relationship between the vehicles dynamic and the terrain features is considered comprehensively. The VAF is constructed to realize the effective identified of traversable areas under complex environment with uneven terrain. The TTA model that can output continuous terrain traversability scores is also established. The effectiveness of TTA model trained using simulation data in real environments is verified.

(2) The path search algorithm Patch-RRT* is constructed for on-demand terrain evaluation, which efficient traversable path search using 2D manifolds embedded in 3D space without explicit topology extraction is achieved in complex environment with uneven terrain.

(3) A constraint-aware trajectory optimization method that incorporates terrain traversability is proposed, using a custom objective function to consider both terrain features and vehicle dynamics. Motion trajectories can be generated that meet safety, continuity, and dynamic feasibility constraints.

(4) The method is adequately simulated and experimentally validated in multiple scenarios. The environmental adaptability, robustness, and advancement of the method are verified.

The method has been fully validated in simulations and experiments, proving its effectiveness in complex environment with uneven terrain and dispersed obstacle. It can efficiently identify the traversable areas of vehicles in complex environments, and quickly search out the path from the starting point to the goal point, which is then inflated to obtain the safety constraints. Optimization with constraints on a custom objective function enables the generation of safe, feasible and efficient vehicle motion trajectories. The method demonstrates high environmental adaptability and realizability in multi-scenario simulations and real-world experiments. Compared to other advanced motion planning methods, the proposed method is computationally efficient and is able to generate motion trajectories with more stable travel as well as faster travel speeds. The method offers a promising solution for autonomous motion planning of wheeled vehicles in construction scenarios.

6. Applicability and future work

Although this research focuses on wheeled vehicles operating in construction scenarios, the proposed frameworks, such as the TTA model, Patch-RRT* path search, and constraint-aware trajectory optimization, are inherently modular and extensible.

The TTA model uses IMU data from the body to assess terrain traversability, which is very effective for vehicles with rigid/quasi-rigid connections between the chassis and ground contact points, such as tracked vehicles, articulated vehicles, etc. However, for legged robots, which are vehicles with indirect contact between the body and terrain, the flexibility of the legs results in significant decoupling between the terrain and the body. This results in a body pose that does not accurately reflect local terrain geometry, making it difficult to apply the TTA model directly for judgment. In addition, the TTA model focuses on terrain traversability assessment in hard-surface environments, and is particularly applicable to complex terrain containing uneven structures (e.g., rocks, slopes, loose gravel, etc.). In these scenarios, there is a clear contact interface between the vehicle and the terrain, and the body pose can truly reflect the terrain undulation features. However, for soft terrain such as mud and sand, the deformation features can significantly affect the wheel-ground contact mechanism, making the IMU pose response no longer reliable.

The Patch-RRT* and constraint-aware trajectory optimization methods rely on the results of terrain traversability assessment and construct the constraints into a generic mathematical form. This feature makes it independent of a specific vehicle structure and can be effectively applied to wheeled, tracked and other vehicle chassis with well-defined path constraints. In addition, Patch-RRT* performs effective path search based on discrete terrain assessment costs, rather than the geometric structure of the terrain itself. Therefore, by quantifying the cost of terrain traversability, including terrain stiffness, slope, and rolling resistance, it can be applied to more complex driving environments. The applicability of this study and what can be expanded in the future is shown in the Table 18.

The proposed motion planning method still has limitations, primarily including:

(1) The dynamic obstacles in complex environments is ignored, such as construction workers. The validates the feasibility of the method over a large environmental range. Avoiding dynamic obstacles would require further consideration of dynamic real-time map construction and local re-planning methods.

(2) The method ignores the effect of unknown areas of the map on planning. For environments without prior knowledge, it is crucial to establish methods for dynamic adjustment of maps as well as effective strategies for dealing with unknown areas.

Future research will further develop the proposed motion planning method to accommodate a more generalized operational environment, primarily including:

(1) Exploring terrain traversability assessment models for multi-configuration chassis in combination with whole-body kinematic mechanisms.

(2) Fusion of multi-sensor signals to explore traversability assessment models for loose, muddy, and other complex terrain.

Table 18
Applicability and potential extensions.

Method component	Current applicability	Future expandable study
TTA model	<ul style="list-style-type: none"> Suitable for vehicles with rigid connection between chassis and wheel system. Applicable to rigid and compact terrain. 	<ul style="list-style-type: none"> Applicable to articulated or non-rigid chassis (e.g., legged robots). Loose soil, muddy surfaces, and highly deformable terrains.
Patch-RRT* and constraint-aware trajectory optimization	<ul style="list-style-type: none"> Suitable for various vehicle structures, due to its reliance on discrete traversability cost maps. 	<ul style="list-style-type: none"> Integration with diverse environmental representation and modeling techniques.

(3) Local trajectory replanning will be implemented to better adapt to dynamic changes in the operational environment.

(4) Human expertise in complex environments will be integrated into the motion planning algorithm to generate more flexible and empirical motion trajectories, enhancing driving safety and work efficiency.

(5) Higher-order constraints such as vehicle nonlinear dynamics characteristics (e.g., drive torque limitation, center of gravity offset) are further introduced to improve the physical executability of trajectories under extreme dynamic conditions.

CRediT authorship contribution statement

Chenlong Feng: Data curation, Methodology. **Jixin Wang:** Supervision, Project administration. **Qi Wang:** Software, Methodology. **Yuying Shen:** Writing – review & editing, Investigation. **Yunda Wei:** Validation, Data curation. **Shaokai Zhang:** Software, Investigation. **Shuyi Zhang:** Investigation, Data curation. **Quan Zhang:** Writing – review & editing, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by National Natural Science Foundation of China (No. 52272434) and China Postdoctoral Science Foundation (No. 2025M771361).

References

- [1] Y. Zhao, B. Lu, M. Alipour, Optimized structural inspection path planning for automated unmanned aerial systems, *Autom. Constr.* 168 (2024) 105764, <http://dx.doi.org/10.1016/j.autcon.2024.105764>.
- [2] Y. Shen, J. Wang, S. Mo, X. Gu, Data augmentation aided excavator activity recognition using deep convolutional conditional generative adversarial networks, *Adv. Eng. Inform.* 62 (2024) 102785, <http://dx.doi.org/10.1016/j.aei.2024.102785>.
- [3] X. Tong, S. Yu, G. Liu, X. Niu, C. Xia, J. Chen, Z. Yang, Y. Sun, A hybrid formation path planning based on A* and multi-target improved artificial potential field algorithm in the 2D random environments, *Adv. Eng. Inform.* 54 (2022) 101755, <http://dx.doi.org/10.1016/j.aei.2022.101755>.
- [4] J. Cho, K. Kim, Detection of moving objects in multi-complex environments using selective attention networks (SANet), *Autom. Constr.* 155 (2023) 105066, <http://dx.doi.org/10.1016/j.autcon.2023.105066>.
- [5] Y. Wang, X. Liu, Z. Ren, Z. Yao, X. Tan, Synchronized path planning and tracking for front and rear axles in articulated wheel loaders, *Autom. Constr.* 165 (2024) 105538, <http://dx.doi.org/10.1016/j.autcon.2024.105538>.
- [6] L. Bo, T. Zhang, H. Zhang, J. Hong, M. Liu, C. Zhang, B. Liu, 3D UAV path planning in unknown environment: A transfer reinforcement learning method based on low-rank adaption, *Adv. Eng. Inform.* 62 (2024) 102920, <http://dx.doi.org/10.1016/j.aei.2024.102920>.
- [7] H. Xie, C. Cui, X. Zhong, X. Zhong, Q. Liu, Real-time support terrain mapping and terrain adaptive local planning for quadruped robots, *IEEE Robot. Autom. Lett.* 9 (12) (2024) 11018–11025, <http://dx.doi.org/10.1109/LRA.2024.3486211>.
- [8] F. Aceró, K. Yuan, Z. Li, Learning perceptual locomotion on uneven terrains using sparse visual observations, *IEEE Robot. Autom. Lett.* 7 (4) (2022) 8611–8618, <http://dx.doi.org/10.1109/LRA.2022.3188108>.
- [9] R.L. Medrano, G.C. Thomas, C.G. Keais, E.J. Rouse, R.D. Gregg, Real-time gait phase and task estimation for controlling a powered ankle exoskeleton on extremely uneven terrain, *IEEE Trans. Robot.* 39 (3) (2023) 2170–2182, <http://dx.doi.org/10.1109/TRO.2023.3235584>.
- [10] L. Xu, K. Chai, Z. Han, H. Liu, C. Xu, Y. Cao, F. Gao, An efficient trajectory planner for car-like robots on uneven terrain, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2023, pp. 2853–2860, <http://dx.doi.org/10.1109/IROS5552.2023.10341558>.
- [11] G. Waibel, T. Low, M. Nass, D. Howard, T. Bandyopadhyay, P.V.K. Borges, How rough is the path? Terrain traversability estimation for local and global path planning, *IEEE Trans. Intell. Transp. Syst.* 23 (9) (2022) 16462–16473, <http://dx.doi.org/10.1109/TITS.2022.3150328>.
- [12] A. Manoharan, A. Sharma, H. Belsare, K. Pal, K.M. Krishna, A.K. Singh, Bi-level trajectory optimization on uneven terrains with differentiable wheel-terrain interaction model, 2024, <http://dx.doi.org/10.48550/arXiv.2404.03307>, arXiv: 2404.03307.
- [13] X. Hou, M. Gan, W. Wu, T. Zhao, J. Chen, Risk assessment and interactive motion planning with visual occlusion using graph attention networks and reinforcement learning, *Adv. Eng. Inform.* 62 (2024) 102941, <http://dx.doi.org/10.1016/j.aei.2024.102941>.
- [14] M. Sombolestan, Q. Nguyen, Adaptive-force-based control of dynamic legged locomotion over uneven terrain, *IEEE Trans. Robot.* 40 (2024) 2462–2477, <http://dx.doi.org/10.1109/TRO.2024.3381554>.
- [15] Z. Jian, Z. Lu, X. Zhou, B. Lan, A. Xiao, X. Wang, B. Liang, PUTN: A plane-fitting based uneven terrain navigation framework, 2022, arXiv:2203.04541.
- [16] M.V. Gasparino, A.N. Sivakumar, Y. Liu, A.E.B. Velasquez, V.A.H. Huguit, J. Rogers, H. Tran, G. Chowdhary, WayFAST: Navigation with predictive traversability in the field, *IEEE Robot. Autom. Lett.* 7 (4) (2022) 10651–10658, <http://dx.doi.org/10.1109/LRA.2022.3193464>.
- [17] J. Wang, L. Xu, H. Fu, Z. Meng, C. Xu, Y. Cao, X. Lyu, F. Gao, Towards efficient trajectory generation for ground robots beyond 2D environment, 2023, arXiv:2302.03323.
- [18] G. Hu, M. Cheng, E.H. Houssein, H. Jia, CMPSO: A novel co-evolutionary multi-group particle swarm optimization for multi-mission UAVs path planning, *Adv. Eng. Inform.* 63 (2025) 102923, <http://dx.doi.org/10.1016/j.aei.2024.102923>.
- [19] X. Jin, M.J. Er, Cooperative path planning with priority target assignment and collision avoidance guidance for rescue unmanned surface vehicles in a complex ocean environment, *Adv. Eng. Inform.* 52 (2022) 101517, <http://dx.doi.org/10.1016/j.aei.2021.101517>.
- [20] Z. Han, Y. Wu, T. Li, L. Zhang, L. Pei, L. Xu, C. Li, C. Ma, C. Xu, S. Shen, F. Gao, An efficient spatial-temporal trajectory planner for autonomous vehicles in unstructured environments, 2023, arXiv:2208.13160.
- [21] S. Mohammad Langari, F. Vahdatikhaki, A. Hammad, Improving the performance of RRT path planning of excavators by embedding heuristic rules, *Adv. Eng. Inform.* 62 (2024) 102724, <http://dx.doi.org/10.1016/j.aei.2024.102724>.
- [22] R. Sáez, D. Toratani, R. Mori, X. Prats, A flight-procedure generation framework based on an RRT* path planning algorithm, *Adv. Eng. Inform.* 64 (2025) 102975, <http://dx.doi.org/10.1016/j.aei.2024.102975>.
- [23] C. Guo, H. Liu, S. Nie, F. Zhang, H. Wan, L. Han, Terrain-adaptive hierarchical speed planning method for off-road environments, *IEEE Trans. Veh. Technol.* (2024) 1–16, <http://dx.doi.org/10.1109/TVT.2024.3450203>.
- [24] W. Li, Y. Wang, Y. Liang, D.T. Pham, Learning from demonstration for autonomous generation of robotic trajectory: Status quo and forward-looking overview, *Adv. Eng. Inform.* 62 (2024) 102625, <http://dx.doi.org/10.1016/j.aei.2024.102625>.
- [25] J. Zhao, C. Yang, W. Wang, Y. Li, T. Qie, B. Xu, An improved elitist-Q-Learning path planning strategy for VTOL air-ground vehicle using convolutional neural network mode prediction, *Adv. Eng. Inform.* 65 (2025) 103316, <http://dx.doi.org/10.1016/j.aei.2025.103316>.
- [26] Y.K. Nakka, S.-J. Chung, Trajectory optimization of chance-constrained nonlinear stochastic systems for motion planning under uncertainty, *IEEE Trans. Robot.* 39 (1) (2023) 203–222, <http://dx.doi.org/10.1109/TRO.2022.3197072>.
- [27] G. Hu, B. Du, K. Chen, G. Wei, Super eagle optimization algorithm based three-dimensional ball security corridor planning method for fixed-wing UAVs, *Adv. Eng. Inform.* 59 (2024) 102354, <http://dx.doi.org/10.1016/j.aei.2024.102354>.
- [28] C. Rosmann, F. Hoffmann, T. Bertram, Kinodynamic trajectory optimization and control for car-like robots, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, Vancouver, BC, 2017, pp. 5681–5686, <http://dx.doi.org/10.1109/IROS.2017.8206458>.

- [29] W. Ding, L. Zhang, J. Chen, S. Shen, Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor, 2019, [arXiv:1906.09788](https://arxiv.org/abs/1906.09788).
- [30] Q. Bi, X. Zhang, S. Zhang, R. Wang, L. Li, J. Yuan, LRAE: Large-region-aware safe and fast autonomous exploration of ground robots for uneven terrains, IEEE Robot. Autom. Lett. 9 (12) (2024) 11186–11193, <http://dx.doi.org/10.1109/LRA.2024.3486229>.
- [31] J. Dallas, M.P. Cole, P. Jayakumar, T. Ersal, Terrain adaptive trajectory planning and tracking on deformable terrains, IEEE Trans. Veh. Technol. 70 (11) (2021) 11255–11268, <http://dx.doi.org/10.1109/TVT.2021.3114088>.
- [32] Y. Ji, Y. Tanaka, Y. Tamura, M. Kimura, A. Umemura, Y. Kaneshima, H. Murakami, A. Yamashita, H. Asama, Adaptive motion planning based on vehicle characteristics and regulations for off-road UGVs, IEEE Trans. Ind. Inform. 15 (1) (2019) 599–611, <http://dx.doi.org/10.1109/TII.2018.2870662>.
- [33] V. Guizilini, F. Ramos, Variational Hilbert regression for terrain modeling and trajectory optimization, Int. J. Robot. Res. 38 (12–13) (2019) 1375–1387, <http://dx.doi.org/10.1177/0278364919844586>.
- [34] J. Seo, S. Sim, I. Shim, Learning off-road terrain traversability with self-supervisions only, IEEE Robot. Autom. Lett. 8 (8) (2023) 4617–4624, <http://dx.doi.org/10.1109/LRA.2023.3284356>, arXiv:2305.18896.
- [35] W. Li, M. Liao, W. Zou, A terrain segmentation network for navigable areas with global strip reliability evaluation and dynamic fusion, Expert Syst. Appl. 265 (2025) 125964, <http://dx.doi.org/10.1016/j.eswa.2024.125964>.
- [36] R.A. MacDonald, S.L. Smith, Active sensing for motion planning in uncertain environments via mutual information policies, Int. J. Robot. Res. 38 (2–3) (2019) 146–161, <http://dx.doi.org/10.1177/0278364918772024>.
- [37] D. Kappler, F. Meier, J. Issac, J. Mainprice, C.G. Cifuentes, M. Wuthrich, V. Berenz, S. Schaal, N. Ratliff, J. Bohg, Real-time perception meets reactive motion generation, IEEE Robot. Autom. Lett. 3 (3) (2018) 1864–1871, <http://dx.doi.org/10.1109/LRA.2018.2795645>.
- [38] Z. Jian, S. Zhang, L. Sun, W. Zhan, N. Zheng, M. Tomizuka, Long-term dynamic window approach for kinodynamic local planning in static and crowd environments, IEEE Robot. Autom. Lett. 8 (6) (2023) 3294–3301, <http://dx.doi.org/10.1109/LRA.2023.3266664>.
- [39] M. Dobrevski, D. Skočaj, Dynamic adaptive dynamic window approach, IEEE Trans. Robot. 40 (2024) 3068–3081, <http://dx.doi.org/10.1109/TRO.2024.3400932>.
- [40] L. Chang, L. Shan, C. Jiang, Y. Dai, Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment, Auton. Robots 45 (1) (2021) 51–76, <http://dx.doi.org/10.1007/s10514-020-09947-4>.
- [41] Y. Wang, C. Gong, J. Gong, P. Jia, Motion planning for off-road autonomous driving based on human-like cognition and weight adaptation, J. Field Robot. n/a (n/a) <http://dx.doi.org/10.1002/rob.22345>.
- [42] J.-K. Huang, J.W. Grizzle, Efficient anytime CLF reactive planning system for a bipedal robot on undulating terrain, IEEE Trans. Robot. 39 (3) (2023) 2093–2110, <http://dx.doi.org/10.1109/TRO.2022.3228713>.
- [43] D. Chen, M. Zhuang, X. Zhong, W. Wu, Q. Liu, RSPMP: Real-time semantic perception and motion planning for autonomous navigation of unmanned ground vehicle in off-road environments, Appl. Intell. (2022) <http://dx.doi.org/10.1007/s10489-022-03283-z>.
- [44] D.H. Lee, S.S. Lee, C.K. Ahn, P. Shi, C.-C. Lim, Finite distribution estimation-based dynamic window approach to reliable obstacle avoidance of mobile robot, IEEE Trans. Ind. Electron. 68 (10) (2021) 9998–10006, <http://dx.doi.org/10.1109/TIE.2020.3020024>.
- [45] Z. Gao, J. Qin, S. Wang, Y. Wang, Boundary gap based reactive navigation in unknown environments, IEEE/CAA J. Autom. Sin. 8 (2) (2021) 468–477, <http://dx.doi.org/10.1109/JAS.2021.1003841>.
- [46] Z. Xu, Y. Chen, Z. Jian, J. Tan, X. Wang, B.L. Liang, Hybrid trajectory optimization for autonomous terrain traversal of articulated tracked robots, IEEE Robot. Autom. Lett. 9 (1) (2024) 755–762, <http://dx.doi.org/10.1109/LRA.2023.3337593>.
- [47] J. Ou, S.H. Hong, G. Song, Y. Wang, Hybrid path planning based on adaptive visibility graph initialization and edge computing for mobile robots, Eng. Appl. Artif. Intell. 126 (2023) 107110, <http://dx.doi.org/10.1016/j.engappai.2023.107110>.
- [48] T. Zhang, S. Xu, Y. Gao, P. Wang, P. Schonfeld, Y. Zou, Q. He, 3D constrained hybrid A*: Improved vehicle path planning algorithm for cost-effective road alignment design, Autom. Constr. 166 (2024) 105645, <http://dx.doi.org/10.1016/j.autcon.2024.105645>.
- [49] D. González, J. Pérez, V. Milanés, F. Nashashibi, A review of motion planning techniques for automated vehicles, IEEE Trans. Intell. Transp. Syst. 17 (4) (2016) 1135–1145, <http://dx.doi.org/10.1109/TITS.2015.2498841>.
- [50] D.S. Sundarsingh, J. Bhagya, J. Chatrola, P. Jagtap, Autonomous exploration using ground robots with safety guarantees, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2023, pp. 9745–9750, <http://dx.doi.org/10.1109/IROS5552.2023.10341929>.
- [51] S.-W. Yoo, E.-I. Son, S.-W. Seo, Traversability-aware adaptive optimization for path planning and control in mountainous terrain, IEEE Robot. Autom. Lett. 9 (6) (2024) 5078–5085, <http://dx.doi.org/10.1109/LRA.2024.3387042>.
- [52] M. Thoresen, N.H. Nielsen, K. Mathiassen, K.Y. Pettersen, Path planning for UGVs based on traversability hybrid a*, IEEE Robot. Autom. Lett. 6 (2) (2021) 1216–1223, <http://dx.doi.org/10.1109/LRA.2021.3056028>.
- [53] H. Lee, J. Kwon, C. Kwon, Learning-based uncertainty-aware navigation in 3D off-road terrains, in: 2023 IEEE International Conference on Robotics and Automation, ICRA, IEEE, London, United Kingdom, 2023, pp. 10061–10068, <http://dx.doi.org/10.1109/ICRA48891.2023.10161543>.
- [54] M. Toscano-Moreno, A. Mandow, M.A. Martínez, A. García-Cerezo, DEM-AIA: Asymmetric inclination-aware trajectory planner for off-road vehicles with digital elevation models, Eng. Appl. Artif. Intell. 121 (C) (2023) <http://dx.doi.org/10.1016/j.engappai.2023.105976>.
- [55] J. Liu, X. Chen, J. Xiao, S. Lin, Z. Zheng, H. Lu, Hybrid map-based path planning for robot navigation in unstructured environments, 2023, <http://dx.doi.org/10.48550/arXiv.2303.05304>, arXiv:2303.05304.
- [56] J. Hu, Y. Hu, C. Lu, J. Gong, H. Chen, Integrated path planning for unmanned differential steering vehicles in off-road environment with 3D terrains and obstacles, IEEE Trans. Intell. Transp. Syst. 23 (6) (2022) 5562–5572, <http://dx.doi.org/10.1109/TITS.2021.3054921>.
- [57] F. Atas, G. Cielniak, L. Grimstad, Navigating in 3D uneven environments through supervoxels and nonlinear MPC, in: 2023 European Conference on Mobile Robots, ECMR, IEEE, Coimbra, Portugal, 2023, pp. 1–8, <http://dx.doi.org/10.1109/ECMR5916.2023.10256342>.
- [58] I. Jeong, Y. Jang, J. Park, Y.K. Cho, Motion planning of mobile robots for autonomous navigation on uneven ground surfaces, J. Comput. Civ. Eng. 35 (3) (2021) 04021001, [http://dx.doi.org/10.1061/\(ASCE\)CP.1943-5487.0000963](http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000963).
- [59] C. Wang, L. Meng, S. She, I.M. Mitchell, T. Li, F. Tung, W. Wan, M.Q.-H. Meng, C.W. de Silva, Autonomous mobile robot navigation in uneven and unstructured indoor environments, 2017, [arXiv:1710.10523](http://arxiv.org/abs/1710.10523).
- [60] P. Krüsi, P. Furgale, M. Bosse, R. Siegwart, Driving on point clouds: Motion Planning, trajectory optimization, and terrain assessment in generic nonplanar environments, J. Field Robot. 34 (5) (2017) 940–984, <http://dx.doi.org/10.1002/rob.21700>.
- [61] H. Zheng, Y. Lu, J. Jie, B. Hou, M. Zhang, Y. Zhang, Gaussian adaptive strategy based multi-objective evolutionary optimization for path planning on uneven terrains, IEEE Robot. Autom. Lett. 9 (1) (2024) 539–546, <http://dx.doi.org/10.1109/LRA.2023.3334675>.
- [62] J. Zhou, R. He, Y. Wang, S. Jiang, Z. Zhu, J. Hu, J. Miao, Q. Luo, Autonomous driving trajectory optimization with dual-loop iterative anchoring path smoothing and piecewise-jerk speed optimization, IEEE Robot. Autom. Lett. 6 (2) (2021) 439–446, <http://dx.doi.org/10.1109/LRA.2020.3045925>.
- [63] C. Shen, S. Yu, B.I. Epureanu, T. Ersal, An efficient global trajectory planner for highly dynamical nonholonomic autonomous vehicles on 3-D terrains, IEEE Trans. Robot. 40 (2024) 1309–1326, <http://dx.doi.org/10.1109/TRO.2023.3344030>.
- [64] H. Li, P. Chen, G. Yu, B. Zhou, Y. Li, Y. Liao, Trajectory planning for autonomous driving in unstructured scenarios based on deep learning and quadratic optimization, IEEE Trans. Veh. Technol. 73 (4) (2024) 4886–4903, <http://dx.doi.org/10.1109/tvt.2023.3330581>.
- [65] K. You, L. Ding, Y. Jiang, Z. Wu, C. Zhou, End-to-end deep learning for reverse driving trajectory of autonomous bulldozer, Knowl.-Based Syst. 252 (2022) 109402, <http://dx.doi.org/10.1016/j.knosys.2022.109402>.
- [66] T. Zhang, T. Fu, T. Ni, H. Yue, Y. Wang, X. Song, Data-driven excavation trajectory planning for unmanned mining excavator, Autom. Constr. 162 (2024) 105395, <http://dx.doi.org/10.1016/j.autcon.2024.105395>.
- [67] J. You, H. Shi, Z. Jiang, Z. Huang, R. Gan, K. Wu, X. Cheng, X. Li, B. Ran, V2X-VLM: End-to-end V2X cooperative autonomous driving through large vision-language models, 2024, <http://dx.doi.org/10.48550/arXiv.2408.09251>, arXiv:2408.09251.
- [68] X. Zhou, X. Han, F. Yang, Y. Ma, A.C. Knoll, OpenDriveVLA: Towards end-to-end autonomous driving with large vision language action model, 2025, <http://dx.doi.org/10.48550/arXiv.2503.23463>, arXiv:2503.23463.
- [69] X. Hou, J. Zhang, C. He, Y. Ji, J. Zhang, J. Han, Autonomous driving at the handling limit using residual reinforcement learning, Adv. Eng. Inform. 54 (2022) 101754, <http://dx.doi.org/10.1016/j.aei.2022.101754>.
- [70] H. Park, Trajectory planning for autonomous vehicle using iterative reward prediction in reinforcement learning, 2024, <http://dx.doi.org/10.48550/arXiv.2404.12079>, arXiv:2404.12079.
- [71] E. Zhang, R. Zhang, N. Masoud, Predictive trajectory planning for autonomous vehicles at intersections using reinforcement learning, Transp. Res. Part C: Emerg. Technol. 149 (2023) 104063, <http://dx.doi.org/10.1016/j.trc.2023.104063>.
- [72] W. Huang, Y. Li, S. Yuan, J. Teng, H. Qin, Y. Bian, Reachable sets-based trajectory planning combining reinforcement learning and iLQR, 2025, <http://dx.doi.org/10.48550/arXiv.2503.17398>, arXiv:2503.17398.
- [73] J. Manuel Davila Delgado, L. Oyedele, Robotics in construction: A critical review of the reinforcement learning and imitation learning paradigms, Adv. Eng. Inform. 54 (2022) 101787, <http://dx.doi.org/10.1016/j.aei.2022.101787>.
- [74] A. Hussein, M.M. Gaber, E. Elyan, C. Jayne, Imitation learning: A survey of learning methods, ACM Comput. Surv. 50 (2) (2017) 21:1–21:35, <http://dx.doi.org/10.1145/3054912>.

- [75] R. Li, Z. Zou, Enhancing construction robot learning for collaborative and long-horizon tasks using generative adversarial imitation learning, *Adv. Eng. Inform.* 58 (2023) 102140, <http://dx.doi.org/10.1016/j.aei.2023.102140>.
- [76] J. Gao, Z. Tao, N. Jaquier, T. Asfour, K-VIL: Keypoints-based visual imitation learning, *IEEE Trans. Robot.* 39 (5) (2023) 3888–3908, <http://dx.doi.org/10.1109/TRO.2023.3286074>.
- [77] A. Yarovoi, Review of simultaneous localization and mapping (SLAM) for construction robotics applications, *Autom. Constr.* 162, 105344, <http://dx.doi.org/10.1016/j.autcon.2024.105344>.
- [78] P. Fankhauser, M. Bloesch, M. Hutter, Probabilistic terrain mapping for mobile robots with uncertain localization, *IEEE Robot. Autom. Lett.* 3 (4) (2018) 3019–3026, <http://dx.doi.org/10.1109/LRA.2018.2849506>.
- [79] F. Gao, W. Wu, Y. Lin, S. Shen, Online safe trajectory generation for quadrotors using fast marching method and Bernstein basis polynomial, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, Brisbane, QLD, 2018, pp. 344–351, <http://dx.doi.org/10.1109/ICRA.2018.8462878>.