Journal of Intelligent & Robotic Systems (2024) 110:41
https://doi.org/10.1007/s10846-024-02073-8

REGULAR PAPER

Terrain-Shape-Adaptive Coverage Path Planning With Traversability Analysis

Wenwei Qiu[1] · Dacheng Zhou[1] · Wenbo Hui[1] · Afimbo Reuben Kwabena[1] · Yubo Xing[1] · Yi Qian[1] · Quan Li[1] · Huayan Pu[1] · Yangmin Xie[1,2]

Received: 14 May 2023 / Accepted: 4 February 2024 / Published online: 7 March 2024

Abstract

Coverage path planning (CPP) is in great demand with applications in agriculture, mining, manufacturing, etc. Most research in this area focused on 2D CPP problems solving the coverage problem with irregular 2D maps. Comparatively, CPP on uneven terrains is not fully solved. When there are many slopy areas in the working field, it is necessary to adjust the path shape and make it adapt to the 3D terrain surface to save energy consumption. This article proposes a terrain-shape-adaptive CPP method with three significant features. First, the paths grow by themselves according to the local terrain surface shapes. Second, the growth rule utilizes the 3D terrain traversability analysis, which makes them automatically avoid entering hazardous zones. Third, the irregularly distributed paths are connected under an optimal sequence with an improved genetic algorithm. As a result, the method can provide an autonomously growing terrain-adaptive coverage path with high energy efficiency and coverage rate compared to previous research works. It is demonstrated on various maps and is proven to be robust to terrain conditions.

1 Introduction

Coverage Path Planning (CPP) is a fundamental problem in the field of robotics, involving the determination of an "optimal" continuous path that effectively covers a given workspace while avoiding obstacles to ensure safe operation. Its primary objectives are to minimize trajectory repetition for efficiency in terms of energy consumption and time utilization while maximizing coverage rate to achieve optimal operational performance. The CPP technique finds applications across various domains, including but not limited to vacuum robots [1, 2], lawnmowers [3], agricultural equipment [4, 5], mine detectors [6], deep-sea mining machines [7], space rovers [8] and underwater structure inspection [9]. These applications involve scenarios where mobile robots are required to cover their designated work areas comprehensively. The major performance considerations include the turning costs [10], the path length [11], and the coverage rate [12].

Early research primarily concentrated on resolving the coverage issue in 2D maps, which assume a flat terrain shape. The primary goal of 2D algorithms was to identify an efficient continuous path that covers irregularly shaped maps while taking obstacles into account [13]. 2D methods can be categorized into traditional algorithms, heuristic-based algorithms, and deep learning methods. Traditional algorithms frequently employ a Zig-zag path pattern [14, 15] or the Spanning Tree Coverage (STC) algorithm [16] to plan the coverage of target areas. The Zig-zag method necessitates simplification for intricate 2D maps with irregular shapes and obstacles. In such cases, the whole workspace is typically partitioned into obstacle-free subsections. Zig-zag paths are independently planned within each subsection and subsequently connected to form a continuous path. STC is acknowledged for its effectiveness in handling complex maps and non-backtracking paths but often requires a down-sampling of the grid map. Heuristic algorithms frequently employ swarm intelligence [17, 18], which is inspired by the social behavior of living organisms. This involves collaboration within the swarm to iteratively optimize path generation through information exchange. Swarm intelligence demonstrates strong adaptability but generally requires significant computational resources and a high level of environmental information accuracy. Another newly developed Heuristic algorithm is the Glasius Bio-inspired Neural Network (GBNN) algorithm [19], which stands out by incorporating bio-inspired neu-

✉ Yangmin Xie
xieym@shu.edu.cn

Extended author information available on the last page of the article

ral network principles. It adapts to dynamic environments through the use of difference equations and dynamic neural activities. Deep learning methods primarily use reinforcement learning to learn path-generation actions [20, 21]. It trains the model through repeated episodes, and its performance is sensitive to the setup of the reward function. Additionally, it typically requires a considerable amount of time for training to achieve stable and desirable results.

When dealing with distinct working areas, researchers often needed to find the shortest path that connects all subsections once the paths within each subsection were established [22]. This task resembles the well-known Traveling Salesman Problem (TSP) [23, 24]. The optimization challenge was usually tackled using Genetic Algorithms (GA), where the chromosomes represented the sequence of partitions and the orientation of each partition's template. More recently, deep reinforcement learning techniques have emerged as an alternative to traditional optimization methods for solving the TSP, aiming to achieve higher time efficiency and improved optimality [25].

However, the 2D CPP approaches introduced above are not suitable for scenarios involving uneven terrains. In such cases, the relationship between driving directions and the terrain slope directions becomes the most critical factor affecting energy efficiency [10]. As illustrated in Fig. 1, working on paths that involve significant changes in elevation (up-down paths) can result in nearly twice the energy cost compared to following contours on the same 3D terrain. Consequently, researchers have turned their attention to studying CPP in the context of 3D terrains over the past decade.

Most 3D CPP studies have adopted the Zig-zag path pattern inherited from the 2D cases [23], with modifications made primarily to the cost function. In other words, the energy consumption function now incorporates additional parameters such as the local terrain inclination angle and the heading direction angle of the vehicles. Some literature [26] has introduced an additional term in the cost function, considering the path distance deformation and the corresponding loss of coverage. However, these studies have not deviated from the traditional straight-line path setting.

Contrary to the traditional approach, two notable studies [4, 10] have emphasized the significance of adapting the local path shape to elevation changes in CPP. These works highlighted that simply adjusting the driving direction along straight lines, as depicted in Fig. 1(a), does not yield optimal performance across complex 3D terrains. Instead, they demonstrated that driving along the contours of the terrain, as shown in Fig. 1(b), significantly reduces the cost of work in terms of energy consumption and soil erosion.

Although these studies introduced the innovative concept of terrain shape-adaptive curve patterns in CPP, they suffer from certain limitations that hinder their widespread implementation. In one study [4], the entire field was decomposed into subsections, and a contour curve was used as the seed path, with parallel curves grown to cover each subsection. However, this approach necessitates manual field decomposition and seed curve construction, which is neither convenient nor optimal. Moreover, it assumes a field without unpassable areas, which is unrealistic in many real-world scenarios. The other study [10] also requires pre-decomposition of the field into subsections containing only monotonic surfaces [27]. This process is time-consuming and can result in fragmented partitions when dealing with complex surface shapes. Considering these concerns, there is a need for a CPP method that covers 3D terrains and offers a more convenient and practical implementation process.

Firstly, it is worth noting that path planning for Unmanned Ground Vehicles (UGVs) to avoid hazards on 3D surfaces relies on terrain traversability analysis, an extensively studied field [28, 29]. The CPP problem, which is closely related to terrain shapes, can potentially benefit from the findings of traversability analysis for path generation purposes. By leveraging these results, the need for complex cost function optimization can be reduced, and the avoidance of unpassable areas in CPP can be automatically achieved. Secondly, the necessity of planning coverage paths within separate subsections raises questions, as it often introduces additional complexity in generating the subsections and leads to impractical disconnection of paths at the boundaries.

Based on the considerations mentioned above, we have developed a terrain shape-adaptive CPP approach applicable to ground vehicle scenarios navigating uneven terrains, where energy efficiency is a critical factor. Examples include agricultural activities in mountainous regions or rover operations on irregular surfaces of exoplanets. This incorporates the following attractive and innovative features:

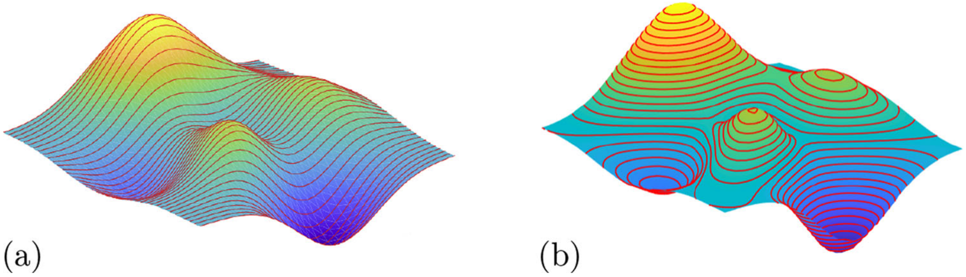


Fig. 1 The comparison of two kinds of coverage paths (a) driving up-down hills (b) driving along the contours

(1). It generates the coverage path in accordance with a traversability map, which combines obstacle avoidance capabilities with field coverage path planning.

(2). The paths automatically grow across the entire terrain surface, eliminating the need for field decomposition and simplifying the path planning process.

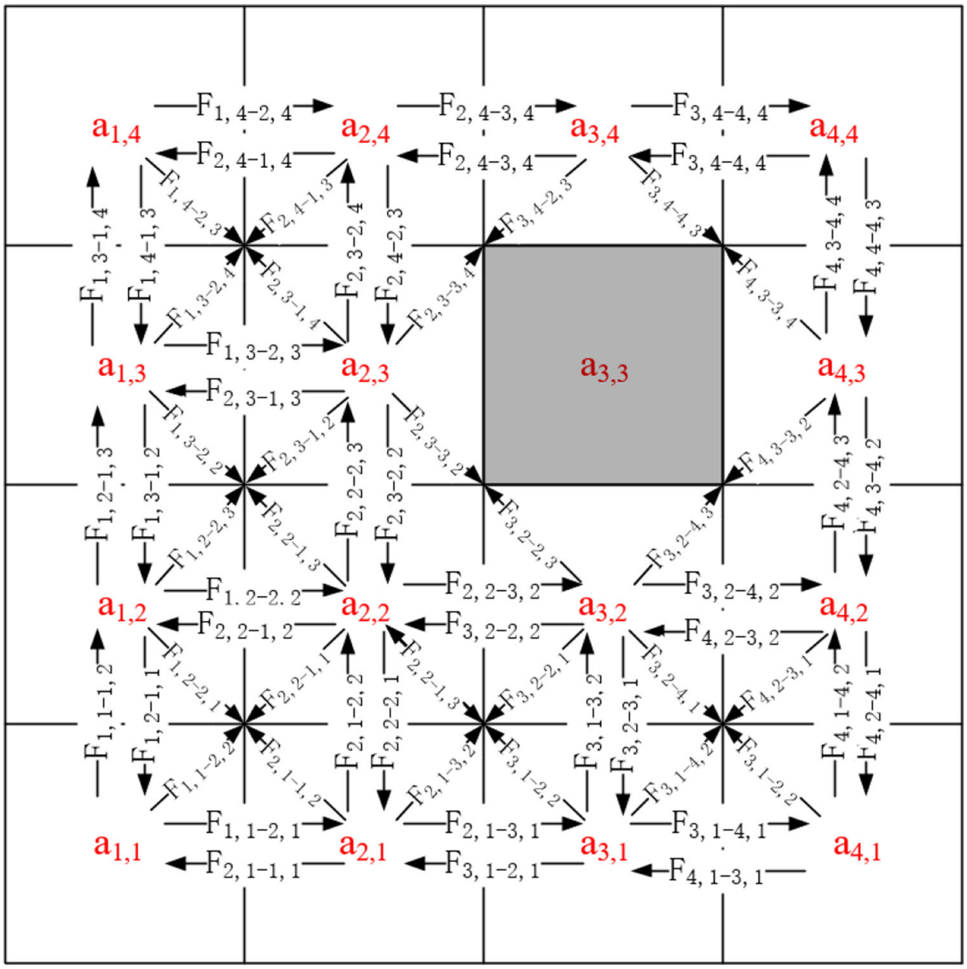
(3). It determines the optimal starting and ending points of the paths to minimize total travel cost, utilizing a modified GA to solve the path connection problem.

The remainder of this article is organized as follows: In Part II, we introduce the concept of a traversability map for 3D environments as a preliminary step. Part III outlines the method for generating CPP paths based on the traversability map. Part IV presents the modified GA algorithm designed to address the path connection problem. In Part V, we provide experimental data to validate and discuss the effectiveness of our proposed methods. Finally, in Part VI, we conclude the article, summarizing the key findings and implications of our terrain shape-adaptive CPP approach.

2 Preliminary Study On Traversability Analysis

The traversability map used in this article is from the authors' previous work on hazard avoidance path planning over 3D irregular terrains [28]. For any discretized vehicle location on the map, it provides the driving safety indexes in 8 driving directions. As shown in Fig. 2, $a_{i,j}$ denotes the cell with i and j as the indexes in the x and y directions, and $F_{i,j-(i+1,j+1)}$ denotes the safety cost when the center of the vehicle is placed at $a_{i,j}$ in the direction from $a_{i,j}$ to $a_{i+1,j+1}$. If a cell has no terrain data, it is marked as unknown, as the $a_{3,3}$ in Fig. 2.

Fig. 2 The structure of the traversability map



For traversable areas, the local traversability indexes are determined by the tilt angle of the vehicle. As shown in Fig. 3, $\mathbf{Z}_W$ indicates the vertical direction, and $\mathbf{Z}$ is the normal vector of the chassis plane with its vector elements denoted as (z_1, z_2, z_3). The angle between the two is the tilt angle α. The details to calculate $\mathbf{Z}$ by the interacted relationship between the vehicle and the terrain surface can be found in [28]. The tilt angle is calculated by

$$\alpha = \arccos \frac{\mathbf{Z} \cdot \mathbf{Z}_W}{|\mathbf{Z}||\mathbf{Z}_W|} \tag{1}$$

Denote $\bar{\alpha}$ as the largest tolerable tilt angle, and the safety cost of the vehicle at (i, j) facing the neighbor cell (m, n) is calculated by

$$F_{i,j-m,n} = \begin{cases} \tan(\alpha) & \alpha < \bar{\alpha} \\ \infty & \alpha \geq \bar{\alpha} \end{cases} \tag{2}$$

With the aforementioned traversaibility analysis, we can utilize the results to further find proper driving direction with efficient energy consumption for the CPP problem. Theoretically, when the vehicle is driving along a contour, the pitch angle is 0. In this case, the climbing energy is minimal and the soil erosion is largely prevented. Therefore, we relate the travel cost of a local path with the pitch angle of the vehicle chassis. Simply speaking, driving in a direction with a smaller pitch angle is beneficial to save energy when the vehicle is on a slope.

$\mathbf{Y}$ is the heading direction of the vehicle with its vector elements denoted as (y_1, y_2, y_3), where $y_1 = m - i$ and $y_2 = n - j$. Then y_3 can be calculated as

$$y_3 = \frac{z_1 y_1 + z_2 y_2}{-z_3} \tag{3}$$

The pitch angle α_p can be calculated as

$$\alpha_p = \frac{\pi}{2} - \arccos \frac{\mathbf{Y} \cdot \mathbf{Z}_W}{|\mathbf{Y}||\mathbf{Z}_W|} \tag{4}$$

3 Coverage Path Generation

The most distinguishing feature of our method is that the paths are not planned by an explicit rule. Instead, they grow automatically by adapting themselves to the local terrain shape and following the neighbour path in the meanwhile to achieve terrain coverage. We accomplish this by following two principles: (1) the growth of the path mimics the shape of previous paths, and (2) the terrain adaption happens only on terrains with considerable inclinations.

The operational flow of the method is shown in Fig. 4. In the first step, the field is classified into several categories by terrain classification rules so that the paths can grow by different criteria. The TSA path growth algorithm alternates between two main functional modules: seed path growth and subordinary path growth. A seed path autonomously grows to adapt to the terrain shape without necessitating shape mimicry. Once a seed path is established, the subordinary paths emulate previous paths until their termination conditions are met. The seed path and its subordinary paths collectively form a set of coverage path curves. The algorithm continues to extend this set until no new subordinary paths can be generated, at which point it initiates a search for a new seed path growth. The process terminates until the required coverage percentage is achieved. In the following content, we introduce the details of the method in a sequence of terrain classification, seed path growth, and subordinary path growth.

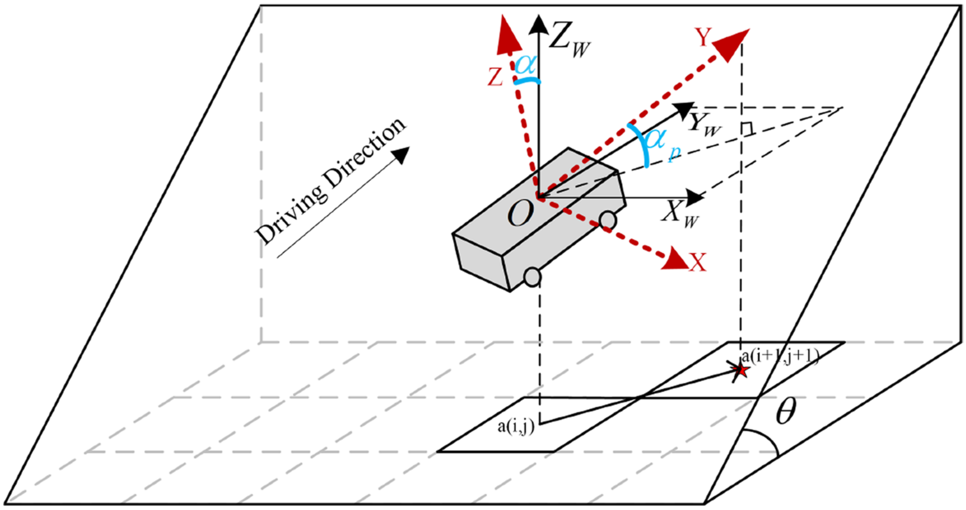


Fig. 3 The geometric relationship between a vehicle and the local uneven terrain surface

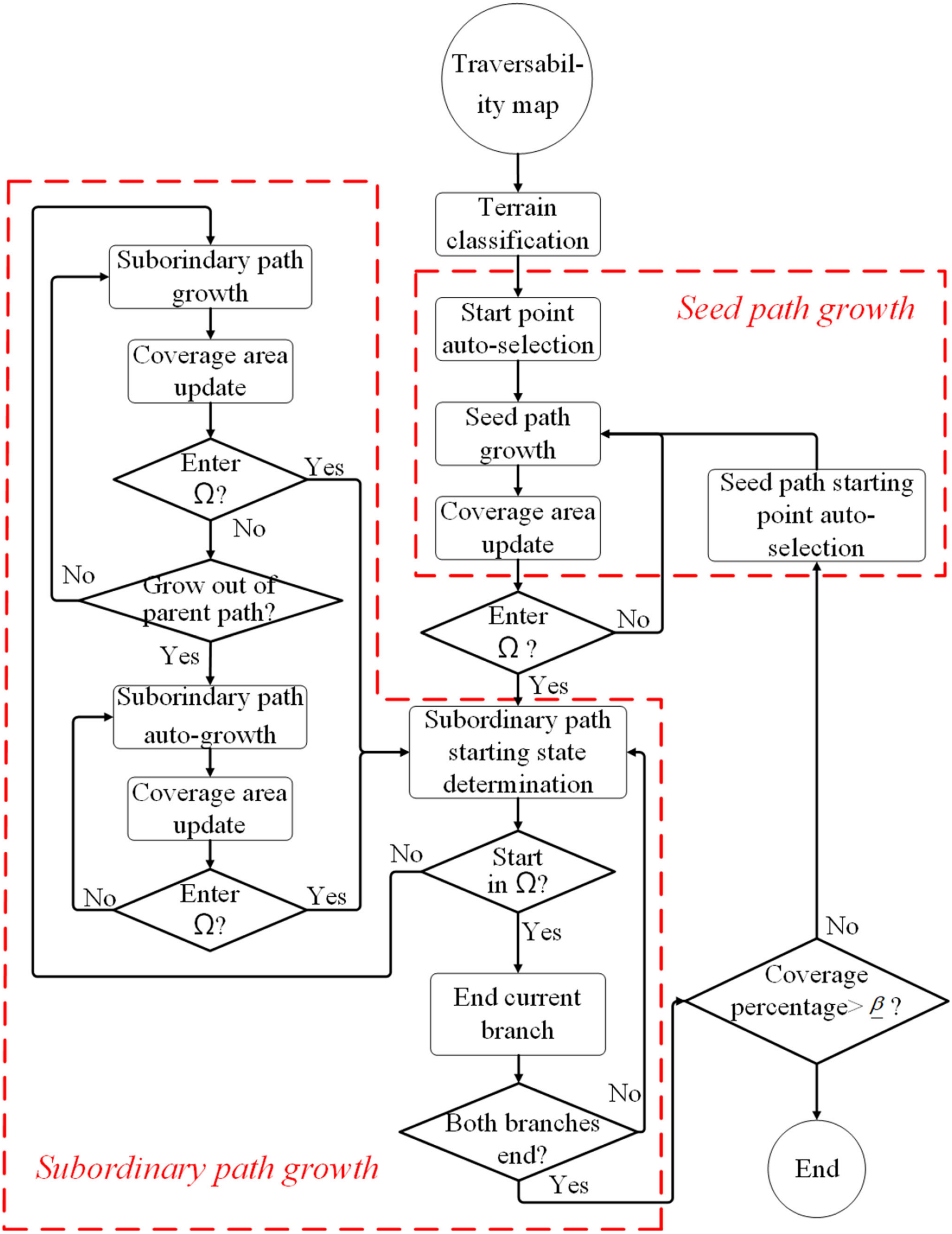


Fig. 4 The flowchart for coverage path growth

3.1 Terrain classification

As illustrated in Section 2, the traversability analysis for the uneven terrain provides a gridded map with directional driving safety indexes in 8 directions for all the locations in the entire area. We can utilize the information for two purposes: (1) to separate the passable area from the unpassable areas and (2) to distinguish the flat areas from the uneven areas. The former serves to prevent planning paths in dangerous places, and the latter helps to build path planning criteria based on the local terrain shape conditions. The traversability map's passable cells must exhibit dominance to ensure a coherent and continuous path-growing space for effective path-planning algorithms. The success of the solution hinges on the prevalence of connected passable regions, as opposed to scattered or discontinuous areas, to optimize path planning outcomes.

As shown in Fig. 5, when projecting the 3D surface on the horizontal plane with grids, they are categorized by the corresponding traversability indexes $F_{i,j-(m,n)}$ in Fig. 2. For a cell $a_{i,j}$, if any of the eight elements in $F_{i,j-(m,n)}$ is larger than some threshold value, the cell is considered unpassable. Otherwise, it is passable. α_s is a user-specified angle to distinguish between flat and slopy surfaces. For a passable cell, if the maximum value in $F_{i,j-(m,n)}$ is larger than $tan\alpha_s$, the grid is a slopy cell; otherwise, it is a flat one.

Given the high-resolution nature of the original grid for accurate traversability analysis, we find it necessary to downsample it to facilitate rapid path growth and to avoid serrated paths caused by small local terrain variations. We denote the downsampling factor as N. For an $N * N$ patch (as illustrated with $N = 2$ in Fig. 5), if it contains more than half unpassable cells, it is classified as an unpassable area; otherwise, it is deemed passable. In the case of a passable area, if it has more sloped cells than flat cells, it is categorized as a sloped area; otherwise, it is considered flat. With all terrain cells labeled by categories, they are prepared for the subsequent path growth processes.

3.2 Seed path growth

A seed path grows by itself, without the need to follow the shape of any other ones. For the very first seed path, its start point is selected as the center point of the largest slopy area. There are two reasons for this choice. (1) The seed path influences the shape of its subordinary paths. This choice provides a well-defined initial adaptation to the three-dimensional terrain, particularly effective in sloped areas, to achieve optimal energy efficiency. (2) Initiating from the central region allows for growth on both sides of the path, expanding the influence area of the seed path and reducing the number of required seed paths. It reflects the terrain shape tendency in uneven areas. The covered area of the existing path accumulates and updates as long as a new path node grows. Denote the covered area as S_{cov}. The unfeasible area is the union of the covered area and the unpassable area, denoted as $\Omega = S_{cov} \bigcup S_{up}$. When a seed path grows into Ω, it stops. The left feasible area is the complementary set of Ω, denoted as $\complement_{\Omega}$.

Start point auto-selection. When no more subordinary path can grow in $\complement_{\Omega}$ by following the existing ones, the algorithm has to generate another new seed path. In this scenario, the initiation point of a new seed path is identified in the vicinity of the termini of existing paths, ensuring that the newly growing paths closely align with previously expanded regions, minimizing the gaps between different growing sets. Local square areas around every endpoint of the paths are examined as candidates for new starting points, as shown in Fig. 6. Denote the index of an existing path as p and the square areas at the two ends of the p^{th} path as S_1^p and S_2^p,

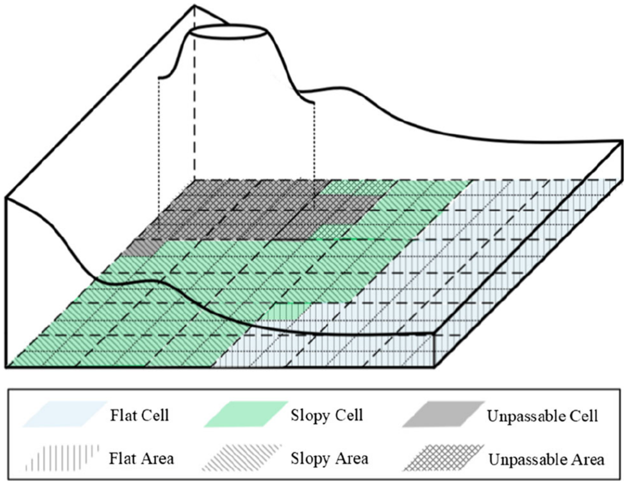


Fig. 5 Terrain shape classification

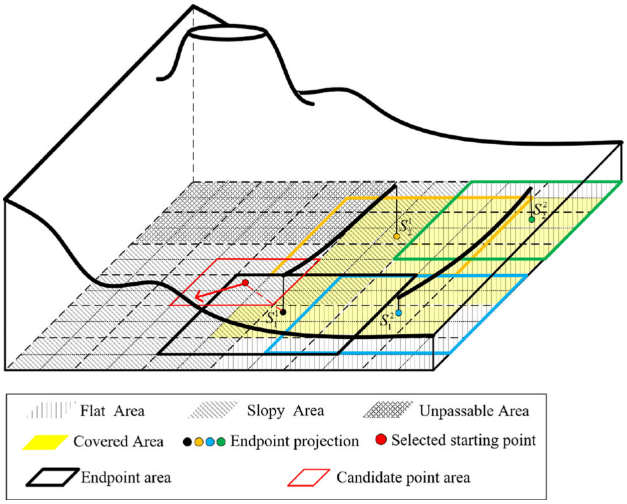


Fig. 6 Starting point auto-selection for a seed path

respectively. The side length of the areas is denoted as l_{se}. If the coverage radius of a path is d_{pl}, l_{se} is usually selected as $2d_{pl}$. The percentages of the cells belonging to $\mathbb{C}_{\Omega}$ in all S_1^p and S_2^p are calculated, which are denoted as wp_1^p and wp_2^p. The end area with the largest value of wp_1^p or wp_2^p is selected as the starting area for the new seed path. Denote the corresponding path endpoint as p_n. In Fig. 6, p_n corresponds to the S_1^1.

The cells belonging to $\mathbb{C}_{\Omega}$ in S_1^1 are candidate spots for the starting point of the following seed path. For each candidate, calculate the percentage of the $\mathbb{C}_{\Omega}$ cells in the square with length l_{ss} and centered at the candidate. The candidate point with the highest percentage of $\mathbb{C}_{\Omega}$ cells wins, denoted as p_{ns}. In Fig. 6, the red dot is the cell in S_1^1 with the largest number of $\mathbb{C}_{\Omega}$ in its neighborhood, so it is selected as the next starting point for a new seed path. The operation above is to find the optimal solution with a starting place that is close enough to the ending locations of the existing paths and, in the meantime, far enough to avoid too much overlap with the covered area. Once the starting point is determined, an initial growing direction needs to be specified. It is chosen by finding the feasible driving direction closest to the perpendicular line of $\overrightarrow{p_{ns}p_n}$, as shown in Fig. 6 by the red arrow. This helps the new path to grow parallel to the existing ones.

Seed path growth. With the starting point and the initial growing direction determined, the seed curve can grow by specific rules. We considered two constraints in defining a cost function to determine the path-growing direction: the climbing slope and the turning cost. When the vehicle is in slopy areas, it is desirable to drive along the contours. In this case, the smaller the pitch angle along a certain driving direction, the smaller the climbing energy cost could be. In addition, the vehicle is expected to keep its original driving direction to save energy in the redirection actions. Accordingly, we design the path-growing cost in the slopy areas in Eq. 5. The first term in the equation symbolizes the inclination to traverse in the direction with a gentler slope, while the second term signifies the necessity to maintain a consistent driving direction. Consequently, the path exhibits a tendency to extend in the direction characterized by the smallest J_{s1}. In regions where the terrain is predominantly flat, the impact of terrain slope becomes negligible, rendering the first term in Eq. 5 ineffective. As a result, the seed path undergoes growth in an unaltered direction, essentially forming a straight line.

$$J_{s1}(k) = \frac{tan(\alpha_p^k)}{tan(\bar{\alpha})} + \lambda_{s1} * tan(\Delta\omega^k) \tag{5}$$

where $k = 1, 2, \cdots, 8$ is the index for the eight driving directions to choose, and $\Delta\omega^k$ is the angle between the current and the next step driving directions, λ_{s1} is a weight factor. A lower value for λ_{s1} underscores a focus on terrain adaptation, while a higher value prioritizes the avoidance of sharp turns.

3.3 Subordinary path growth

Once a seed path (the red curve in Fig. 7) has grown, two child paths (the black curves in Fig. 7) grow by its two sides. Then the child paths have their child paths grow until the start point for the new child path enters Ω. Therefore, for a seed curve, two branches of descendant paths grow beside it. Similar to the seed path generation, we need to determine the starting point, the initial growing direction, and the growth strategy for the subordinary paths.

Starting state determination. The starting point and initial direction of a child path are closely related to its parent path. As shown in Fig. 7, projecting the parent path on the horizontal plane, the starting point of a child path is chosen as the closest cell at the perpendicular line of the projection at the parent path's starting point, with a shift distance d_{pl}. The initial growing direction is chosen as the feasible direction that is mostly parallel to the tangential line of the parent path at its starting point.

Subordinary path growth. With a starting point and its initial growing direction determined, a child path can grow. The growth strategy of the subordinary paths is more complicated than the seed path. It has an additional requirement of following the shape of a parent path so that the working area is probably covered. They still need to consider climbing and turning costs, so the cost function for selecting the best driving direction becomes Eq. 6 in the slopy areas or Eq. 7 in the flat areas. The first two terms in Eq. 6 are the same as the seed path growth, and the third term reflects how much it follows a desirable shape of its parent path. We design a tangent function that has a dramatically increasing value if the shape deviation becomes large. In this way, it becomes a solid restriction to avoid significant gaps or overlaps. When the path grows in flat areas, the influence from the terrain

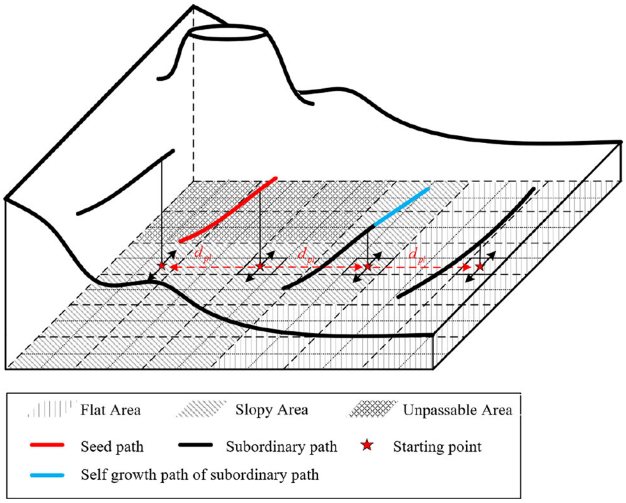


Fig. 7 Starting status determination for subordinary paths

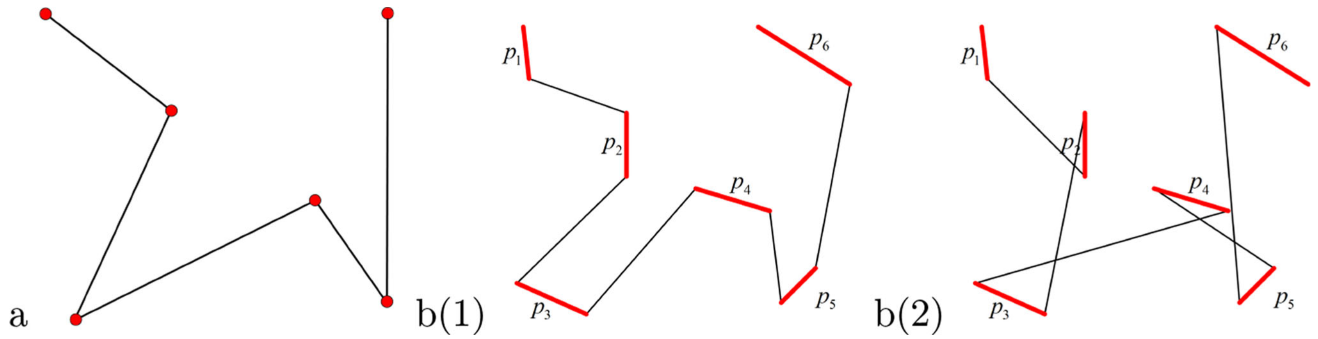


Fig. 8 The comparison of different path connection problems. (a) the traditional point-to-point traveling salesman problem; (b1) the path connection with the consideration of the end-point sequence; (b2) the path connection without considering the end-point sequence

slope is negligible. The first term in Eq. 6 is dropped out, which gives a cost function as in Eq. 7.

$$J_{s2}(k) = \frac{tan(\alpha_p^k)}{tan(\bar{\alpha})} + \lambda_{s1} * tan(\Delta\omega^k) + \beta_{s1} * tan(|\frac{d(k) - d_{pl}}{d_{pl}}| * \frac{\pi}{2}) \tag{6}$$

$$J_{s3}(k) = \lambda_{s1} * tan(\Delta\omega^k) + \beta_{s2} * tan(|\frac{d(k) - d_{pl}}{d_{pl}}| * \frac{\pi}{2}) \tag{7}$$

where β_{s1} and β_{s2} are weight factors for the shape-mimicking term, large values of which signify a more pronounced emphasis on conforming the shape of the seed path to its parent path. $d(k)$ represents the distance between the expected cell of the child path and its parent path. It is worth noting that, to maintain consistency in form and value range for the three terms, we incorporate a tangential function for the newly introduced shape-conforming demand for seed path growth value functions.

Mixed growth mode. There is one more condition to address. When a child path grows out of the range of its parent path, as the blue path in Fig. 7, it enters the auto-growth mode. In other words, it locally becomes a seed curve without any reference to follow. In this case, the growing law becomes the same as a seed path. Similar to seed path growth, the subordinary path stops when it enters Ω.

3.4 Optimal path connection with a modified GA

Traditionally, the CPP problem connecting paths from different subsections is simplified as the typical TSP problem [23, 24], as shown in Fig. 8. The traditional point-to-point TSP is shown in Fig. 8(a). It only needs to consider the connecting sequence of the waypoints. Using TSP to solve the CPP introduces an extra problem: the connecting sequence of the path endpoints also influences the traveling cost to a large extent. As shown in Fig. 8(b1) and Fig. 8(b2), if the endpoint sequence of the path connection is not optimized, it could cause unnecessary detours. It did not largely affect the coverage path length in previous research since there were only a few subsections in the whole working area. The increased traveling cost caused by the improper choice of endpoint sequences is bearable. However, in the technical path of this paper, there is no need to partition the working area. Instead, many individual paths autonomously grow following the rules in Section 3. The number of paths largely increases. As a result, the total travel cost would dramatically rise if the endpoint sequence was not carefully arranged. Therefore, in this paper, we need to simultaneously address the path connection sequence and the endpoint connection sequence, which is called a path connection traveling salesman problem (PC-TSP).

We treat the PC-TSP as a two-layer problem. The upper layer solves a common TSP problem using GA. The lower layer uses a dynamical programming method to find the best endpoint connection sequence for a given chromosome in the upper layer, in other words, a particular path sequence.

In the upper layer, a gene S^p in the GA is defined as a path with two endpoints$\{S_1^p, S_2^p\}$, as shown in Fig. 9. p is the index of a gene in a chromosome. The two endpoints in a gene are denoted as S_1^p and S_2^p, respectively, according to their storage sequence in chromosomes. In this way, the GA mutation of the chromosome corresponds to a pure ordering problem for the genes. We use the order crossover OX1 of Davis [30] as the crossover operator to perform the information exchange between chromosome individuals.

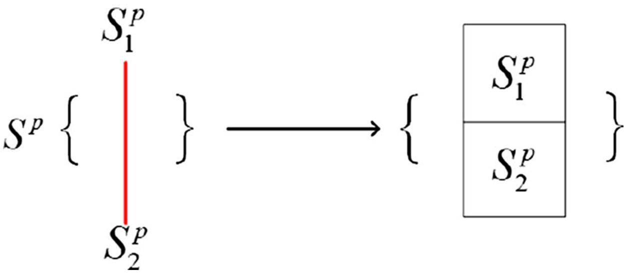


Fig. 9 A genetic node

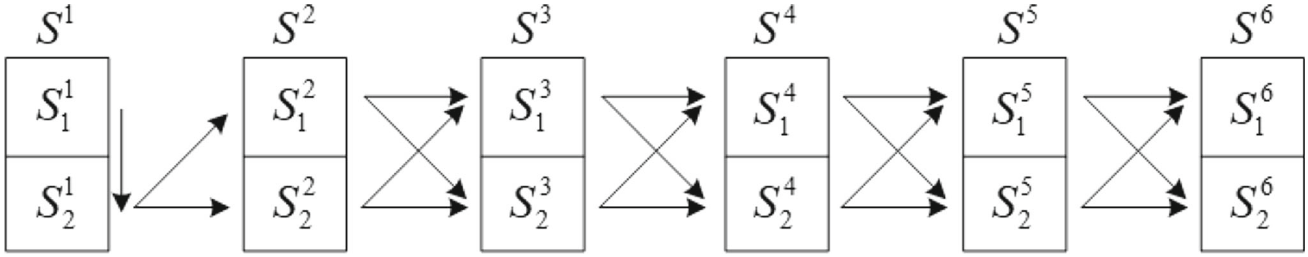


Fig. 10 The connection of the genetic nodes with all possible endpoint sequences

On the other hand, the lower layer calculates the fitness value for each chromosome. The problem can be defined as finding the shortest path from the first gene to the last one with a designated starting point S_1^1, among all possible endpoint connection sequences depicted in Fig. 10. It is a typical dynamic programming problem, and the corresponding state transition equations are shown in Eq. 8 and Eq. 9. The two states for the p^{th} gene connection are defined as $d_p[1, p]$ and $d_p[2, p]$. They represent the travel cost of connecting S_1^p or S_2^p with the $p - 1$ genes ahead. $dis(\cdot, \cdot)$ represents the distance between two endpoints. With the state equations defined above, dynamic programming can be applied to find the endpoint sequence with the smallest travel cost. The fitness of the chromosome can be calculated by Eq. 10.

$$\begin{aligned} d_p[1, p] = \min(&dis(S_1^p, S_1^{p-1}) \\ &+d_p[2, p-1], dis(S_1^p, S_2^{p-1}) + d_p[1, p-1]) \end{aligned} \tag{8}$$

$$\begin{aligned} d_p[2, p] = \min(&dis(S_2^p, S_1^{p-1}) \\ &+d_p[2, p-1], dis(S_2^p, S_2^{p-1}) + d_p[1, p-1]) \end{aligned} \tag{9}$$

$$F = \min(d_p[1, M], d_p[2, M]) \tag{10}$$

where M is the chromosome length, and i is the index of the chromosome.

Combining the algorithms above, we can obtain a modified GA solving the PC-TSP problem. The algorithm is formally presented in Algorithm 1.

Algorithm 1 The modified GA for the PC-TSP problem

```
Require: the population size $P_n$, the probability of crossover $P_c$, the
    probability of mutation $P_m$, the generation of current evolution $t$,
    the generation of terminating evolution $G$, the threshold of fitness
    function $T_f$ and the population fitness $F_m(0)$;
Ensure: Path sequence with endpoint connections
initial population $P_0$
while the population fitness $F_m(t) < T_f \vee$ the reproductive generation $t < G$ do
    $t = t + 1$, $F_m(t) = \infty$
    for $i = 1 \rightarrow P_n$ do
        $d_p[0, 0] = 0, d_p[1, 0] = 0$
        for $p = 1 \rightarrow M$ do
            calculate $d_p[1, p]$ and $d_p[2, p]$ using Eq. 8 and Eq. 9
        end for
        $F(i) = min(d_p[1, M], d_p[2, M])$
        $F_m(t) = min(F(i), F_m(t))$
    end for
    for $i = 1 \rightarrow P_n$ do
        Select operation to the current population $P_t$
    end for
    if $random(0, 1) < P_c$ then
        for $i = 1 \rightarrow P_n/2$ do
            Ordered crossover operation to $P_t$
        end for
    end if
    if $random(0, 1) < P_m$ then
        for $i = 1 \rightarrow P_n$ do
            Mutation operation to $P_t$
        end for
    end if
    $P_{t+1} = P_t$
end while
```

4 Experimental Results and Discussion

We conducted three experiments using different test datasets sourced from various origins to showcase the algorithm's effectiveness across diverse terrain conditions. In the subsequent sections, we present the outcomes of applying our method to three distinct maps. Subsequently, we perform a comparative analysis, evaluating the performance of the TSA-CPP method in contrast to prior methodologies.

4.1 Experiment 1-open-source data

In this experiment, the test data is displayed in Fig. 11, representing open topography data of an agricultural field obtained through aerial LiDAR in Nelson, New Zealand [31]. The technical parameters for the tests are detailed in Table 1, encompassing vital dimensions of the vehicles, map areas, and parameter values pertinent to the TSA-CPP algorithm. The vehicle size is akin to that of a tractor, highlighting its suitability for agricultural applications. The weight factors in Eq. 5, Eq. 6, and Eq. 7, represented by λ_{s1}, β_{s1}, and β_{s2},

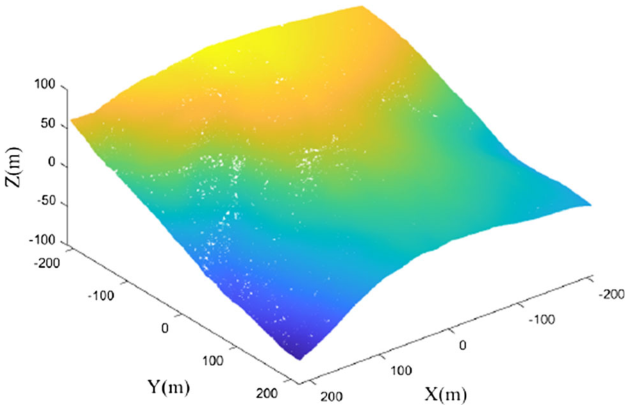


Fig. 11 The open source data in Nelson, New Zealand

respectively, are set to 0.5, 4, and 7. We conducted sensitivity analyses and fine-tuned these parameters to strike a balance that optimally aligns with the objectives. This set of weight factors proves to be robust against variations in terrain conditions and remains consistent across all three experiments. It achieves a balanced performance in terms of path smoothness, path shape conformation, and terrain shape adaptation, as demonstrated in the following three cases.

The traversability maps for the test field are presented in Fig. 12. The left map illustrates the initial traversability conditions, identifying four terrain conditions: the border area of the field, the region with turning-over risk, the uneven area posing chassis collision risk, and the traversable area-the only safe condition for the specified vehicle to navigate. The right image displays the terrain classification for the CPP problem based on the principles outlined in Section 3.1. The CPP prioritizes avoiding entry into unpassable areas for enhanced driving safety. Flat areas are treated as a 2D case, while sloped areas employ a path-growing algorithm with terrain-adaptive functions. The limited presence of flat areas in the open-source data underscores the need for terrain shape adaptation when planning an energy-efficient path. In regions where the slope is excessively steep, they are marked as unpassable areas unsuitable for the vehicle to work on.

Table 1 Parameter list for the open-source and simulation data

Parameter	Value	Parameter	Value
vehicle size/$m * m$	$2.5 * 2$	wheel Radius/m	0.3
chassis Height/m	0.4	weight/kg	1500
$\bar{\alpha}$	15	field area/m^2	$420 * 420$
cell side length/m	3	l_{ss}	6
N	7	$\alpha^s/\circ$	8
d_{pl}/m	18	λ_{s1}	0.5
β_{s1}	4	β_{s2}	7

The illustration of the path auto-growth process is presented in Fig. 13. In the context of open-source data, paths exhibit a tendency toward uniform growth directions due to gradual slope changes. All paths halt before entering hazardous zones. However, small portions of areas at the boundaries between safe and hazardous zones remain uncovered. This conservative approach is implemented to mitigate potential operational risks. Despite this, the path-growth algorithm effectively covers the majority of irregular terrain. Path intervals are generally maintained consistently, and the path shape aligns with the contour lines. There are a total of 35 separate paths generated.

PC-TSP in Section 3.4 is applied to connect the paths with optimal sequences. The resulting connections are shown in Fig. 14. Since most of the paths have a close-to-parallel pattern for the open source data, the algorithm finds the neighbor paths to be connected in sequence. There are no apparent twists or detours.

4.2 Experiment 2 - simulation data

In this experiment, we employ mathematical functions to model the topography depicted in Fig.15. The terrain is characterized by a steep peak and a valley, illustrating scenarios where the landscape exhibits abrupt elevation changes within a confined space. Additionally, smaller peaks and valleys are dispersed throughout the surrounding region, contributing to the complexity of the terrain. The experimental parameters are identical to those of Experiment 1, outlined in Table 1.

The traversability maps for the simulation data are depicted in Fig. 16. The initial traversability analysis reveals that the terrain poses challenges primarily on the steep slopes near the peak and valley, where passage is hazardous. Consequently, two unpassable regions are identified on these slopes, with the majority of the left area characterized as slopy areas. Flat areas are dispersed around the periphery, while the central region exhibits an intricate interplay of various terrain conditions.

The 53 auto-growing paths on the simulation data are illustrated in Fig. 17, with little surface left uncovered. These paths maintain a contour-like configuration, efficiently covering the majority of passable areas while circumventing potential driving hazards. The depiction distinctly reveals six peaks and valleys, with the path automatically forming a spiral pattern centered around the highest and lowest points, which is close to the theoretical energy-optimal design in [10]. This design is presumably advantageous for conserving energy during movement, minimizing the need for repetitive ascent and descent actions.

Despite the intricate distribution pattern of paths, as depicted in Fig. 18, the PC-TSP adeptly identifies an optimized and rational connection sequence for continuous

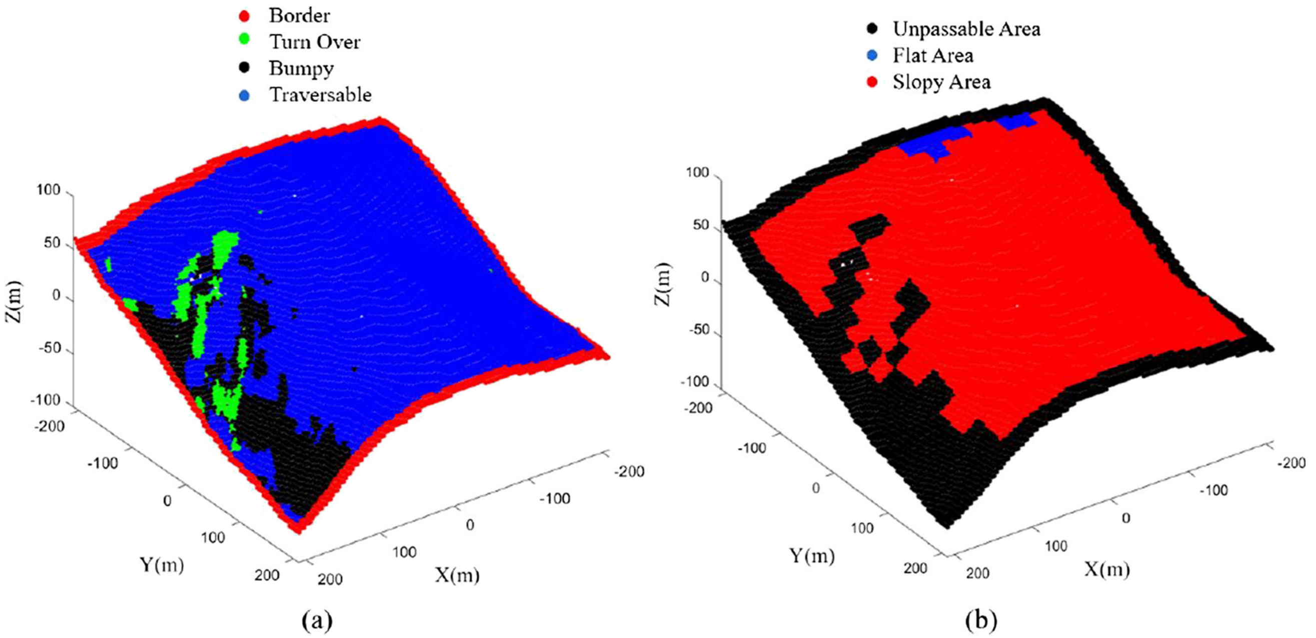


Fig. 12 The traversability map and region classification for the open-source data

operations. The connection of the paths successfully mitigates long jumps and eliminates unnecessary detours.

4.3 Experiment 3 - experimental data

In this experiment, we utilize an actual Unmanned Ground Vehicle (UGV) for data collection and conduct path planning tests. The UGV is depicted in Fig. 19(c). The vehicle is equipped with a dense LiDAR system, which generates a point cloud map of the experimental scene shown in Fig. 19(b). The presence of shrubs and obstacles scattered throughout the scene adds complexity to the driving conditions, serving as a valuable test for the robustness of the TSA-CPP. The acquired data, illustrated in Fig. 19(a), reveals a slope with distributed obstacles. At the summit of the slope, shrubs pose obstacles that impede the UGV's passage. The parameter list for the experimental data is presented in Table 2. The vehicle-related parameters are set to be consistent with the UGV.

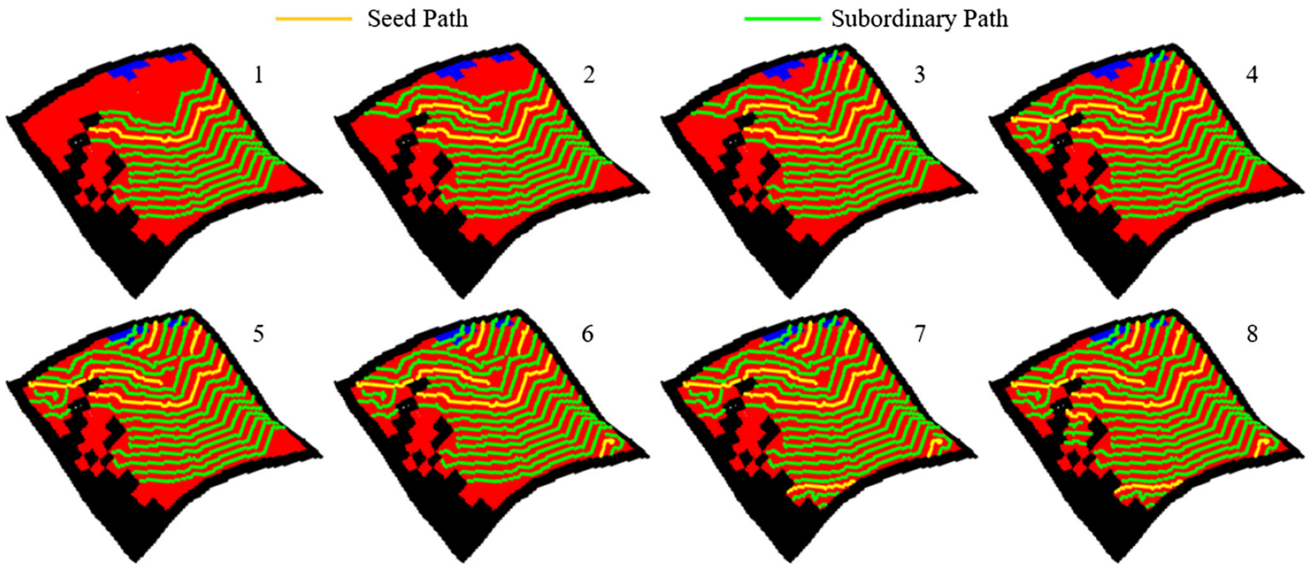
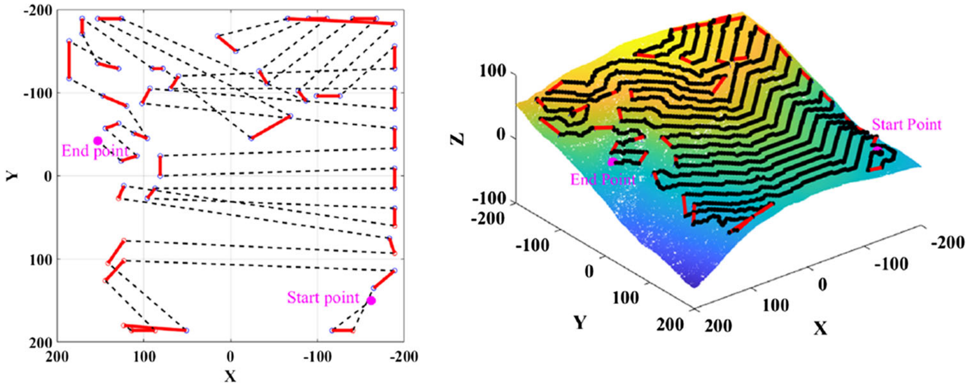


Fig. 13 The process of path growth for open-source data



(a)

(b)

Fig. 14 The path connection results for the open-source data. (a) The connection sequence from the designated start point to the end point, projected to the xy-plane. (b) The connected continuous coverage path on the 3D terrain surface, where the black curves are the auto-growing path and the red curves are the connection between the path segments

The traversability maps for the experimental data are illustrated in Fig. 20. The majority of passable areas are characterized as flat terrain, with only a few sloped spots scattered throughout. This is attributed to the relatively gentle and smooth slopes. Several obstacle zones are embedded within extensive passable areas, and in the upper-left corner, a passable region becomes inaccessible due to obstacle barriers.

In this map, a total of 23 paths have been generated from 5 seed paths (Fig. 21). As evident, the majority of areas exhibit a flat slope, resulting in parallel growth of most paths. The only exceptions occur in the upper-left region, where a seed path extends into a sloped area, forming a small circular path. Subsequent paths then emulate this pattern, resulting in layered curves. Similar to the two aforementioned experiments, the PC-TSP algorithm identifies the shortest connection path, forming a continuous operational route from the starting point, as illustrated in Fig. 22.

4.4 Comparative study

To compare the performance of the TSA-CPP method with previous research, we use three recent works on CPP as the contrast reference. The first one is the optimal Zig-zag pattern planning for 3D terrains [24]. The optimal Zig-zag algorithm finds the driving angle of the Zig-zag lines with the best energy consumption. Though without the terrain-shape adaptive property, it theoretically provides the most energy-saving driving direction within all possible fixed direction choices. The second approach is the GBNN algorithm [19], which is not explicitly tailored for 3D terrains but an innovative bio-inspired method dealing with complex 2D maps. In contrast, STC is a widely used and sophisticated method [16] that has received much research attention [32, 33]. GBNN and STC are both grid-based methods, where GBNN uses a grid map with the same cell edge length as the path shift distance d_{pl} and STC requires a cell edge length as $2d_{pl}$.

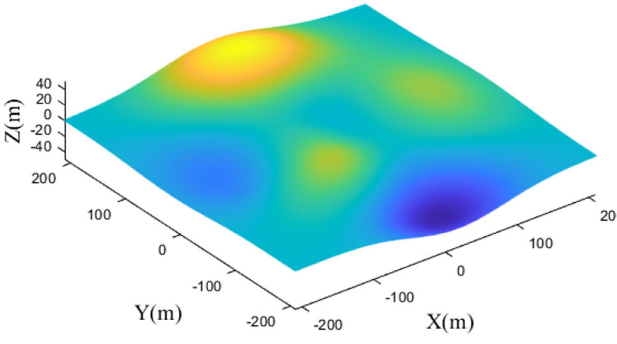


Fig. 15 The simulation data

For an equitable comparison, we maintain a consistent path shift distance (d_{pl}) for all four methods and generate functional grid maps incorporating designated unpassable regions through our traversability analysis. This ensures environmental uniformity across the three methods employed for comparative analysis. The methodologies were executed on a personal computer equipped with an Intel(R) Xeon(R) W-2145 3.70GHz CPU and 128GB of memory. Following the model outlined in [34], the rolling resistance coefficient is estimated at 0.023, assuming a rubber material for the tires and a muddy terrain environment.

The visual comparison of the four methods is shown in Fig. 23. The path produced by the TSA method exhibits distinctive features when compared to the other three methods, showcasing contour-shaped curves as opposed to straight lines. The initial Zig-zag path is truncated by unpassable areas, prompting the application of the GA method in this paper for effective path reconnection. The GBNN method demonstrates relatively limited proficiency in generating obstacle-free path connections, particularly evident in the experimental map where the path segment crosses unpassable regions. Furthermore, downsampling of the original grid is necessary for GBNN to achieve a cell edge size equivalent to d_{pl}, leading to some uncovered passable areas. The situation is exacerbated in STC as it mandates a cell edge size of $2d_{pl}$, resulting in a lower resolution and a higher prevalence of uncovered areas.

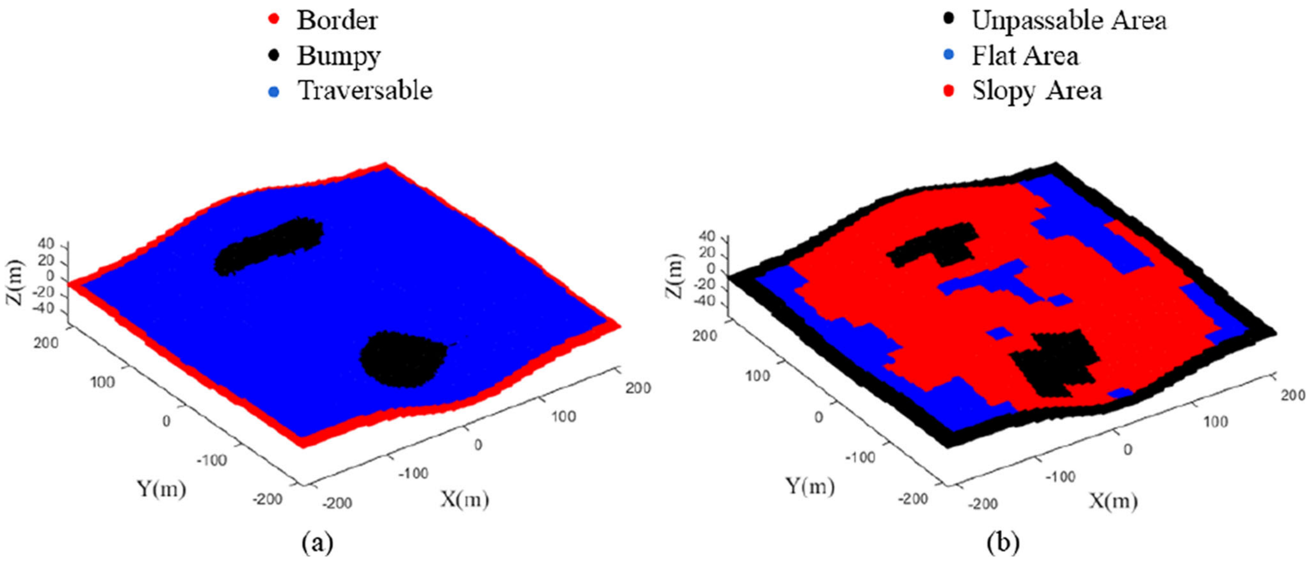


Fig. 16 The traversability map and region classification for the simulation data

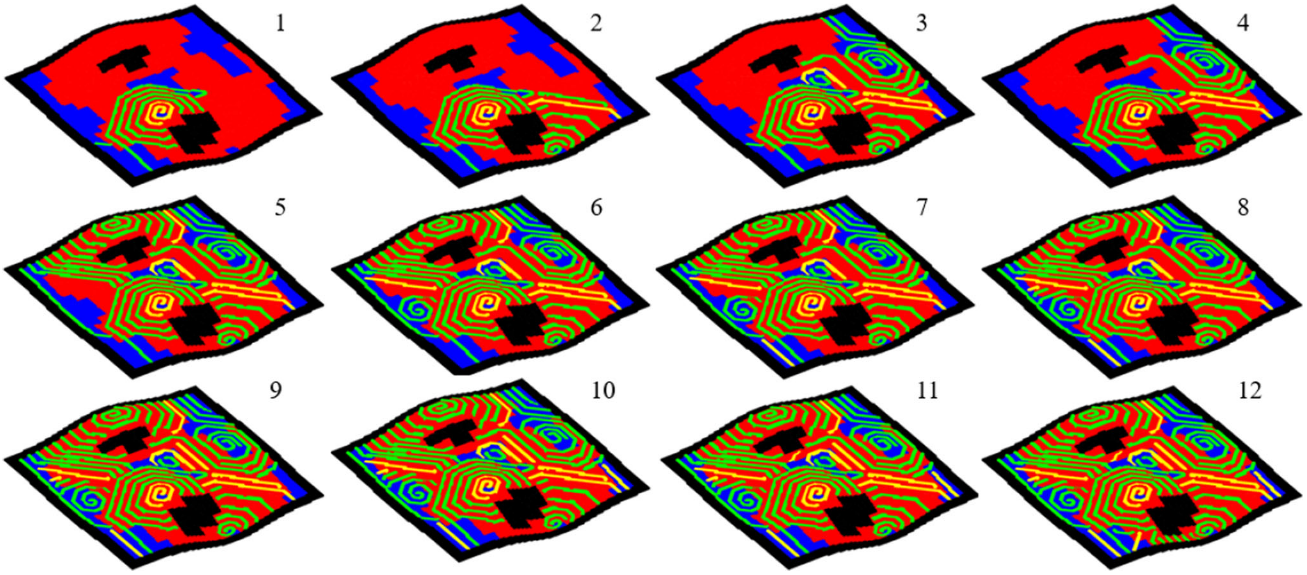
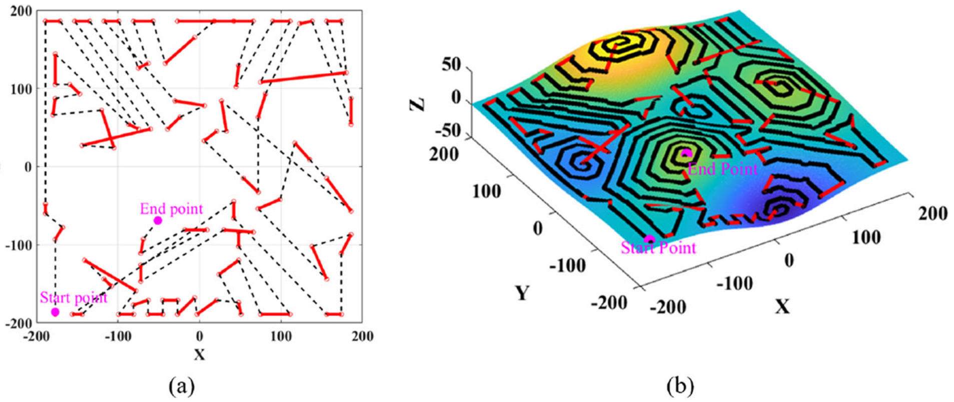


Fig. 17 The process of path growth for simulation data

Fig. 18 The path connection results for the simulation data



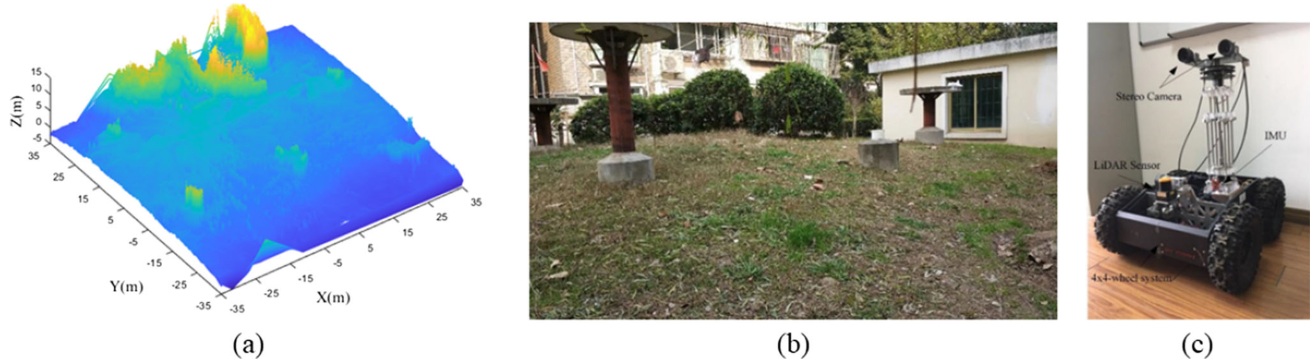


Fig. 19 The experimental setup and data. (a) the experimental data, (b) the experimental scene, and (c) the UGV used for data collection and tests

The climbing energy, path length, and coverage rate of the four algorithms are detailed in Table 3. The TSA algorithm exhibits a noticeable advantage in reducing climbing energy. Across all three maps, TSA demands the least climbing energy and attains the highest coverage rate on the third map, ranking second on the first two maps (with marginal differences of 0.4% and 0.5%, slightly trailing the Zig-zag method). Its climbing energy is 72.3%, 52%, and 91.8% of the other three methods on average, respectively, while achieving average coverage rates 0.7%, 2.4%, and 3% higher. In other words, TSA achieves the lowest energy assumption while maintaining the largest map coverage.

As for the path length, TSA is the longest for the first two maps but the shortest for the third one. TSA exhibits a similar path length to the Zig-zag method for the first two maps, which is reasonable given their comparable coverage rates. STC has the shortest path for these maps due to its low coverage rate. The high coverage rate for the third map illustrates that TSA has advantages in handling maps with complex obstacle distributions.

Additionally, we conducted a comparison of their computational time costs. The Zig-zag method consistently incurs a time cost of around 6 seconds across all three maps. STC exhibits the overall lowest time cost, ranging from 3 to 4 seconds. The time cost of GBNN is highly dependent on the map size, with a larger value exceeding 17 seconds for the first two maps and a considerably smaller one of about 4 seconds for the third map. In comparison, TSA's processing time is intermediate and not only correlates with map size but also with path complexity. On Map 1, characterized by relatively simple terrain variations, TSA's processing time is similar to Zig-zag. However, on Map 2, featuring intricate terrain shape variation, the processing time increases to 14.5 seconds. In contrast, on Map 3, with relatively gentle terrain variations and a smaller map size, TSA incurs the shortest processing time among all methods, merely 2.7 seconds.

Table 2 Parameter list for the experimental data

Parameter	Value	Parameter	Value
vehicle size/$m * m$	$0.5 * 0.5$	wheel Radius/m	0.1
chassis Height/m	0.15	weight/kg	20
$\bar{\alpha}$	15	field area/m^2	$70 * 70$
cell side length/m	0.5	l_{ss}	6
N	7	$\alpha^s/^\circ$	8
d_{pl}/m	3	λ_{s1}	0.5
β_{s1}	4	β_{s2}	7

5 Conclusion

Coverage path planning on uneven 3D terrains remains an ongoing challenge. As highlighted in this article, the path should ideally adapt to the shape of the terrain, primarily following elevation contours to optimize energy consumption during climbing. In this work, we present a novel terrain-shape-adaptive method that introduces innovations in three key aspects. Firstly, it utilizes traversability analysis to generate paths, effectively integrating obstacle avoidance and coverage path planning into a unified process. This ensures that the planned paths prioritize driving safety and avoid hazardous areas. Secondly, rather than adhering to a predetermined policy, the algorithm autonomously allows paths to grow on the terrain surface, facilitating terrain shape adaptation. Lastly, by transforming the traditional Traveling Salesman Problem (TSP) into a Path Connection TSP (PC-TSP), we propose an improved GA that determines the optimal path connection sequence with the lowest travel cost.

The algorithm demonstrates the ability to generate TSP paths based on point cloud maps. We evaluated its performance using three distinct datasets, and the results indicate its applicability across various terrain conditions. The generated coverage paths not only adhere to terrain contours on slopes but also successfully avoid all hazardous areas. The improved GA effectively makes path connection decisions

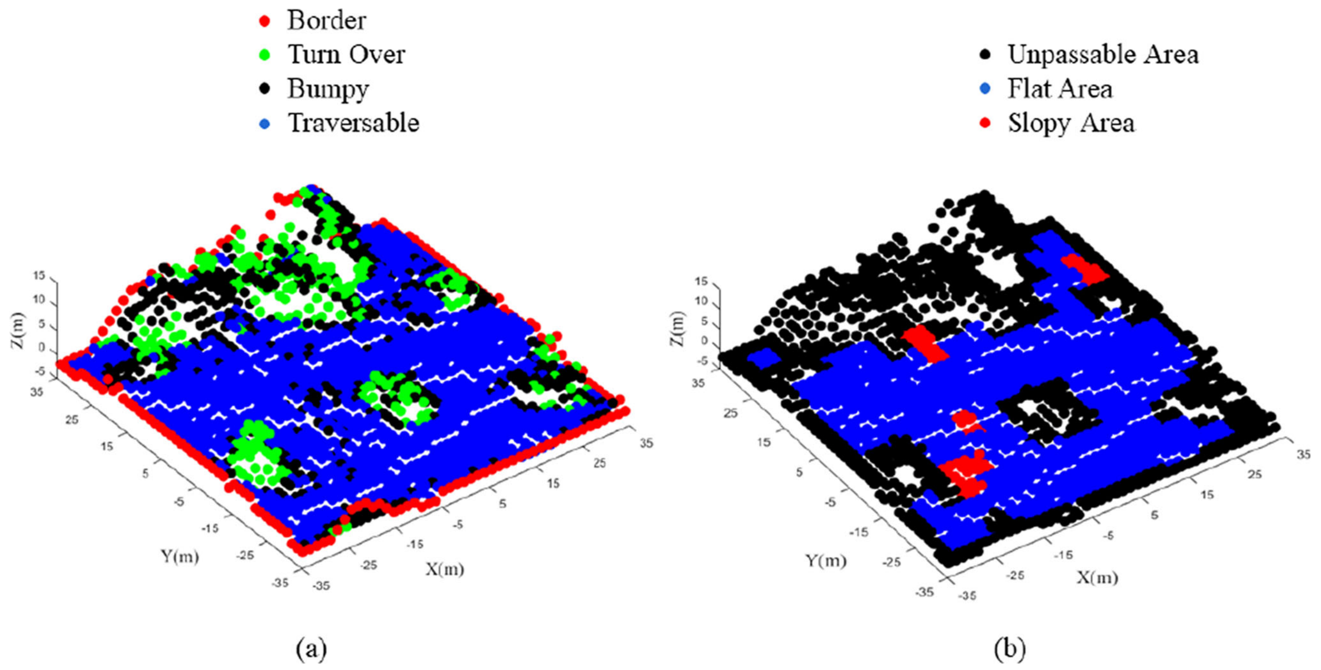


Fig. 20 The traversability map and region classification for the experimental data

with minimal travel costs. A comparison with three other previous works revealed a significant reduction in climbing energy ranging from 8.2% to 48% with relatively high coverage rates.

In conclusion, this work is well-suited for ground vehicle applications involving irregular terrain surfaces where energy consumption is a crucial consideration. It represents a significant advancement in 3D CPP by autonomously generating terrain-adaptive paths and facilitating hazard avoidance. It is important to note that the proposed method requires a preliminary analysis of terrain traversability, which may entail additional computational costs. The traversability map needs to have a grid format and contain the quantification of driving safety in different directions. This enables the map to capture the local terrain shape and guide the growth of a path that adapts to the terrain. However, considering that traversability

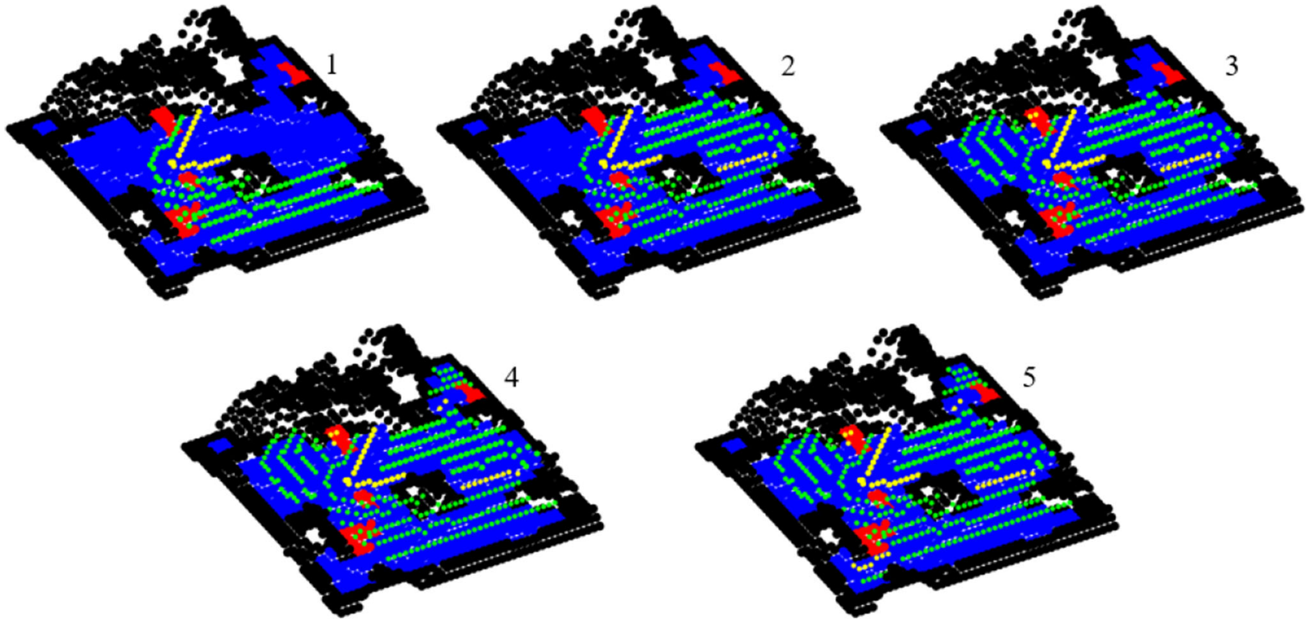


Fig. 21 The process of path growth for experimental data

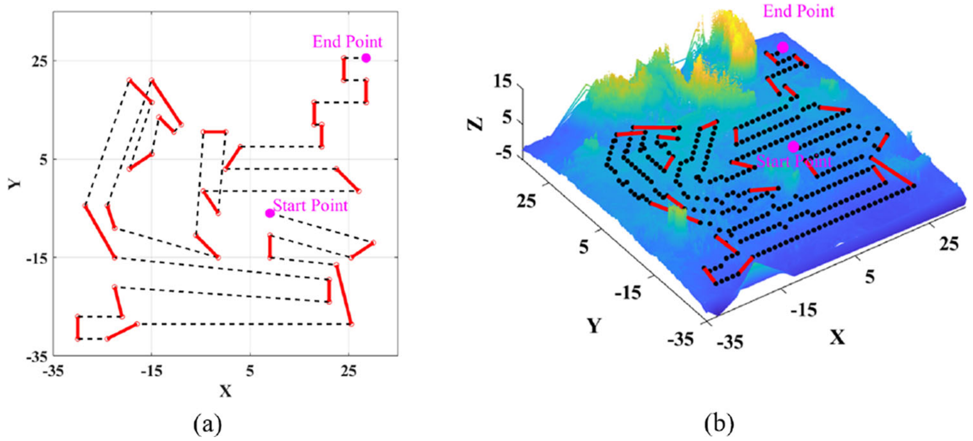


Fig. 22 The path connection results for the experimental data

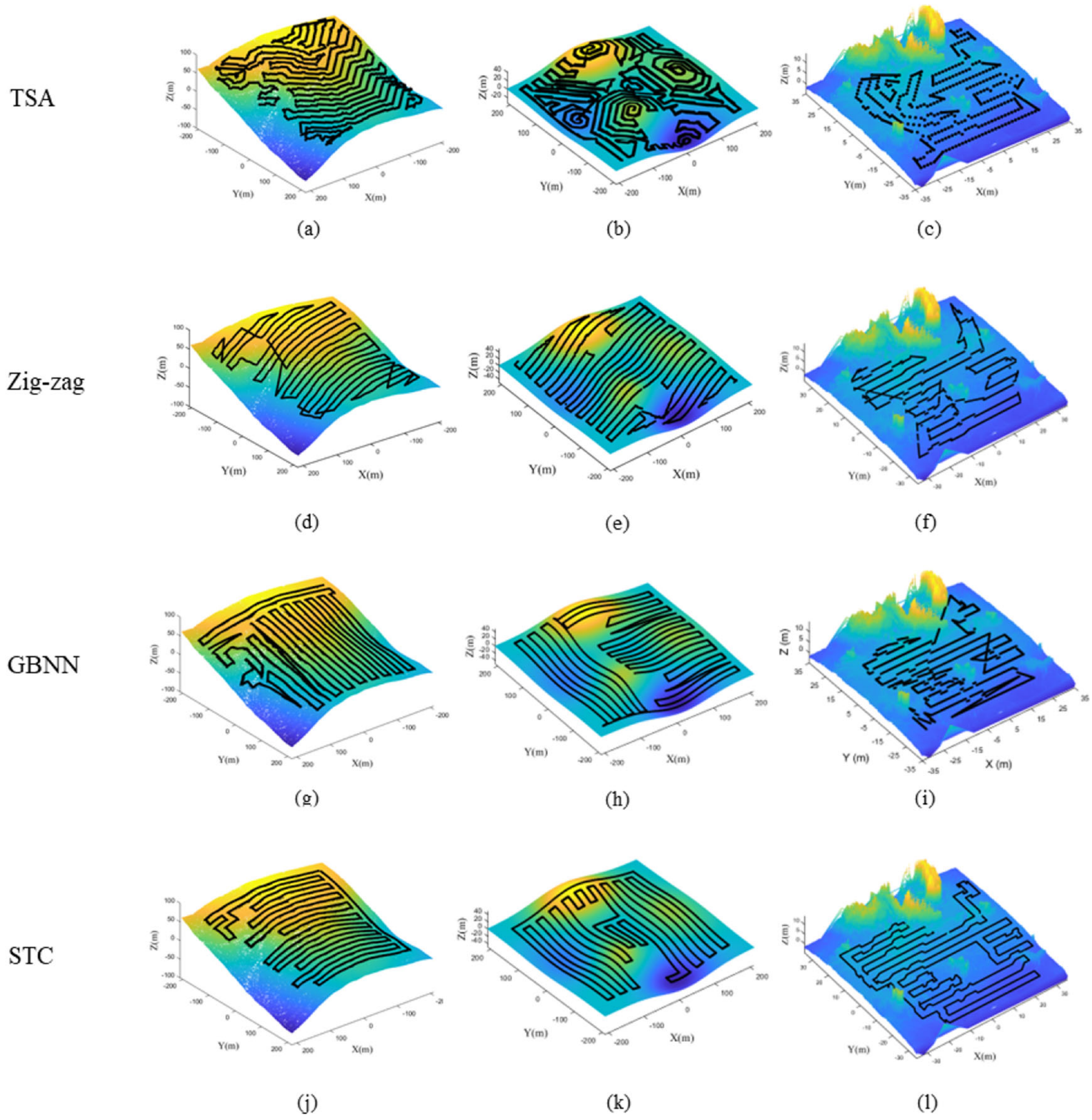


Fig. 23 The comparison of the CPP results. The first row shows the paths by our TSA method, and the following three rows are by Zig-zag, GBNN, and STC methods, respectively

Table 3 Comparison of the CPP performance

Algorithm	Metric	Open Source Data	Simulation Data	Experimental Data
TSA	climb energy/KJ	**8.23E+06**	**4.80E+06**	**7.20E+05**
	path length/m	7.96E+03	8.88E+03	**6.34E+02**
	coverage rate/%	98.2	99.4	**96.1**
	time/s	6.5	14.5	**2.7**
Zig-zag	climb energy/KJ	1.20E+07	1.07E+07	7.60E+05
	path length/m	7.94E+03	8.50E+03	7.02E+02
	coverage rate/%	**98.6**	**99.9**	93.3
	time/s	6.1	6.0	6.0
GBNN	climb energy/KJ	1.26E+07	9.67E+06	7.97E+05
	path length/m	7.82E+03	8.11E+03	7.71E+02
	coverage rate/%	98.2	97.8	91.6
	time/s	17.4	17.3	4.2
STC	climb energy/KJ	9.90E+06	7.82E+06	7.97E+05
	path length/m	**6.46E+03**	**7.09E+03**	8.99E+02
	coverage rate/%	95.6	93.4	94.4
	time/s	**3.8**	**3.1**	3.7

analysis is generally necessary for obstacle avoidance path planning on irregular 3D terrains, this requirement is not a significant drawback from a systematic perspective. Moreover, the algorithm's auto-growth property makes it a viable candidate for addressing multi-robot CPP, as it simplifies the problem by transforming it from complex terrain segmentation to a multi-TSP setting.

Author Contributions [Wenwei Qiu] and [Dacheng Zhou] are declared as joint first authors, signifying their equal contribution to this research in coding, data collection, analysis and paper writing. [Wenbo Hui] contributed to the construction and comparative analysis of two contrasting algorithms, GBNN and STC. [Afimbo Reuben Kwabena] and [Yubo Xing] played a key role in the interpretation and analysis of the data. The algorithm employed for the traversability analysis in this study was developed by [Quan Li]. [Yi Qian] took charge of figure design and visualization in this research. [Huayan Pu] provided support in the design of algorithms and experiments for this study. [Yangmin Xie] propose the original idea for the TSA-CPP algorithm and in charge of the paper writing.

Funding This research was funded by the National Natural Science Foundation of China, grant number 62173220, and the Shanghai Science and Technology Innovation Action Plan, grant number 19DZ1207305.

Declarations

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

Ethics approval Not applicable.

Consent to participate Informed consent was obtained from all individual participants included in the study.

Consent to publish Not applicable.

References

1. Yang, S., Luo, C.: A neural network approach to complete coverage path planning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). **34**(1), 718–724 (2004)
2. Hassan, M., Liu, D.: Ppcpp: A predator-prey-based approach to adaptive coverage path planning. IEEE Trans. Rob. **36**(1), 284–301 (2020)
3. Hameed, I.A.: Coverage path planning software for autonomous robotic lawn mower using dubins' curve. In 2017 IEEE International Conference on Real-time Computing and Robotics (RCAR). pp. 517–522 (2017)
4. Jin, J., Tang, L.: Coverage path planning on three-dimensional terrain for arable farming. Journal of field robotic. **28**(3), 424–440 (2011)
5. Hameed, I.A.: Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain. Journal of Intelligent & Robotic Systems. **74**(3), 965–983 (2014)
6. Zhao, J., Zhu, L., Liu, G., iu, G., Han, Z.: A modified genetic algorithm for global path planning of searching robot in mine disasters. In 2009 International Conference on Mechatronics and Automation. pp. 4936–4940 (2009)
7. Park, S.J., Yeu, T.K., Yoon, S.M., Hong, S., Sung, K.Y.: A study of sweeping coverage path planning method for deep-sea manganese

nodule mining robot. In OCEANS'11 MTS/IEEE KONA. pp. 1–5 (2011)

8. Sasaki, T., Otsu, K., Thakker, R., Haesaert, S., Agha-mohammadi, A.: Where to Map? Iterative Rover-Copter Path Planning for Mars Exploration. IEEE Robotics and Automation Letters. **5**(2), 2123–2130 (2020). https://doi.org/10.1109/LRA.2020.2970650
9. Ye, X., Luo, L., Hou, L., Duan, Y., Wu, Y.: Laser ablation manipulator coverage path planning method based on an improved ant colony algorithm. Applied Sciences. **10**(23) (2020)
10. Wu, C., Dai, C., Gong, X., Liu, Y.J., Wang, J., Gu, X.D., Wang, C.C.L.: Energy-Efficient Coverage Path Planning for General Terrain Surfaces. IEEE Robotics and Automation Letters. **4**(3), 2584–2591 (2019). https://doi.org/10.1109/LRA.2019.2899920
11. Cabreira, T.M., Brisolara, L.B., Ferreira, P.R., Jr.: Survey on Coverage Path Planning with Unmanned Aerial Vehicles. Drones. **3**(1), 4 (2019). https://doi.org/10.3390/drones3010004
12. Tan, C.S., Mohd-Mokhtar, R., Arshad, M.R.: A Comprehensive Review of Coverage Path Planning in Robotics Using Classical and Heuristic Algorithms. In IEEE Access. **9**, 119310–119342 (2021). https://doi.org/10.1109/ACCESS.2021.31081
13. Oksanen, T., Visala, A.: Coverage path planning algorithms for agricultural field machines. Journal of field robotics. **26**(8), 651–668 (2009)
14. Jin, J., Tang, L.: Optimal coverage path planning for arable farming on 2d surfaces. Trans. ASABE **53**(1), 283–295 (2010)
15. Le, A.V., Prabakaran, V., Sivanantham, V., Mohan, R.E.: Modified A-Star Algorithm for Efficient Coverage Path Planning in Tetris Inspired Self-Reconfigurable Robot with Integrated Laser Sensor. Sensors. **18**, 2585 (2018). https://doi.org/10.3390/s18082585
16. Kapoutsis, A.C., Chatzichristofis, S.A., Kosmatopoulos, E.B.: DARP: Divide Areas Algorithm for Optimal Multi-Robot Coverage Path Planning. Journal of Intelligent & Robotic Systems. **86**, 663–680 (2017)
17. Zhang, J., Zu, P., Zhou, M.: A Herd Foraging-Based Adaptive Coverage Path Planning in Unbounded Environments. In 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2062–2066 (2022). https://doi.org/10.1109/SMC53654.2022.9945206
18. Tran, V.P., Garratt, M.A., Kasmarik, K.E., Anavatti, S.G., Abpeikar, S.: Frontier-led swarming: Robust multi-robot coverage of unknown environments. Swarm Evol. Comput. **75**, 101171 (2021)
19. Zhu, D., Tian, C., Sun, B., Luo, C.: Complete Coverage Path Planning of Autonomous Underwater Vehicle Based on GBNN Algorithm. Journal of Intelligent & Robotic Systems. **94**, 237–249 (2019)
20. Wang, Y., He, Z., Cao, D., Ma, L., Li, K., Jia, L., Cui, Y.: Coverage path planning for kiwifruit picking robots based on deep reinforcement learning. Computers and Electronics in Agriculture. **205**, p. 107593 (2023), ISSN: 0168–1699. https://doi.org/10.1016/j.compag.2022.107593
21. Kyaw, P.T., Paing, A., Thu, T.T., Mohan, R.E., Le Vu, A., Veerajagadheswar, P.: Coverage Path Planning for Decomposition Reconfigurable Grid-Maps Using Deep Reinforcement Learning Based Travelling Salesman Problem. IEEE Access. **8**, 225945–225956 (2020). https://doi.org/10.1109/ACCESS.2020.3045027
22. Kumar, K., Kumar, N.: Region coverage-aware path planning for unmanned aerial vehicles: A systematic review. Phys. Commun. **59**(C), pp. 102073 (2023). https://doi.org/10.1016/j.phycom.2023.102073
23. Dogru, S., Marques, L.: Towards fully autonomous energy-efficient coverage path planning for autonomous mobile robots on 3d terrain. In 2015 European Conference on Mobile Robots (ECMR). pp. 1–6 (2015)
24. Shen, M., Wang, S., Wang, S., Su, Y.: Simulation study on coverage path planning of autonomous tasks in hilly farmland based on energy consumption model. Mathematical Problems in Engineering. **2020** (2020)
25. Apuroop, K.G.S., Le, A.V., Elara, M.R., Sheu, B.J.: Reinforcement learning-based complete area coverage path planning for a modified htrihex robot. Sensors. **21**(4) (2021)
26. Hameed, I.A., la Cour-Harbo, A., Osen, O.L.: Side-to-side 3d coverage path planning approach for agricultural robots to minimize skip/overlap areas between swaths. Robot. Auton. Syst. **76**, 36–45 (2016)
27. Zhao, H., Gu, F., Huang, Q.X., Garcia, J., Chen, Y., Tu, C., Benes, B., Zhang, H., Cohen-Or, D., Chen, B.: Connected fermat spirals for layered fabrication. ACM Transactions on Graphics (TOG). **35**(4), 1–10 (2016)
28. Tang, Y., Cai, J., Chen, M., Yan, X., Xie, Y.: An autonomous exploration algorithm using environment-robot interacted traversability analysis. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4885–4890 (2019)
29. Arena, P., Blanco, C.F., Noce, A.L., Taffara, S., Patanè, L.: Learning traversability map of different robotic platforms for unstructured terrains path planning. In 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (2020)
30. Deep, K., Mebrahtu, H.: New variations of order crossover for travelling salesman problem. International Journal of Combinatorial Optimization Problems and Informatics. **2**(1), 2–13 (2011)
31. Nelson city council, toitū te whenua land information new zealand (linz). nelson, new zealand 2021. collected by aerial surveys, distributed by opentopography and linz (2021). https://doi.org/10.5069/G9R78CD4
32. Lu, J., Zeng, B., Tang, J., Lam, T.L., Wen, J.: TMSTC*: A Path Planning Algorithm for Minimizing Turns in Multi-Robot Coverage. In IEEE Robotics and Automation Letters. **8**(8), 5275–5282 (2023). https://doi.org/10.1109/LRA.2023.3293319
33. Gao, G.Q., Xin, B.: A-STC: Auction-based Spanning Tree Coverage Algorithm for Motion Planning of Cooperative Robots. Frontiers of Information Technology & Electronic Engineering. **20**, 18–31 (2019). https://doi.org/10.1631/FITEE.1800551
34. Rakha, H., Lucic, I., Demarchi, S.H., et al.: Vehicle dynamics model for predicting maximum truck acceleration levels. J. Transp. Eng. **127**(5), 418–425 (2001)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Wenwei Qiu earned her B.S. in Intelligent Manufacturing Engineering from the School of Mechatronics Engineering and Automation at Shanghai University, Shanghai, China, in 2023. Currently, she is actively pursuing an M.S. degree at the Shanghai Key Laboratory for Robotics, Shanghai University. Her current research specialization encompasses auto-navigation techniques for unmanned systems and reinforcement learning.

Dacheng Zhou holds a Bachelor's degree in Mechanical Engineering from Nantong University, School of Mechanical Engineering, Nantong, China. He further obtained a Master's degree from the School of Mechatronics Engineering and Automation at Shanghai University, Shanghai, China. He mainly focuses on multi-agent autonomous navigation.

Wenbo Hui received his B.S degree from Shanghai University, Shanghai, China. He is currently pursuing an M.S degree at the Shanghai Key Laboratory for Robotics, Shanghai University. His main search interests include Multi-agent coverage path planning and cooperative control of multi-agent.

Afimbo Reuben Kwabena obtained a Bachelors degree in mechanical engineering and automation from China University of Petroleum(UPC) in 2024. Currently pursuing Masters Research in the same field at Shanghai University and researches mainly on robotic manipulation and path planning.

Yubo Xing who has obtained a Bachelor of Engineering degree from the University of Shanghai for Science and Technology, is currently a postgraduate student at the School of Mechanical Engineering and Automation, Shanghai University. He is engaged in research at the Shanghai Key Laboratory for Robotics. His primary research focus is on multi-view three-dimensional reconstruction technology and SLAM (Simultaneous Localization and Mapping) techniques.

Yi Qian received the B.S. in Intelligent Manufacturing Engineering from the School of Mechatronics Engineering and Automation, Shanghai University, Shanghai, China, in 2023. And she is currently pursuing a Master in Hong Kong. Her research is mainly focus on LiDAR-based autonomous UAVs.

Quan Li completed his undergraduate studies at Henan Polytechnic University, graduating in 2018 from the School of Mechanical and Power Engineering with a major in Mechanical Design and Manufacturing and Automation. He received a master's degree in mechatronics engineering from Shanghai University in 2021. His research interests include path planning and SLAM technology for autonomous robots.

Huayan Pu received the M.Sc. and Ph.D. degrees in mechatronics engineering from Huazhong University of Science and Technology (Wuhan, China) in 2007 and 2011, respectively. Currently, she is a Professor at Shanghai University (Shanghai, China). Her current research interests include vibration controlling and robotics. Dr. Pu was awarded the best paper in biomimetics at the 2013 IEEE International Conferences on Robotics and Biomimetics. She was also nominated as the best conference paper finalist at the 2012 IEEE International Conference on Robotics and Biomimetics.

Yangmin Xie received the B.S. and M.S. degrees from the School of Mechanical Engineering, Bei-hang University, Beijing, China, and the Ph.D. degree in mechanical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA. She is currently a Professor with the School of Mechatronics Engineering and Automation, Shanghai University, Shanghai, China. Her current research interests include intelligent sensing and control of mobile robotics. She mainly focuses on environmental modeling, machine vision and autonavigation techniques for unmanned systems.

Authors and Affiliations

Wenwei Qiu[1] · Dacheng Zhou[1] · Wenbo Hui[1] · Afimbo Reuben Kwabena[1] · Yubo Xing[1] · Yi Qian[1] · Quan Li[1] · Huayan Pu[1] · Yangmin Xie[1,2]

Wenwei Qiu
qiuwenwei@shu.edu.cn

Dacheng Zhou
zhoudacheng@shu.edu.cn

Wenbo Hui
atlantis_hui@shu.edu.cn

Afimbo Reuben Kwabena
reubenkwabena2@gmail.com

Yubo Xing
xyb_2000@shu.edu.cn

Yi Qian
qianyi@shu.edu.cn

Quan Li
liquan2021@163.com

Huayan Pu
phygood_2001@shu.edu.cn

[1] School of Mechatronics Engineering and Automation, Shanghai University, No. 99 Shangda Road, Shanghai 200444, China

[2] Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, No. 99 Shangda Road, Shanghai 200444, China