electronics

MDPI

Article

Integrated Rover Path Planning and Validation on Real Outdoor Terrain Scenarios Using Satellite Information to Conduct a Real Achievable Trajectory

Stelian Brad * and Bogdan Balog *

Design and Robotics Engineering Department, Technical University of Cluj-Napoca, Blv. Muncii 103-105, 400641 Cluj, Romania

* Correspondence: stelian.brad@staff.utcluj.ro (S.B.); bogdan.balog@campus.utcluj.ro (B.B.)

Abstract: The reliable and efficient navigation for mobile robots across challenging outdoor terrains is critical for autonomous robotics. Traditional methods for planning the path of such robots often emphasize minimizing the travel distance but do not accommodate terrain stability, variability, or energy efficiency. The study proposes an integrated approach between satellite-driven geolocation data and terrain-specific features that enhance the path planning strategies in complex outdoor environments. Our method proposes a controller that uses search-based algorithms to generate energy-efficient and dynamically stable trajectories incorporating surface characteristics and environmental data from satellite imagery. By integrating our method, the proposed framework identifies safer and more reliable routes, achieving a significant 32% improvement in traction characteristics compared to the conventional models of path-finding approaches. Our method's benefits over traditional approaches include improved safety, extended operational efficiency, and the ability to navigate unpredictable and dynamic environments. This makes it ideal for planetary exploration, disaster response in landslide-prone areas, agricultural automation for precision farming in rough terrains, search and rescue operations in earthquake-affected areas, and autonomous delivery systems navigation into rural and unstructured landscapes. It redefines autonomous navigation through terrain-aware planning and delivers a robust performance approach in unpredictable and dynamic environments.

Keywords: autonomous navigation; mobile robots; path planning; satellite data; adaptive navigation; real-time terrain optimization; safe path trajectory

check for updates

Academic Editor: Giuseppe Prencipe

Received: 9 January 2025
Revised: 14 February 2025
Accepted: 17 February 2025
Published: 26 February 2025

Citation: Brad, S.; Balog, B. Integrated Rover Path Planning and Validation on Real Outdoor Terrain Scenarios Using Satellite Information to Conduct a Real Achievable Trajectory. *Electronics* **2025**, *14*, 921. https://doi.org/10.3390/electronics14050921

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the ever-changing field of robotics, mobile robots have become essential tools, revolutionizing manufacturing, logistics, space exploration, and disaster recovery. As these devices operate in complex and unpredictable settings, the difficulty of route planning becomes crucial. How can a robot identify the safest and most efficient path to its destination while traversing changing landscapes and avoiding obstacles?

In recent years, mobile robots have become revolutionary devices in various industries, facilitating automation in intricate and ever-changing settings. These robots are especially beneficial in outdoor situations, where changes in terrain and uncertain conditions pose considerable difficulties. Uses include agriculture, disaster recovery, environmental monitoring, and planetary exploration, demonstrating flexibility and significance in tackling real-world issues [1,2].

1.1. Applications of Mobile Robots in Outdoor Environments

One prominent application of mobile robots is agriculture, where autonomous systems have been utilized for crop monitoring, precise weeding, and harvesting. For example, mission planners for agricultural weeding robots create effective and secure paths, enhancing productivity and minimizing environmental effects [3]. Additionally, affordable robotic systems designed for agricultural purposes improve accessibility and flexibility, allowing for autonomous navigation in operations with limited budgets [4]. Robots incorporating LiDAR and vision systems enhance obstacle avoidance, facilitating smooth operations in unstructured agricultural fields [5]. Recent developments like hybrid heuristic algorithms and deep reinforcement learning models have been effectively utilized for farm robots, enhancing their decision-making and navigation capabilities in dynamic settings [6,7].

Mobile robots significantly contribute to disaster recovery efforts, especially when human involvement is risky or unfeasible. Algorithms such as improved A* models enable robots to navigate debris-filled areas and determine safe paths [8]. Robots with inertial and magnetic sensors assess terrain conditions in real-time, guaranteeing reliable navigation during rescue operations [9]. These abilities illustrate how autonomous systems improve disaster recovery by enabling quick and secure entry to dangerous areas [10].

Environmental Monitoring

Environmental monitoring is another critical domain benefiting from mobile robot navigation. Systems created for outdoor navigation can gather information on soil conditions, plant health, and water quality [11]. Ground robots with sophisticated navigation systems adjust to challenging landscapes, allowing them to conduct ecological assessments [12] accurately. Recent advancements in semantic-geometric planning enhance path optimization for environmental uses, guaranteeing effectiveness and flexibility in response to shifting terrains [13].

These are just a few examples of research conducted in the past years on robot path planning solutions.

1.2. Advancements in Mobile Robot Path Planning

Mobile robots have become essential across numerous fields, including manufacturing, logistics, space exploration, and disaster recovery. Path planning is a significant challenge in robotics, which involves determining an ideal route for a robot to move from a starting point to a destination while steering clear of obstacles. This review consolidates advancements in mobile robot path planning techniques in recent years, classifying the strategies into graph-based methods, heuristic and optimization techniques, machine learning, and hybrid approaches. It additionally emphasizes the difficulties and future avenues for research.

1.2.1. Graph-Based Methods

Graph-based methods have been foundational in the development of path-planning algorithms. Techniques like Dijkstra's and A* algorithms have been widely applied to find the shortest paths in known environments. Dijkstra's algorithm ensures optimal solutions by exploring all possible paths but can be computationally expensive for large-scale problems [14]. The A* algorithm enhances Dijkstra by incorporating a heuristic to prioritize exploration, making it more efficient [14,15]. Variants of A*, such as the T* and Risk-A*, have been developed to address specific issues like handling temporal constraints and improving safety in uncertain environments [16].

Graph-based approaches have demonstrated robustness and scalability for real-world applications like navigation in road networks and robotic exploration of Mars-like terrains.

However, these methods often assume static environments and struggle in dynamic or partially observable settings [14,15].

1.2.2. Heuristic and Optimization Techniques

Heuristic- and optimization-based techniques offer alternative approaches to path planning. Artificial potential fields (APFs), genetic algorithms (GAs), and particle swarm optimization (PSO) provide flexible solutions for path planning. APF methods rely on attractive and repulsive forces to guide the robot but are prone to local minima issues, which newer hybrid approaches attempt to mitigate [17].

Optimization-based methods such as GAs and PSO leverage population-based search strategies to find feasible paths in complex environments. These approaches are computationally intensive, requiring significant computational resources and time to find the best path. For instance, a heuristic PSO algorithm was used to address static obstacle avoidance but faced limitations in real-time adaptability [18]. However, they offer the ability to handle multi-objective problems, such as balancing energy efficiency with time constraints.

Sampling-based planners are well suited for high-dimensional spaces, including Rapidly exploring Random Trees (RRTs) and Probabilistic Roadmaps (PRMs). An RRT quickly generates feasible paths by randomly sampling the configuration space, making it practical for robotic arms and aerial vehicles [16]. However, the lack of optimality in RRT paths has prompted the development of variants like RRT*, which ensures path optimization over time [19].

1.2.3. Machine Learning Approaches

Machine learning (ML) has significantly influenced the evolution of path planning. For instance, reinforcement learning (RL) enables robots to learn optimal policies through interaction with their environments. An RL-based approach was applied in urban search and rescue operations, effectively minimizing search time for hidden targets [17]. Graph neural networks (GNNs) have also been explored for processing graph-structured inputs, offering robust solutions in dynamic settings [20].

Deep learning models have been integrated with traditional path-planning techniques to improve obstacle detection and dynamic decision-making. However, ML methods often require extensive training data and computational resources, making their deployment challenging in real-time applications.

1.2.4. Hybrid Methods

Hybrid approaches combine the strengths of multiple techniques to address the limitations of individual methods. For example, a hybrid method integrating Beetle Antennae Search (BAS) with an APF was proposed for dynamic path planning, achieving superior performance in convergence speed and obstacle avoidance [18]. Another hybrid model merged A* with an RRT to handle indoor navigation effectively [18].

In addition to enhancing traditional path-planning methods, hybrid approaches increasingly leverage machine learning models to improve decision-making in uncertain environments. For instance, deep reinforcement learning (DRL) is integrated with algorithms like hybrid A*, allowing robots to adapt their paths based on real-time environmental feedback [21]. Moreover, evolutionary algorithms such as genetic algorithms (GAs) play a crucial role in optimizing path generation in high-dimensional spaces when integrated with sampling-based planners like RRT*. This integration ensures not just optimization but global optimality, providing a reassuring effectiveness of the methods [22].

1.2.5. Multi-Robot Path Planning

Path planning for multi-robot systems introduces additional complexity, including collision avoidance and coordination. Deterministic approaches like centralized planning often struggle with scalability as the number of robots increases [23]. On the other hand, bio-inspired techniques and distributed planning methods offer promising solutions for enhancing fault tolerance and efficiency in multi-robot teams [23,24].

For instance, ant colony optimization has been adapted for multi-agent systems, showing robust performance in cooperative tasks [19]. These approaches enable efficient coordination among multiple robots, making them suitable for applications such as swarm robotics, automated warehouses, and collaborative exploration missions.

1.3. Recent Advances in Mobile Robot Path Planning

Recent research in mobile robot path planning has shifted from traditional graph-based and heuristic methods to learning-driven approaches that enhance adaptability, real-time decision-making, and uncertainty handling. While A* and Dijkstra's algorithms remain fundamental for shortest-path computation in structured environments, they struggle in dynamic and high-variability terrains [25]. ANFIS-based controllers, fuzzy logic strategies, and reinforcement learning (RL) techniques have been introduced to address these limitations, allowing robots to adapt their navigation policies based on real-time sensor inputs. Controllers based on ANFIS, fuzzy logic approaches, and reinforcement learning (RL) methods have been developed to overcome these constraints, enabling robots to adjust their navigation strategies according to real-time sensor data. ANFIS-based approaches [26] excel due to their distinctive capacity to merge fuzzy logic with neural networks, allowing robots to acquire effective strategies for avoiding obstacles while keeping computational demands minimal. This adaptability reassures us of the robustness of these techniques. Similarly, fuzzy control strategies [27] offer a multi-objective approach, allowing robots to prioritize collision avoidance, goal-reaching, and energy efficiency dynamically. These methods have been particularly effective in agricultural robotics, search-and-rescue, and industrial automation, where uncertain and cluttered environments challenge deterministic planners.

Fuzzy logic has gained traction primarily due to its unique ability to handle uncertainty, a crucial navigation aspect. Unlike deterministic methods, fuzzy logic provides a robust framework for managing imprecise sensor data and dynamically adjusting navigation strategies. Recent studies have demonstrated the effectiveness of fuzzy controllers in real-time obstacle avoidance, trajectory optimization, and collision-free motion planning in highly dynamic environments [16]. Fuzzy-based path planning has been instrumental in applications such as search-and-rescue missions and agricultural automation, where the terrain conditions and obstacles continuously change.

Artificial neural networks (ANNs) and adaptive neuro-fuzzy inference systems (ANFISs) have been integrated into path planning, enhancing decision-making abilities with their autonomous nature. Utilizing machine learning methods, these models are designed to adjust to environmental shifts, thereby enhancing navigation routes using experience and current sensor information [18]. The ANFIS, a fusion of fuzzy logic and neural networks, empowers mobile robots to learn the best paths and make decisions independently, much like a human would. These methods have been effectively utilized in self-driving cars in urban settings, automating warehouses, and exploring planets, showcasing their autonomy and decision-making prowess in practical uses [24].

Despite significant advancements, mobile robot path planning faces challenges like dynamic environments. Many traditional algorithms assume static environments, which are unrealistic for most real-world applications. Real-time adaptability remains a critical area of improvement [18]. Robots operating in 3D or high-dimensional spaces require

scalable algorithms to handle computational complexity [16]. This is implied since these platforms must be operated in increasingly large and high-dimensional spaces. Due to this fact, increasing the traveled surfaces requires an increased involvement in energy efficiency by optimizing the paths for energy consumption while maintaining efficient time, which is vital for battery-powered robots [16]. Seamlessly integrating path planning with perception systems like SLAM (Simultaneous Localization and Mapping) is essential for robust navigation [15,16] solving a series of problems, but the specificity of the applications and the complexity of the platforms require increased development to create platforms that can manage extensive exploration spaces.

Reinforcement learning has emerged as a game-changer in autonomous navigation, empowering robots to self-learn optimal paths without needing predefined maps. The study on curriculum learning-based RL models [28] is a testament to this, showcasing how progressive training can equip robots to navigate simple terrains before tackling complex, obstacle-rich environments. This approach mitigates the sample inefficiency problem often associated with deep RL approaches. Furthermore, integrating sensor fusion (LiDAR, vision, IMU) and graph neural networks (GNNs) has significantly enhanced context-aware decision-making, making RL-based models suitable for urban navigation, planetary exploration, and autonomous driving. Despite these advancements, challenges such as scalability, computational cost, and hybrid model integration remain key areas for future research. Future initiatives should focus on merging rule-based heuristics with deep learning methods, utilizing cloud computing for immediate adaptation, and guaranteeing that learning-guided path planning can be used in various environments.

Developing adaptive and learning-based algorithms that can generalize across environments, combining reinforcement learning with traditional methods, exploring decentralized approaches for multi-robot systems, and leveraging cloud-based computation for real-time planning are promising directions [20,23].

2. Materials and Methods

Figure 1 presents the step-by-step methodology for terrain-aware path planning. It begins with data collection, including geographic information (latitude and longitude) for the area of interest, followed by generating a GeoJSON file that specifies the region, start, and goal points. The process continues by combining elevation data and friction coefficients. Elevation data are extracted and visualized in a 3D plot, while terrain types are classified from satellite imagery to generate a friction map. The final steps involve running a search-based path planning algorithm to determine an optimal trajectory and validating the generated path for practical use.

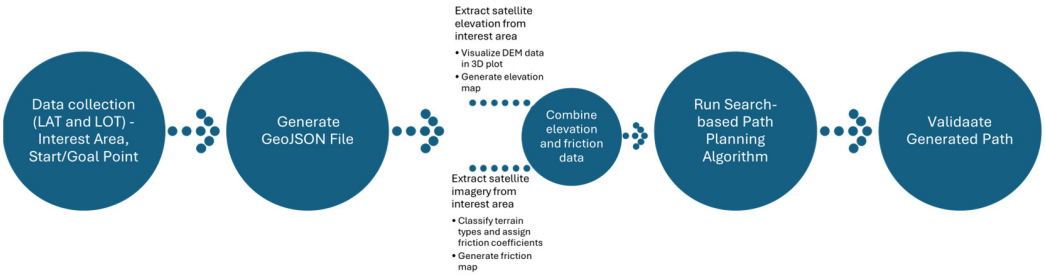


Figure 1. Methodology flowchart.

2.1. Elevation Data Colection

2.1.1. Map Data Collection with Latitude and Longitude Information

The code creates an interactive map with the Leaflet library, enabling users to explore and engage with a worldwide map centered at the coordinates [0, 0]. It offers instruments

for creating figures like polygons and lines, which can be stored in GeoJSON format for additional uses. An export function is provided, allowing users to download the created shapes as a GeoJSON file, and the shapes' data can also be displayed via alerts or output to the console.

Figures 2 and 3 show the html interface of the zone picker.

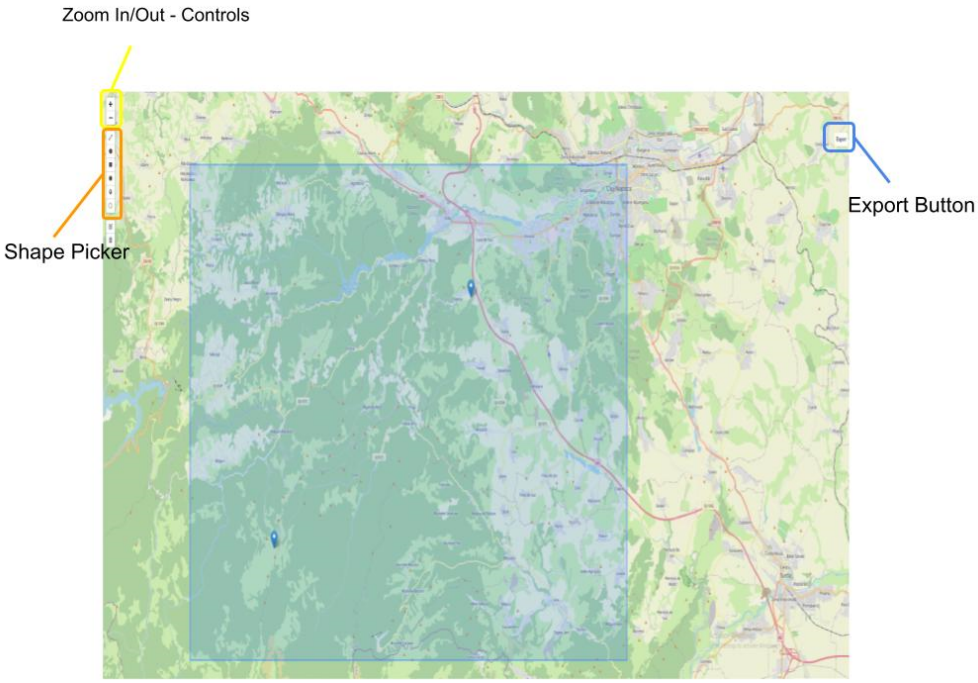


Figure 2. Interface to choose the region of interest.

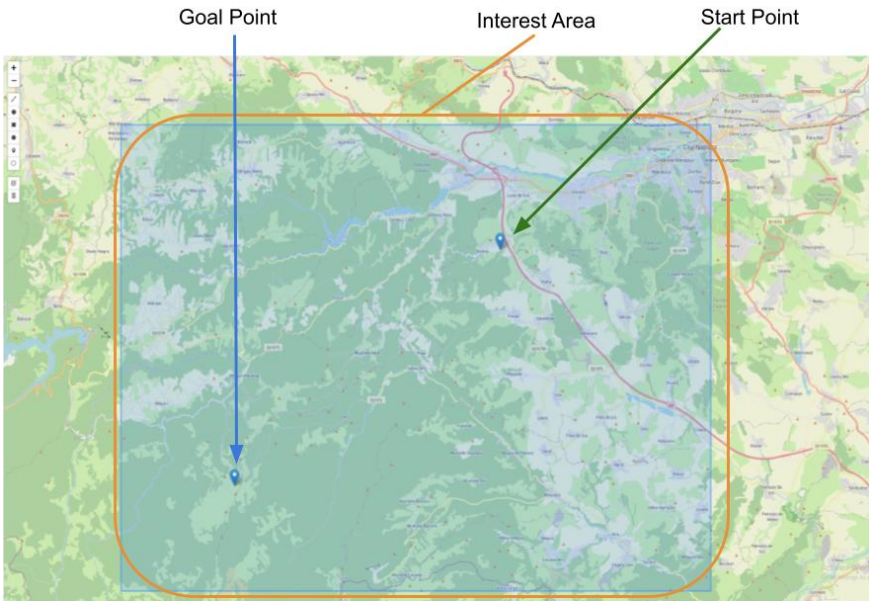


Figure 3. Interface region of interest selector.

At this point, the GeoJSON file is saved, which contains the specific latitude and longitude of the selected rectangular shape and the start and goal points, as can be seen in Table 1.

Table 1. Interest area, start point, and goal point are in GeoJSON format.

	GeoJSON Format Data
Interest area	{"type":"Polygon","coornates": [[[23.086395,46.524383], [23.086395,46.781255], [23.599319,46.781255], [23.599319,46.524383], [23.086395,46.524383]]]}
Start Point	{"type":"Point","coordinates": [23.415985,46.712325]}
Goal Point	{"type":"Point","coordinates": [23.184929,46.582226]}

2.1.2. Elevation Data Collection Using Open Topography API

The established framework for analyzing and processing Digital Elevation Models (DEMs) offers a thorough and effective method for terrain assessment. It combines various Python tools and libraries to automate the process from data collection to visualization and export. The procedure starts with obtaining a bounding box from a GeoJSON file. This phase guarantees that the region of interest is precisely identified by checking and handling the geometries of the elements in the GeoJSON. The framework prevents errors in later data acquisition stages by verifying geometries with non-zero areas.

The bounding box requests elevation data from the OpenTopography API, offering high-resolution DEMs in the designated geographic area. For this implementation, the 30 m DEM from the SRTMGL1 (Shuttle Radar Topography Mission) is selected, balancing resolution and computational efficiency. The API request is set up to deliver the data in the GeoTIFF format, which is perfect for geospatial analysis. After downloading, the DEM data are stored locally in a GeoTIFF format, allowing for subsequent processing.

Visualizing the DEM is an essential process for comprehending the physical characteristics of the terrain. Using Rasterio, the GeoTIFF file is accessed to obtain elevation data and georeferenced coordinate frameworks. The information is subsequently transformed into a 3D surface plot using Matplotlib 3.9.2. The depiction in Figure 4 offers a clear illustration of the landscape. The X and Y axes indicate longitude and latitude, respectively, whereas the Z-axis reflects elevation in meters. The terrain colormap is utilized to improve the visual differentiation of various elevation levels, employing warm colors to signify higher altitudes and darker shades to denote lower areas. This visual representation is beneficial for recognizing significant landforms like ridges, valleys, and inclines.

The elevation data are processed and saved as a CSV file to enhance analytical usage. This phase involves looping through the grid points to gather longitude, latitude, and elevation data, which are structured into a Pandas data frame. The generated CSV file includes three columns, Longitude, Latitude, and Elevation, offering a well-organized data set that can be effortlessly incorporated into different geospatial analyses or modeling processes. For example, these structured data can be utilized in pathfinding algorithms or to assess the appropriateness of terrain for robotic navigation.

The process includes strong error-handling measures to guarantee dependability. Invalid geometries within the GeoJSON are marked, and API replies are verified to manage possible errors. Throughout the export procedure, exceptions are handled to avoid data loss or damage. The framework's modular architecture guarantees flexibility, enabling it to be expanded for different uses, like hydrological research, disaster response, or robot terrain navigation.

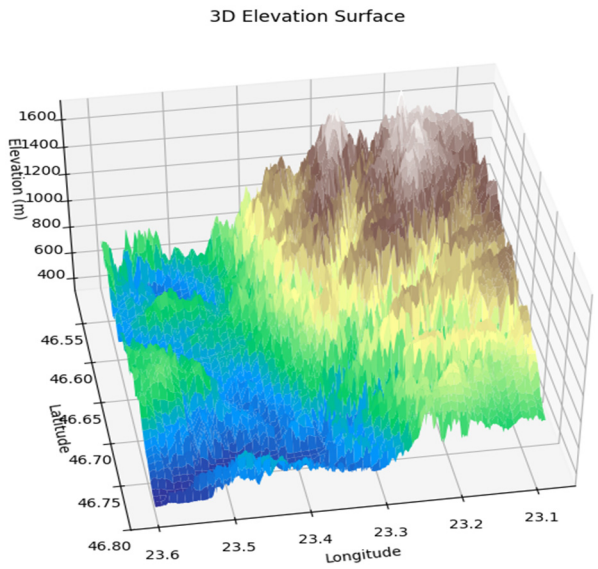


Figure 4. Three-dimensional plot of the geolocation with the elevation plot generated for the interest area.

2.2. Method for Extracting Surface Friction Coefficients from Satellite Imagery

A technique was created to produce a friction map from satellite images, aiding terrain analysis and enhancing robotic navigation applications. This procedure allocates friction coefficients to different terrain kinds, facilitating improved path planning and mobility evaluation for robotic systems.

The process starts by dividing a satellite image into land types like forests, grassland, water bodies, roads, and concrete areas. Predefined RGB color limits were utilized to categorize each terrain type according to its visual features. These thresholds were used through OpenCV's inRange() function to create binary masks for every category, isolating pixels that match particular terrain types.

After identifying the types of terrain, rough estimates of friction coefficients were allocated to each category according to values frequently referenced in the literature. For instance, coefficients were assigned 0.1 for water, indicating low friction, and 0.9 for concrete, indicating high friction. The values used can be seen in Table 2. These figures offer a valuable assessment of the slip resistance for various surfaces. The friction coefficients from the table below were classified based on the surface structure of the tested zone of the area fact, which was verified based on the information from the field. Figure 5 illustrates some of the terrain conformation from the interest area. The appreciation of the friction coefficient is tested on the visual inspection of a rubber tier capability to pass the specific terrain surface because of the surface's complexity. A safety coefficient based on the terrain's structural configuration was also considered. Since satellite imagery does not allow for centimeter-scale evaluation, this approach estimates friction zones using an average friction coefficient representing the surface's traction characteristics.

Table 2. Friction map-based coefficients.

Surface	Approximate Friction Coefficient
Water	0.1
Vegetation	0.2
Dirt	0.5
Grass	0.6
Forest	0.7
Stone	0.7
Road	0.8
Concrete	0.9

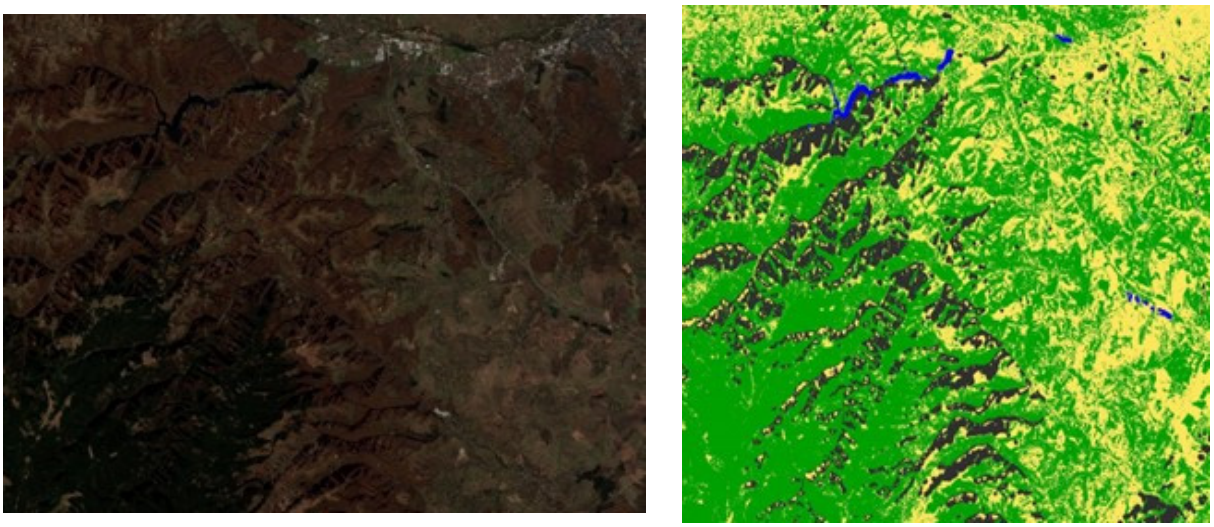


(**a**) (**b**) (c)

Figure 5. The friction coefficient of the terrain surface evaluation: (**a**) vegetation—0.2 friction coefficient; (**b**) dirt—0.5 friction coefficient; (**c**) stone—0.7 friction coefficient.

A friction map was created utilizing binary masks and the allocated friction coefficients. This required generating an empty map and filling it with friction values corresponding to the recognized terrain types. Every pixel on the map was revised with the corresponding coefficient according to its terrain type.

The resulting friction map was displayed with a color gradient (Figure 6), with distinct colors illustrating different degrees of friction (Figure 7). The map was stored in a NumPy array format, allowing integration into robotic controllers or applications for additional analysis. This depiction offers an in-depth comprehension of the terrain's traversability and aids in formulating firm mobility plans.



(**a**) (**b**)

Figure 6. Friction map data from satellite data image in the specific interest area: (**a**) real view satellite image; (**b**) proceed image with color spectrum over the surface of interest.

This method of friction mapping provides considerable benefits for robotic systems functioning in varied environments. By integrating friction information into path planning algorithms, robots can foresee and adjust to difficulties presented by different types of ter-

rain. This ability improves autonomous navigation's effectiveness, security, and reliability, particularly in complex or unpredictable environments.

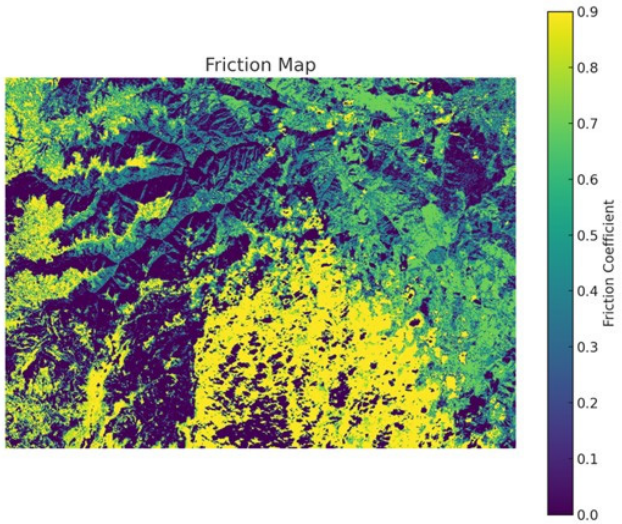


Figure 7. The friction map was processed using OpenCV with a friction coefficient.

Data Set Generation for Search-Based Algorithm

In this step, the script uses the generated CSV file with elevation information explained at point 2.1.2. Elevation data collection was performed using Open Topography API. In this stage, the image was transformed into a matrix form in a 2D format explained at point 2.2 in the same interest area by collecting the satellite image correlating from the area described in Section 2.1.1 to match the elevation coordinates (latitude and longitude). This ensures correspondence between each elevation point's elevation and friction data. Table 3 shows the data configuration inside the elevation and friction data set.

Table 3. Final data configuration of data set for search-based algorithm.

Longitude	Latitude	Elevation	Friction
23.086250000026894	46.78152777777603	715	0.9
23.086527777804672	46.78152777777603	715	0.9
23.086805555582448	46.78152777777603	715	0.9
...			

Note: Data available in Supplementary Materials Annex 1—Elevation_with_Friction.csv.

2.3. Final Data Configuration of Data Set for Search-Based Algorithm Baseline Route Without Friction Coefficient

The algorithm solely analyzes elevation data to calculate movement costs. Every cell in the terrain grid is assessed exclusively by its elevation value. This basic model leads to routes that minimize overall elevation expense, potentially overlooking difficulties in surface navigation.

2.3.1. Improved Route with Resistance

The cost model is expanded to incorporate surface friction, where the expense of moving through each cell is based on multiplying its elevation and friction values. This modification replicates realistic movement conditions, imposing penalties for traversing high-friction areas and directing the algorithm to steer clear of these regions when possible.

The initial and target points, given as latitude and longitude coordinates, are converted into grid indices to guarantee compatibility with the computational grid. These inputs run both algorithm iterations, producing two unique paths.

2.3.2. The Analysis of Routes Centers Around Various Essential Elements

The two routes' visual contrast shows friction's impact on trajectory choices. For instance, the path mindful of friction might steer clear of high-resistance areas, even if they belong to a more direct route. This illustrates the planner's capacity to favor practical paths over merely ideal ones regarding distance.

The overall expense of the friction-aware route is usually more significant than that of the friction-free route. This heightened expense illustrates the extra labor needed to navigate friction-impacted zones, highlighting the real-world compromise between effectiveness and practicality.

Paths created without resistance might traverse unfeasible or dangerous regions, like steep inclines or very slick surfaces. Conversely, paths that consider friction typically show enhanced adaptability, moving through navigable terrain more efficiently. This difference emphasizes the shortcomings of simplistic cost models and reinforces the need to incorporate environmental considerations.

Incorporating surface friction into trajectory planning dramatically improves the performance and dependability of autonomous robots as they traverse various rugged terrains. This method enables planners to consider actual conditions and make choices that correspond with the robot's physical abilities and the limitations of the environment.

2.3.3. Realistic Navigation

Integrating friction into the trajectory planning process allows the simulation of actual traversal conditions. Planners considering friction can assess the challenges of moving across various surface types, like sandy deserts, muddy fields, or rocky terrains. This guarantees that the chosen trajectories are effective regarding distance or time and practical considering the terrain's features. By considering actual conditions, friction-aware planning enhances the robot's capacity to carry out its tasks in intricate environments effectively.

Friction-aware path planners allow robots to improve their navigation decisions by considering the additional resistance or difficulties associated with traversing high-friction areas. This enables planners to avoid hazardous areas, including steep, slippery slopes or unstable ground, minimizing navigation mistakes, enhancing operational safety, and maximizing energy efficiency. The reliability of the created routes, which blend efficiency with practicality, guarantees safe and efficient navigation through various terrains. In addition, comparing paths generated with and without friction offers a systematic method to evaluate the efficacy of the trajectory planner. This validation process emphasizes the role of friction in shaping the planner's choices and trajectory results, confirming that the friction model faithfully mirrors real-world scenarios and aids in developing effective navigation strategies. Moreover, the validation process pinpoints aspects where the algorithm might need improvement, thereby increasing the model's dependability and authenticity.

2.3.4. Adaptability in Complex Terrains

Friction-aware planning significantly improves a robot's capacity to adjust to diverse and unpredictable settings. By imposing penalties on paths that traverse surfaces characterized by high resistance or instability, the algorithm promotes routes compatible with the robot's physical abilities and environmental limitations. This flexibility enables the robot to maneuver across intricate landscapes while preserving efficiency and safety in operations. In both harsh natural environments and organized industrial contexts, trajectory planners that account for friction provide the adaptability required for robots to operate dependably across various conditions.

The comparative analysis of path planning with and without friction is a validation technique for evaluating the significance of terrain-aware cost modeling in robotic tra-

jectory planners. It illustrates the essential function of friction in providing realistic and dependable navigation, emphasizing the shortcomings of basic models and the improved effectiveness of planners that consider friction. This approach offers a solid structure for enhancing algorithms and improving navigation tactics in challenging landscapes. Figure 8 provides the output generated between 2 points (Start–Goal) on the tested zone for the algorithm check.

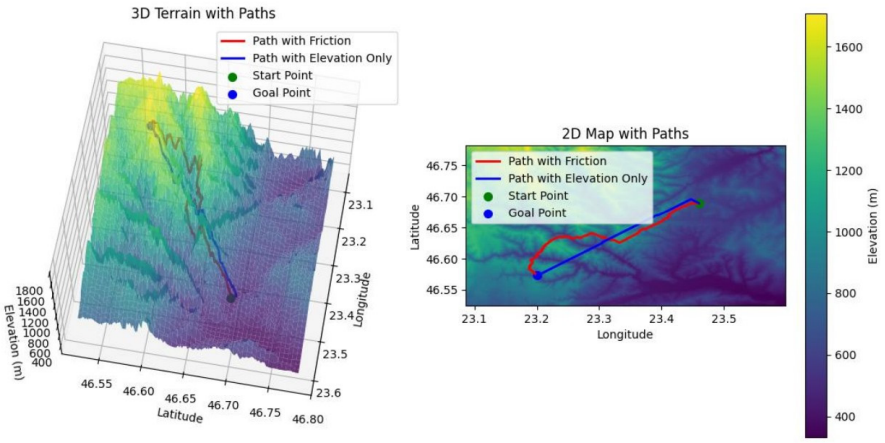


Figure 8. Trajectory generation with friction and elevation coefficients.

2.4. A Pathfinding Algorithm with Terrain Cost and Surface Friction Data Integration*

Effective navigation in challenging landscapes is essential for mobile robotics and self-driving vehicles. The A* algorithm, with its efficient pathfinding abilities, can be adapted to consider terrain factors like elevation and surface friction. Techniques for incorporating these environmental elements into A* offer a more realistic method for pathfinding.

A* is the preferred algorithm for path planning in a structured environment because it combines Dijkstra's optimal path assurance with the efficiency of heuristic search, significantly minimizing processing time and memory consumption. Unlike Dijkstra, which investigates each node in detail, A* focuses only on the most promising routes, resulting in higher speed and scalability for large maps. Furthermore, unlike an RRT, which depends on random sampling and frequently generates suboptimal or rough paths, A* guarantees accurate and consistent paths. This balance of performance, resource efficiency, and optimality positions the A* as particularly ideal for applications that require fast, reliable, and energy-efficient navigation, such as autonomous robots operating in outdoor environments or dangerous.

2.4.1. Terrain Cost Calculation

To accurately assess the challenge of moving through a grid cell, a terrain cost metric is established as outlined below.

$$\text{TerrainCost(i,j)} = \frac{\text{Elevation(i,j)}}{\text{Friction(i,j)} + \epsilon}, \tag{1}$$

Elevation (i, j): Represents the elevation of the grid cell (i, j).
Friction (i, j): Denotes the friction coefficient, which reflects the surface's slipperiness.
ϵ: A small constant 10^{-6} to prevent division by zero.

This equation guarantees increased elevations and reduced friction coefficients result in higher traversal costs. As a result, regions with sharp slopes or slick surfaces are less ideal for navigation, guiding the robot towards safer and more effective pathways.

The A* algorithm improves classic pathfinding by adding a heuristic function that approximates the cost from the present cell to the target. This heuristic and the terrain cost enable A* to prioritize routes more effectively.

The cost of moving from the current cell (x, y) to a neighboring cell (x', y') is calculated as follows:

$$CostToNeighbor(x, y, x', y') = CostSoFar(x, y) + TerrainCost(x', y'), \quad (2)$$

$CostSoFar(x, y)$: Represents the accumulated cost from the start cell to (x, y).

$TerrainCost(x', y')$: Adds the traversal cost for the neighboring cell.

The heuristic function calculates the remaining distance to the goal by applying the Euclidean distance formula:

$$\text{Heuristic}(\text{x}, \text{y}, x_g, y_g) = \sqrt{(x - x_g)^2 + (y - y_g)^2}, \quad (3)$$

This directs the algorithm to prioritize cells nearer the goal, thus minimizing unnecessary calculations.

The overall priority of a cell within the priority queue is defined as follows:

$$Priority\ (x', y') = CostToNeighbor(x, y, x', y') + Heuristic(x', y', x_g, y_g), \quad (4)$$

This combination ensures that the algorithm effectively balances the expenses of navigation and closeness to the target.

The integration of terrain elements like friction creates two unique situations for pathfinding:

In this situation, the terrain cost encompasses both height and friction coefficients:

$$PriorityWithFriction(x', y') = \frac{Elevation\ (x', y')}{Friction(x', y') + \in} + Heuristic(x', y', x_g, y_g), \quad (5)$$

Low friction surfaces increase expenses in this situation, deterring passage unless it is essential. This method illustrates the physical limitations of moving through actual landscapes.

When friction is disregarded, the terrain cost is reduced to consider solely elevation:

$$PriorityWithoutFriction(x', y') = Elevation(x', y') + Heuristic(x', y', x_g, y_g), \quad (6)$$

This situation is more straightforward to compute but is not as reflective of actual challenges as it presupposes consistent surface characteristics throughout the landscape.

2.4.2. Optimization Goal

The goal of the A* algorithm is to determine the most efficient path from the start cell, (x_s, y_s) to the goal cell (x_g, y_g). This is achieved by minimizing the total terrain cost along the path:

$$min \sum_{(x,y) \in Path} TerrainCost(x, y)\ , \quad (7)$$

Considering both elevation and friction, the algorithm determines a route that adheres to the robot's physical limitations and the constraints of the environment.

2.4.3. Movement Band Generation

The deviation band is created around the primary pathway according to a tolerance parameter acceptance tolerance (standard 5%). The adjacent points are examined for every

point along the main path to determine if their elevation and friction values fall within the acceptable tolerance. The formula below describes the requirement for an adjacent point:

$$base_value \times (1 - tolerance) \leq neighbor_value \leq base_value \times (1 + tolerance), \quad (8)$$

where

base_value: The base value is the friction or elevation value of the main path point.

neighbor_value: The neighbor value is the friction or elevation value of the neighboring point.

The tolerance allows for slight deviations from the path based on the terrain's friction and elevation characteristics.

2.4.4. Visualization and Practical Implications

To assess the effect of friction on pathfinding, both situations—with friction and without friction—are illustrated as follows:

- With Friction: The calculated route circumvents slick or unstable areas unless necessary, guaranteeing safer and more feasible navigation.
- No Friction: The route relies entirely on height, possibly resulting in impractical paths because of inadequate grip or lack of stability.

These visualizations emphasize the significance of incorporating terrain friction into the cost function, especially for robots working in rough or slippery landscapes.

This approach dramatically improves the A* algorithm by integrating terrain costs considering elevation and surface friction. By combining these elements, the algorithm attains a more accurate depiction of traversal difficulty, which is especially crucial in settings featuring varied and demanding landscapes.

When friction is included in the terrain cost, the algorithm can precisely evaluate the joint effects of elevation and surface characteristics on the robot's capacity to navigate the terrain. This method guarantees that slippery or unstable surfaces, defined by low friction coefficients, receive increased costs. Consequently, the algorithm instinctively steers clear of these regions unless they are essential for achieving the objective. This enhances the pathfinding procedure to better mimic real-world limitations, especially for robots in rough or dangerous environments.

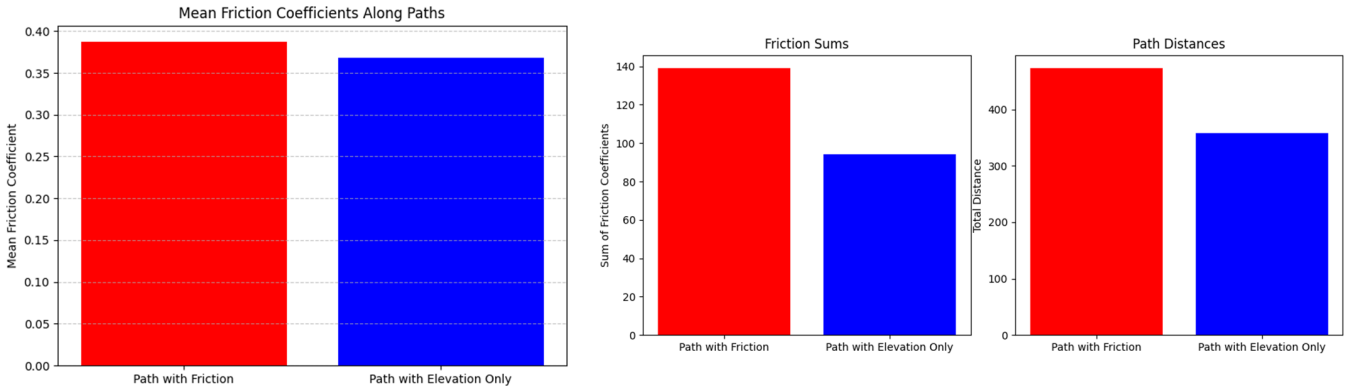
Conversely, when friction is removed from the terrain cost, the algorithm depends only on elevation to assess traversal difficulty. Although this simplification decreases computational complexity, it does not consider the physical characteristics of the surface. As a result, the routes created in this situation might be less feasible since they overlook the difficulties presented by low-friction surfaces. This method could result in inadequate or impractical pathways in scenarios where surface grip is essential.

Incorporating these terrain elements enables the A* algorithm to calculate routes that effectively reduce the overall traversal expense while adhering to environmental limitations and the robot's physical abilities. Through the harmony of elevation and friction, the algorithm reaches a crucial optimization level necessary for autonomous systems to function in real-world settings.

This terrain-sensitive navigation method benefits self-guided movement in varied and uncertain landscapes. It provides a strong foundation for decision-making, allowing robots to adjust to different environmental situations while ensuring operational effectiveness. Consequently, this approach establishes a solid base for upcoming progress in autonomous systems, facilitating enhanced mobility and durability in intricate landscapes.

3. Results

In the case of implementing the presented approach, the improvement results are mainly in terms of safety and the potential for increasing the performance of crossing various surfaces in outdoor environments, as shown in Figure 9. Increases were obtained in the total amount of the coefficient of friction during the newly generated trajectory considering the friction factors of the media to be crossed from the sum of friction coefficients (elevation only): 94.26192384769215 to the sum of friction coefficients (modified priority): 138.98817635275915, meaning an improvement of the general friction factor by over 32%. Also, the distance traveled increased by a percentage of 24% (Picture b from Figure 8).



(**a**) (**b**)

Figure 9. Path considering surface friction and elevation compared with path considering only the elevation factor: (**a**) mean friction coefficient along generated paths; (**b**) friction and path plots with traction and overall distance difference.

We also tried to check the approach on three more data pairs in the results. As Table 4 shows, the pairs were chosen randomly from the same interest area and five different areas (Table 5) to test the efficiency and effectiveness of the proposed method. In this way, we can evaluate the approach in more regions.

Table 4. Test of four different pairs of start–goal trajectory generation.

Trajectory	Start Point		Goal Point	
	Longitude	Latitude	Longitude	Latitude
1	23.337755	46.629181	23.216846	46.567124
2	23.387616	46.546043	23.457506	46.556512
3	23.404064	46.544275	23.521189	46.607488

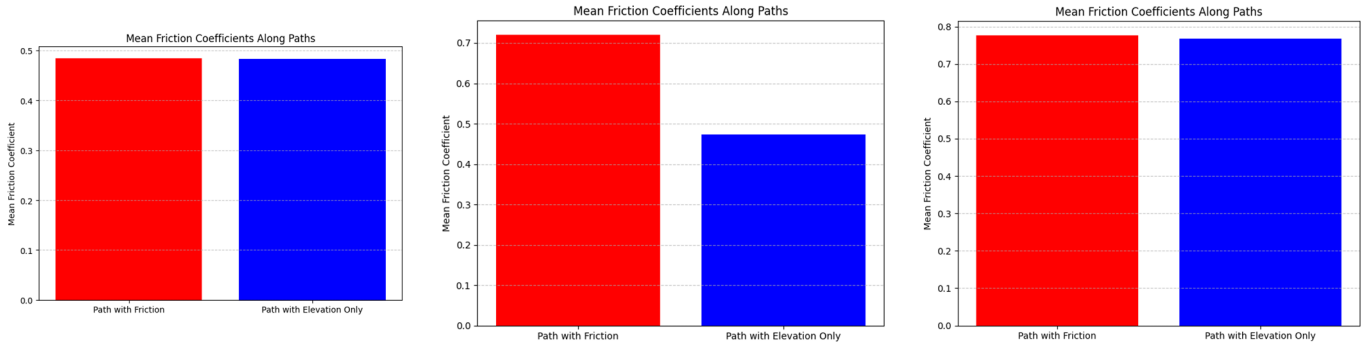
Table 5. Expand method evaluation in more different areas.

Trajectory	Start Point		Goal Point		Total Traction Improvement [%]	Total Distance Increase [%]
	Longitude	Latitude	Longitude	Latitude		
TR-1	22.829418	47.169469	22.831821	47.146707	6.37	1.93
TR-2	22.832083	47.164583	22.821527	47.147638	1.82	0
TR-3	22.831821	47.167949	22.823579	47.16128	8.78	0
TR-4	22.811426	47.16325	22.834357	47.141481	20.27	0.90
TR-5	22.838167	47.167501	22.811054	47.150038	20.57	8.20
TR-6	22.832377	47.170581	22.829413	47.158469	14.79	4.78
TPN-1	22.421036	47.162525	22.576733	47.098292	2.00	0

Table 5. *Cont.*

Trajectory	Start Point		Goal Point		Total Traction Improvement [%]	Total Distance Increase [%]
	Longitude	Latitude	Longitude	Latitude		
TPN-2	22.396596	47.188669	22.531234	47.100567	2.67	0.97
TPN-3	22.500837	47.188669	22.531234	47.100567	2.03	1.46
TPN-4	22.477217	47.077152	22.531234	47.100567	1.71	0
TPN-5	22.370736	47.160462	22.592958	47.103419	1.20	0.60
TSS-1	22.779293	47.269096	22.896366	47.235888	0.59	1.12
TSS-2	22.763242	47.26524	22.896795	47.236575	1.66	1.10
TSS-3	22.792939	47.23133	22.874135	47.295984	5.76	4.39
TSS-4	22.871389	47.240654	22.768907	47.264191	3.98	2.74
TSS-5	22.912563	47.244151	22.733327	47.264658	1.52	2.58
TH-1	22.967777	46.903369	23.05335	46.87228	9.55	0
TH-2	22.951327	46.919343	22.967714	46.876296	8.48	5.32
TH-3	23.070283	46.912958	22.968564	46.903588	7.93	1.27
TH-4	23.027192	46.893884	23.018452	46.867749	2.34	0
TDD-1	28.361893	44.982771	28.587794	44.804036	1.26	0.47
TDD-2	28.30226	44.98928	28.332298	44.836372	4.29	4.65
TDD-3	28.487446	44.814871	28.329964	44.96627	2.37	1.16
TDD-4	28.607139	44.863399	28.654493	45.000995	6.21	4.72
TDD-5	28.465786	44.886438	28.654493	45.000995	5.86	4.43

The data obtained, as can be seen in Figure 10 and Table 6, the mean friction coefficients along the generated trajectory (mean friction coefficient (path with surface friction) from Table 6) is higher than the friction coefficients evaluating only the elevation coefficients to create the trajectory (mean friction coefficient (path without surface friction) from Table 6). This shows that the framework using the terrain friction coefficient improves the overall traction characteristics over the new paths generated. This increases the safe travel over the terrain in a more robust way.



(**a**) (**b**) (c)

Figure 10. Plots for comparing the mean friction coefficient on the trajectory generated considering friction and not considering friction coefficients: (**a**) Trajectory 1; (**b**) Trajectory 2; (**c**) Trajectory 3.

Table 6. Mean friction coefficients for paths generated consider the surface friction coefficient and do not consider the surface friction coefficient.

Trajectory	Mean Friction Coefficient (Path with Surface Friction)	Mean Friction Coefficient (Path without Surface Friction)
1	0.483962	0.483143
2	0.719382	0.473349
3	0.776021	0.767905

We extend the search, and Table 5 shows the data obtained in the new search area with different starting and goal points. The table can also be seen as the total traction improvement on the method-generated path in cooperation with the elevation-only search method. The distance of travel is compared in the same manner.

This fact shows that a safer route is not shorter in terms of crossing distance, but it is safer from the point of view of the risky areas, reducing the dangerous areas and the blockages that the platform can get stacked. This has also been improved by allowing the algorithm to look for not a single linear route but rather a lane on which the platform can travel. This is achieved by allowing the algorithm to generate a route with a friction coefficient deviation of 5%, shown in Figure 11, which plots the trajectory and the deviation possibility zones illustrated by gray "x" in the areas that are safe to travel along the main trajectory from the main path described by a red continuous line between the start point illustrated with a green dot and the goal point illustrated with a blue dot in which the robot can still travel safely from elevation and friction points of view. A fact that can be seen in the image is that both the optimal trajectory and the deviations around it are highlighted in more detail in Figure 12.

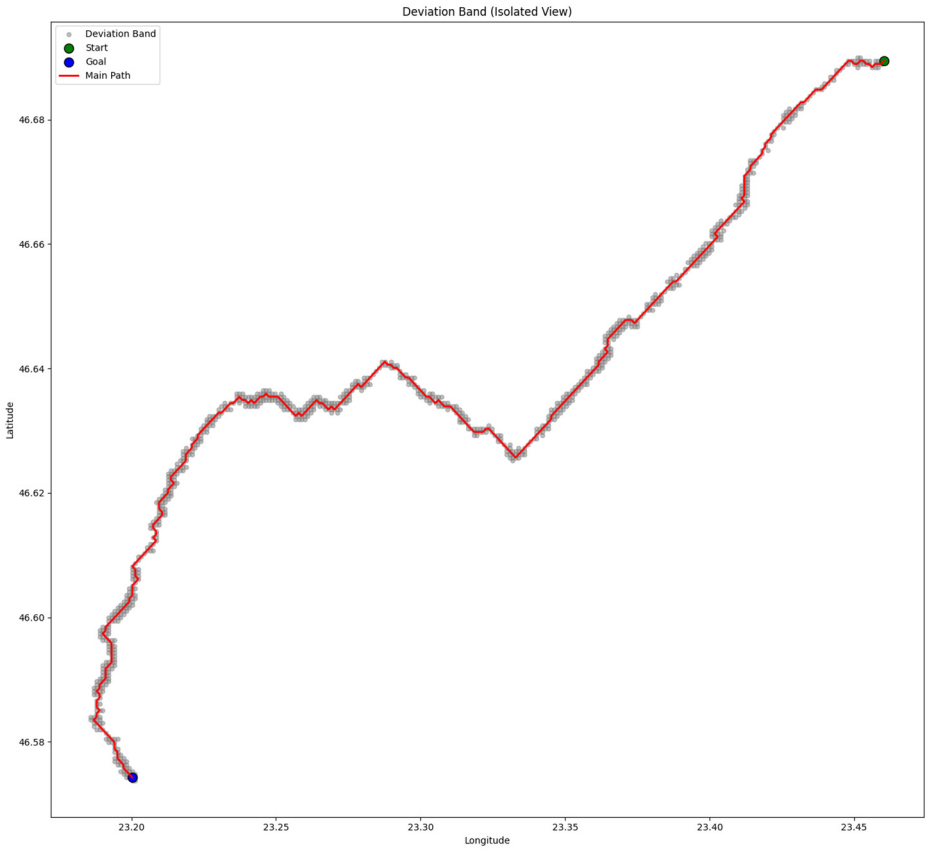


Figure 11. Deviation band around the entire main path.

From the evaluations and verifications on the ground area, it was observed that the trajectory proposed by the applied method is practicable and that this method offers a feasible solution.

Figure 13 reinforces the approach presented as optimal for the platform's safety as it travels on the complex terrestrial trajectory in the outdoor environment. It shows the smooth and consequent terrain area.

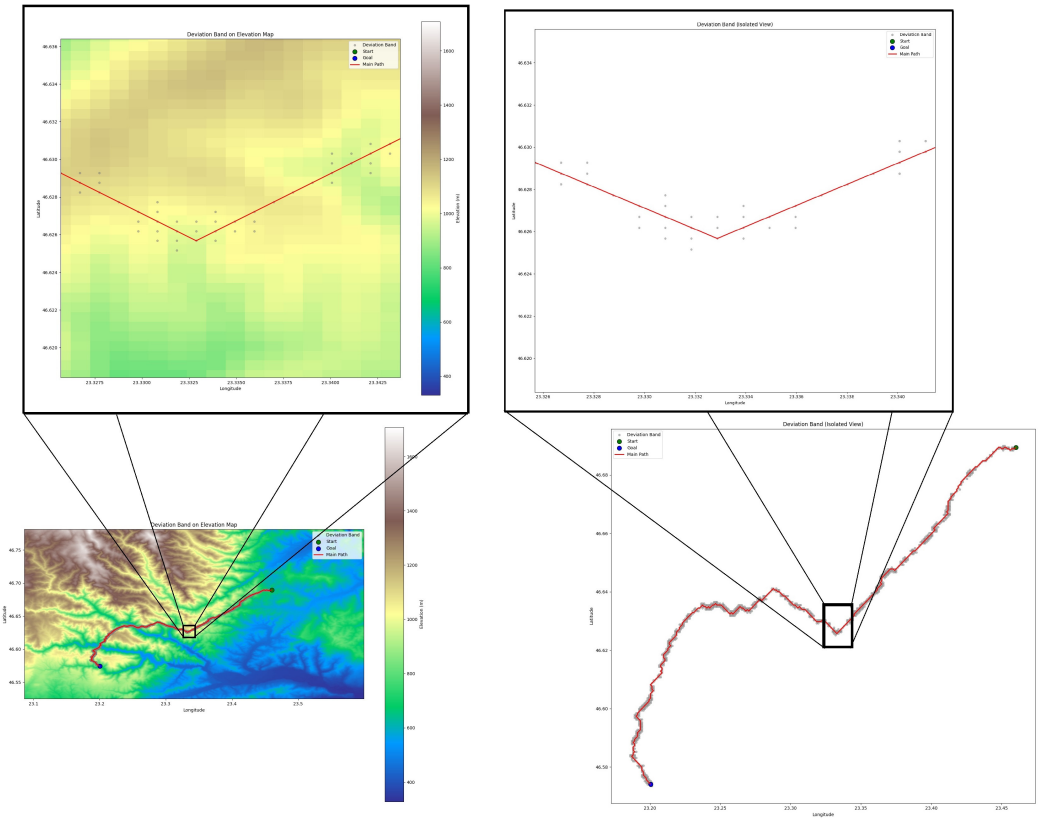


Figure 12. Figure of the deviation band around the main path.

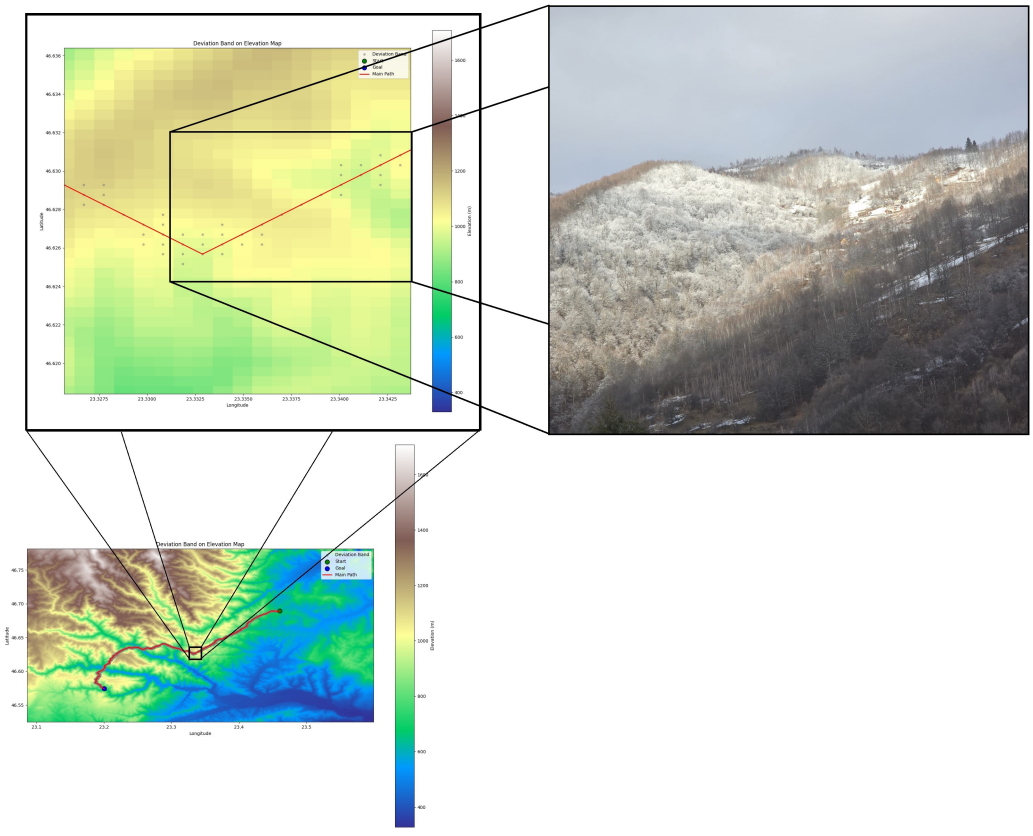


Figure 13. Real path evaluation validation.

4. Discussion

Classical domain route planning methods for a mobile robot mainly focus on reducing the distance or travel time, often ignoring essential aspects of the terrain, such as adhesion, stability, and energy efficiency. Our method proposes a novel combination of altitude data and estimated surface friction coefficients from satellite images to optimize route planning in complex outdoor environments. By integrating terrain cost modeling, our method provides safer, more energy-efficient, and more dynamically stable trajectories than traditional approaches that rely only on altitude data.

This framework significantly optimizes the adaptive capacity of autonomous navigation systems, helping mobile robots predict terrain difficulties, avoid risky areas, and refine traversal routes according to current conditions. Unlike classical algorithms that assume uniformly distributed terrain features, our approach includes a realistic assessment of terrain friction to affect navigation decisions dynamically. This refinement brings a 32% improvement in traction characteristics while maintaining reasonable travel efficiency, making it extremely valuable in applications such as planetary exploration, emergency response, and precision agriculture.

The algorithm's adaptability, which permits a 5% variance in friction coefficients, facilitates wider, safer lanes rather than fixed routes. Validation in real-world scenarios reinforced the significance of accounting for friction factors since routes based only on elevation proved harder to navigate. The method generally guarantees safer and more effective navigation on intricate landscapes.

4.1. Method Limitation

The method has limitations, mainly due to the physical restrictions of the data collection sources. A significant constraint is the reliance on the quality and resolution of the satellite images utilized for data extraction. The precision and dependability of the outcomes are inherently linked to the resolution of the raster data. In this research, the satellite information displayed a raster resolution of 30 m for each pixel. This degree of detail limits the amount of information that can be recorded, especially for landscapes with significant spatial variation. Detailed elements like minor obstacles, micro-topography, or abrupt changes in terrain features might not be precisely depicted, which could result in inaccuracies or oversimplifications in the simulated surface attributes.

Another constraint stems from lacking comprehensive, factual data for the regions assessed. Although remote sensing methods offer a comprehensive perspective, they might fall short in localized accuracy, especially when detecting dynamic surface conditions like changes in soil moisture, vegetation cover, or surface texture. To overcome this limitation, statistical techniques were used to assess and estimate the traction coefficients of the surfaces being studied. These coefficients were obtained from averaged surface characteristics instead of direct in situ measurements, adding another level of uncertainty.

The dependence on statistical estimates underscores the necessity of validating with actual field data to adjust and enhance the models. Integrating these data might boost the predictive precision of terrain navigation simulations and improve the functionality of robotic systems working in intricate outdoor settings. Even with these constraints, combining advanced satellite data with statistical analyses offers a practical approach for preliminary terrain evaluations, particularly when gaining direct access to remote or dangerous locations is difficult.

4.2. Future Work

Considering the results obtained from this research, further work can be extended on applying similar techniques to aerial images and integration of aerial drones and terrestrial

mobile robots for path planning navigation in outdoor environments for exploration or disaster assessment. Using machine learning and AI algorithms to establish the friction coefficient from the surfaces that the robot will have to navigate on in such a way can establish the best way of traversing. Including weather information like temperature and precipitation, almost instant evaluation of the area by combining spectral temperature image, IR image, and temperature sensors evaluation can ensure a better assessment of the surface stability and traction characteristics. This approach can be more suitable for continuously changing environments like volcanic eruption landscapes, areas affected by floods and tsunamis or even areas where landslides occurred, or areas affected by avalanches, ensuring solutions for safe escape routes and paths for the ground teams and rovers by utilizing prediction methods and AI algorithms in drawing the evolution of a multiple scenarios.

Our goal is to expand the research into a variety of scenarios, including arid desert areas, high humidity zones, and areas prone to frost or other factors. This will allow us to test the proposed solution thoroughly. The solution is designed to be adaptable to any environment, making it a decentralized and versatile path-planning solution. It is both specific enough to be effective and general enough to find an optimal trajectory solution.

In our upcoming research, we plan to investigate more effective pathfinding methods with real-world applications. Our suggested method entails streamlining the search area while preserving the precision of navigation that considers terrain. We aim to move from an utterly grid-based model to a topological graph framework, where only essential waypoints are treated as nodes in the A* algorithm. This method will significantly lessen the computational complexity and enhance the algorithm's scalability for more extensive and intricate environments, rendering it more applicable for real-world uses. Furthermore, we will incorporate graph clustering methods and hierarchical path planning to enhance the search process. These methods will enable the planner to function across various levels of abstraction, choosing broad routes initially and then honing them at a more detailed level. Our upcoming research will apply and verify these methods in simulated and actual environments to evaluate their effects on computational efficiency and path quality, thus showcasing their practical relevance and engaging the audience in the potential applications of our work.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/electronics14050921/s1.

Author Contributions: Conceptualization, S.B. and B.B.; methodology, B.B; software, B.B.; validation, S.B. and B.B.; formal analysis, S.B.; investigation, B.B.; resources, S.B.; data curation, B.B.; writing—original draft preparation, B.B.; writing—review and editing, S.B. and B.B.; visualization, B.B.; supervision, S.B.; project administration, S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article/Supplementary Material. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

A*	A-Star Algorithm
APF	Artificial Potential Field
ANNs	Artificial Neural Networks

ANFISs	Adaptive Neuro-Fuzzy Inference Systems
GA	Genetic Algorithm
GNN	Graph Neural Network
PRM	Probabilistic Roadmap
PSO	Particle Swarm Optimization
RL	Reinforcement Learning
RRT	Rapidly exploring Random Tree
SLAM	Simultaneous Localization and Mapping
DEM	Digital Elevation Model
IR	Infrared
DRL	Deep Reinforcement Learning

References

1. Wang, D.; Liu, Q.; Yang, J.; Huang, D. Research on Path Planning for Intelligent Mobile Robots Based on Improved A* Algorithm. *Symmetry* **2023**, *16*, 1311. [CrossRef]
2. Liu, C.; Xie, S.; Sui, X.; Huang, Y.; Ma, X.; Guo, N.; Yang, F. PRM-D* Method for Mobile Robot Path Planning. *Sensors* **2023**, *23*, 3512. [CrossRef] [PubMed]
3. Cordova-Cardenas, R.; Emmi, L.; Gonzalez-De-Santos, P. Enabling Autonomous Navigation on the Farm: A Mission Planner for Agricultural Weeding Robots. *Agriculture* **2023**, *13*, 2181. [CrossRef]
4. Baltazar, J.d.A.; Coelho, A.L.d.F.; Valente, D.S.M.; de Queiroz, D.M.; Villar, F.M.d.M. Development of a Robotic Platform with Autonomous Navigation Using Low-Cost Components. *AgriEngineering* **2024**, *6*, 192. [CrossRef]
5. Han, C.; Wu, W.; Luo, X.; Li, J. Visual Navigation and Obstacle Avoidance Control for Agricultural Robots Based on LiDAR and Vision Cameras. *Remote Sensing* **2023**, *15*, 5402. [CrossRef]
6. Quinones-Ramirez, M.; Rios-Martinez, J.; Uc-Cetina, V. Robot Path Planning Using Deep Reinforcement Learning. *arXiv* **2023**, arXiv:2302.09120.
7. Liu, W.; Wu, H.; Xiong, W.; Li, X.; Cai, B.; Yu, S.; Ma, J. Path Planning for Mobile Robots Based on the Improved DAPF-QRRT Algorithm. *Electronics* **2023**, *13*, 4233. [CrossRef]
8. Liu, Z.; Guo, S.; Yu, F.; Hao, J.; Zhang, P. Improved A* Algorithm for Mobile Robots under Rough Terrain Based on Ground Trafficability Model and Ground Ruggedness Model. *Sensors* **2024**, *24*, 4884. [CrossRef] [PubMed]
9. Sarcevic, P.; Csík, D.; Pesti, R.; Stančin, S.; Tomažič, S.; Tadic, V.; Rodriguez-Resendiz, J.; Sárosi, J.; Odry, A. Online Outdoor Terrain Classification Algorithm for Wheeled Mobile Robots Equipped with Inertial and Magnetic Sensors. *Electronics* **2023**, *12*, 3238. [CrossRef]
10. Gok, M.; Tekerek, M. Hamza Aydemir. Reinforcement Learning Based Local Path Planning for Mobile Robot. *arXiv* **2023**, arXiv:2403.12463.
11. Lv, Q.; Hao, G.; Huang, Z.; Li, B.; Fu, D.; Zhao, H.; Chen, W.; Chen, S. Localized Path Planning for Mobile Robots Based on a Subarea Artificial Potential Field Method. *Sensors* **2023**, *24*, 3604. [CrossRef]
12. Kremer, P.; Bavle, H.; Sanchez-Lopez, J.L.; Voos, H. S-Nav: Semantic-Geometric Planning for Mobile Robots. *arXiv* **2023**, arXiv:2307.01613.
13. Huang, X.; Yan, C.-B. Dubins Curve-Based Continuous-Curvature Trajectory Planning for Autonomous Mobile Robots. *arXiv* **2023**, arXiv:2309.07565.
14. Zeng, W.; Church, R.L. Finding shortest paths on real road networks: The case for A*. *Int. J. Geogr. Inf. Sci.* **2009**, *23*, 531–543. [CrossRef]
15. Potiris, S.; Tompkins, A.; Göktogan, A. Terrain-based path planning and following for an experimental Mars rover. In Proceedings of the Australasian Conference on Robotics and Automation (ACRA), Melbourne, Australia, 2–4 December 2014.
16. Qin, H.; Shao, S.; Wang, T.; Yu, X.; Jiang, Y.; Cao, Z. Review of autonomous path planning algorithms for mobile robots. *Drones* **2023**, *7*, 211. [CrossRef]
17. Sombolestan, S.M.; Rasooli, A.; Khodaygan, S. Optimal path-planning for mobile robots to find a hidden target in an unknown environment based on machine learning. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 1841–1850. [CrossRef]
18. Wu, Q.; Chen, Z.; Wang, L.; Lin, H.; Jiang, Z.; Li, S.; Chen, D. Real-time dynamic path planning of mobile robots: A novel hybrid heuristic optimization algorithm. *Sensors* **2020**, *20*, 188. [CrossRef] [PubMed]
19. Karur, K.; Sharma, N.; Dharmatti, C.; Siegel, J.E. A survey of path planning algorithms for mobile robots. *Vehicles* **2021**, *3*, 448–468. [CrossRef]
20. Osanlou, K.; Guettier, C.; Cazenave, T.; Jacopin, E. Planning and learning: Path-planning for autonomous vehicles—A review of the literature. *arXiv* **2023**, arXiv:2207.13181.

21. Li, X.; Patel, S.; Büskens, C. Let Hybrid A* Path Planner obey traffic Rules: A Deep Reinforcement Learning-Based Planning Framework. *arXiv* **2024**, arXiv:2407.01216. [CrossRef]
22. Xu, H.; Niu, Z.; Jiang, B.; Zhang, Y.; Chen, S.; Li, Z.; Gao, M.; Zhu, M. ERRT-GA: Expert Genetic Algorithm with Rapidly Exploring Random Tree Initialization for Multi-UAV Path Planning. *Drones* **2024**, *8*, 367. [CrossRef]
23. AbuJabal, N.; Rabie, T.; Baziyad, M.; Kamel, I.; Almazrouei, K. Path planning techniques for real-time multi-robot systems: A systematic review. *Electronics* **2024**, *13*, 2239. [CrossRef]
24. Yang, L.; Li, P.; Qian, S.; Quan, H.; Miao, J.; Liu, M.; Memetimin, E. Path planning technique for mobile robots: A review. *Machines* **2023**, *11*, 980. [CrossRef]
25. Stavrinidis, S.; Zacharia, P. An ANFIS-Based Strategy for Autonomous Robot Collision-Free Navigation in Dynamic Environments. *Robotics* **2024**, *13*, 124. [CrossRef]
26. Stavrinidis, S.; Zacharia, P.; Xidias, E. A Fuzzy Control Strategy for Multi-Goal Autonomous Robot Navigation. *Sensors* **2025**, *25*, 446. [CrossRef] [PubMed]
27. Yin, Y.; Chen, Z.; Liu, G.; Yin, J.; Guo, J. Autonomous navigation of mobile robots in unknown environments using off-policy reinforcement learning with curriculum learning. *Expert Syst. Appl.* **2024**, *247*, 123202. [CrossRef]
28. Zhang, Y.; Zhao, W.; Wang, J.; Yuan, Y. Recent progress, challenges and future prospects of applied deep reinforcement learning: A practical perspective in path planning. *Neurocomputing* **2024**, *608*, 128423. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.