International Journal of Natural and Engineering Sciences
E-ISSN: 2146-0086 19(1): 20-32, 2025
Research Article

Slope-Aware Motion Planning for Autonomous Robots: Efficient Path Optimization in 3D Terrains

Suat KARAKAYA

Kocaeli University, Engineering Faculty, Mechatronics Engineering, Kocaeli, Türkiye

*Corresponding author: suat.karakaya@kocaeli.edu.tr

(Received: 14 January 2025 / Accepted: 06 April 2025)

ORCID 0000-0002-3082-0304

Abstract

Efficient path planning in complex, uneven terrains pose significant challenges for autonomous mobile robots, especially when vertical displacement and energy consumption must be considered. Standard 2D motion planning algorithms often neglect the impact of slope, resulting in suboptimal paths that fail to account for the additional effort required to navigate steep inclines. This paper presents a novel slope-aware motion planning algorithm that incorporates a cost function designed to minimize both the path length and energy expenditure by considering terrain slopes. The algorithm was tested in various 3D terrain scenarios, including single-peak, multi-peak, and random terrains. Simulation results demonstrate the effectiveness of the proposed method in generating paths that avoid steep areas while balancing energy efficiency and coverage. Comparisons between the slope-aware and standard path planning approaches show significant improvements in energy consumption, with the robot successfully navigating challenging terrains with minimal energy use. This approach has promising applications in autonomous navigation for search and rescue missions, agricultural robotics, and planetary exploration, where energy conservation is critical.

Keywords: Terrain map, Boustrophedon, Trajectory planning

Introduction

Motion planning is a well-established field within robotics, with extensive research aimed at improving efficiency, robustness, and adaptability in various environments. Over the years, numerous approaches have been proposed to tackle path planning challenges, particularly for robots operating in complex terrains. Classical algorithms such as A*, Dijkstra's algorithm, and Rapidly-exploring Random Trees (RRT) have long been foundational in motion planning. A* algorithm [1], remains one of the most widely used methods for finding optimal paths in grid-based environments, guaranteeing the shortest path. However, it suffers from efficiency problems in large, complex terrains. Similarly, Dijkstra's algorithm [2] finds the shortest path between nodes but is computationally expensive when applied to large-scale grids due to its exhaustive nature. For dynamic and uncertain environments, RRT and its variants, such as RRT* [3], have become popular due to their ability to handle high-dimensional spaces, although they remain computationally intensive when dealing with terrain features such as slopes.

In recent years, energy-efficient motion planning has emerged as a significant research focus, especially for robots operating in environments where power resources are constrained, such as planetary exploration, autonomous drones, or underwater vehicles. Ishigami et al. [4] and Ferguson and Stentz [5] pioneered the integration of terrain data, such as slope and surface roughness, into path planning to generate energy-optimal routes. By taking terrain resistance and robot dynamics into account, these approaches significantly reduced power consumption. Further developments in energy-efficient navigation, such as those proposed by Sun et al. [6], have incorporated factors such as battery models and environmental influences (e.g., wind resistance), making them particularly relevant in environments where energy efficiency is crucial.

In 3D terrain environments, particularly those with uneven or sloped surfaces, terrain-aware path planning has become increasingly important. Damm et al. [7] and Belter et al. [8] introduced terrain mapping techniques that account for slope, roughness, and obstacles, improving safety and efficiency for robots traversing challenging terrains. These methods employ sensor data, such as LIDAR and stereo vision, to model terrain accurately and predict energy expenditure for each possible move. The Boustrophedon Decomposition algorithm, initially developed by Choset [9], efficiently decomposes the environment into cells to ensure full coverage. While effective for coverage, it does not account for energy consumption or terrain complexity. Our work builds on this concept by integrating slope information into the path planning process, allowing the robot to avoid steep inclines and thereby optimize both coverage and energy usage.

The use of slope data in path optimization has been explored in planetary exploration and terrestrial applications. For instance, Bajracharya et al. [10] highlighted the importance of terrain awareness in NASA's Mars rovers, where slope information played a critical role in ensuring both safety and energy efficiency. Similarly, Hrabar et al. [11] demonstrated how unmanned aerial vehicles (UAVs) could optimize their flight paths by considering terrain elevation and wind conditions. Slope-aware motion planning has also been enhanced through optimization techniques, such as the reinforcement learning-based approach proposed by Liu et al. [12], which enables robots to make real-time decisions to conserve energy by avoiding steep inclines.

Multi-objective path planning, which seeks to optimize multiple criteria simultaneously, has also gained traction in the field. Algorithms proposed by Deb et al. [13] and Samaniego et al. [14] offer ways to balance competing objectives such as distance, energy consumption, safety, and time. These approaches often employ Pareto optimality to determine solutions that best satisfy the trade-offs between objectives. In our work, the balance between minimizing path length and avoiding steep slopes is an essential aspect of energy-efficient path planning.

The goal of any motion planning system is to ensure that a robot follows a trajectory that is both feasible and optimal based on specific constraints such as the robot's kinematics, dynamic constraints, and environmental factors. In this context, several algorithms have been developed for different types of environments, ranging from 2D grid-based maps to more complex 3D environments that include varying terrains. The Boustrophedon motion planning technique is one such approach designed primarily for coverage path planning (CPP), which ensures that a robot covers every accessible area of the environment systematically. It is often used in agriculture, cleaning robots, or surveying robots where complete area coverage is a priority. When dealing with 3D terrain maps, motion planning becomes more complex due to the additional physical properties of the terrain, such as slopes, height variations, and surface roughness. These factors directly affect the energy consumption and feasibility of robot motion. For robots operating in outdoor environments or uneven surfaces, the slope of the terrain plays a crucial role in determining the optimal trajectory.

One of the key limitations of standard 2D motion planning algorithms is their inability to account for the vertical displacement and the effort required to navigate terrain inclines. In real-world applications, particularly in rough or uneven environments, these factors significantly impact energy consumption and overall efficiency. Therefore, to address these challenges, we introduce a cost function that incorporates terrain slope, allowing the robot to plan paths that minimize both energy expenditure and travel distance. This approach ensures that the robot can efficiently cover the area while taking into account the additional constraints posed by the terrain, providing a more practical and energy-conscious solution.

Materials and Methods

The Boustrophedon algorithm is particularly well-suited for area coverage tasks. The name originates from an ancient method of writing where the text alternates directions line by line, just like the pattern that this algorithm follows, covering an area by sweeping back and forth. This approach ensures that each grid or cell is visited in a systematic manner, often optimizing for minimal overlaps and efficient coverage.

For a robot operating on flat terrain, the algorithm divides the environment into a grid of cells and follows a boustrophedon pattern to ensure full coverage. Each time the robot reaches the edge of the environment or an obstacle, it changes direction, ensuring that all accessible spaces are covered.

While the Boustrophedon algorithm works efficiently in 2D environments, 3D terrains present new challenges due to elevation changes. In this context, the algorithm must be extended to not only cover the grid cells but also minimize energy consumption by considering the slope of the terrain.

A key concept in 3D terrain motion planning is the cost function, which evaluates the feasibility of moving from one grid cell to another by accounting for both horizontal and vertical movement. The cost of traveling between cells is influenced by the slope of the terrain, with steeper slopes incurring a higher cost due to the increased energy required for traversal.

In our motion planning scheme, we utilize pre-existing slope data from a 3D terrain map. This allows the robot to calculate the slope between the current position and potential next positions, which is then incorporated into a cost function aimed at minimizing both the total travel distance and the effort required to traverse sloped areas. The final goal is to ensure that the robot can efficiently cover all grid cells while minimizing repeated visits and total energy consumption.

Incorporating Optimization in 3D Terrain Coverage

To optimize the boustrophedon algorithm for 3D terrains, our approach focuses on minimizing the following key factors:

- Total Path Length (L): Minimizing the overall distance traveled is crucial for efficiency.
- Effort (E): This refers to the energy consumed based on the slope of the terrain. Steeper slopes require more energy, and thus the algorithm seeks to minimize movement across high-slope regions.
- Coverage Efficiency (G): Ideally, each cell should be visited only once. If a cell must be revisited, this adds to the cost, and the algorithm should minimize such occurrences.

These optimization goals are expressed through a comprehensive cost function that the robot uses to evaluate its next moves. The cost function integrates these components and assigns a weight to each based on its priority. For instance, if energy efficiency is paramount in a given scenario, the weight assigned to effort would be higher than that assigned to path length.

One of the primary advantages of this approach is that it assumes the availability of a pre-existing 3D terrain map. Since the terrain's slope information is already known, the robot does not need to rely on real-time data from sensors like an IMU (Inertial Measurement Unit) to determine the slope. Instead, it can directly use the 3D map to evaluate potential next steps and calculate the slope for each move option. This allows the motion planning algorithm to make informed decisions and minimize the computational overhead typically required for real-time slope estimation.

In summary, the technical background for this motion planning scheme is based on adapting the classic boustrophedon algorithm for a 3D terrain environment, where slope data is utilized to minimize both path length and energy consumption. By incorporating a well-structured cost function, the robot is able to achieve efficient area coverage with minimal overlap and maximum energy savings.

3D Terrain Mapping and Grid Representation

3D terrain mapping typically involves dividing the environment into a grid of cells, where each cell contains information about its position (x, y, z) and slope. In our algorithm, the terrain is discretized into a uniform grid, and each cell stores pre-calculated slope data, which will be used to determine the effort required for motion planning.

In a typical grid-based representation, the following attributes can be stored:

- Grid Cell State: Indicates whether the cell is free or occupied (obstacle).
- Slope Information: Pre-determined slope value for each cell.
- Visited Status: Tracks whether the robot has visited the cell.

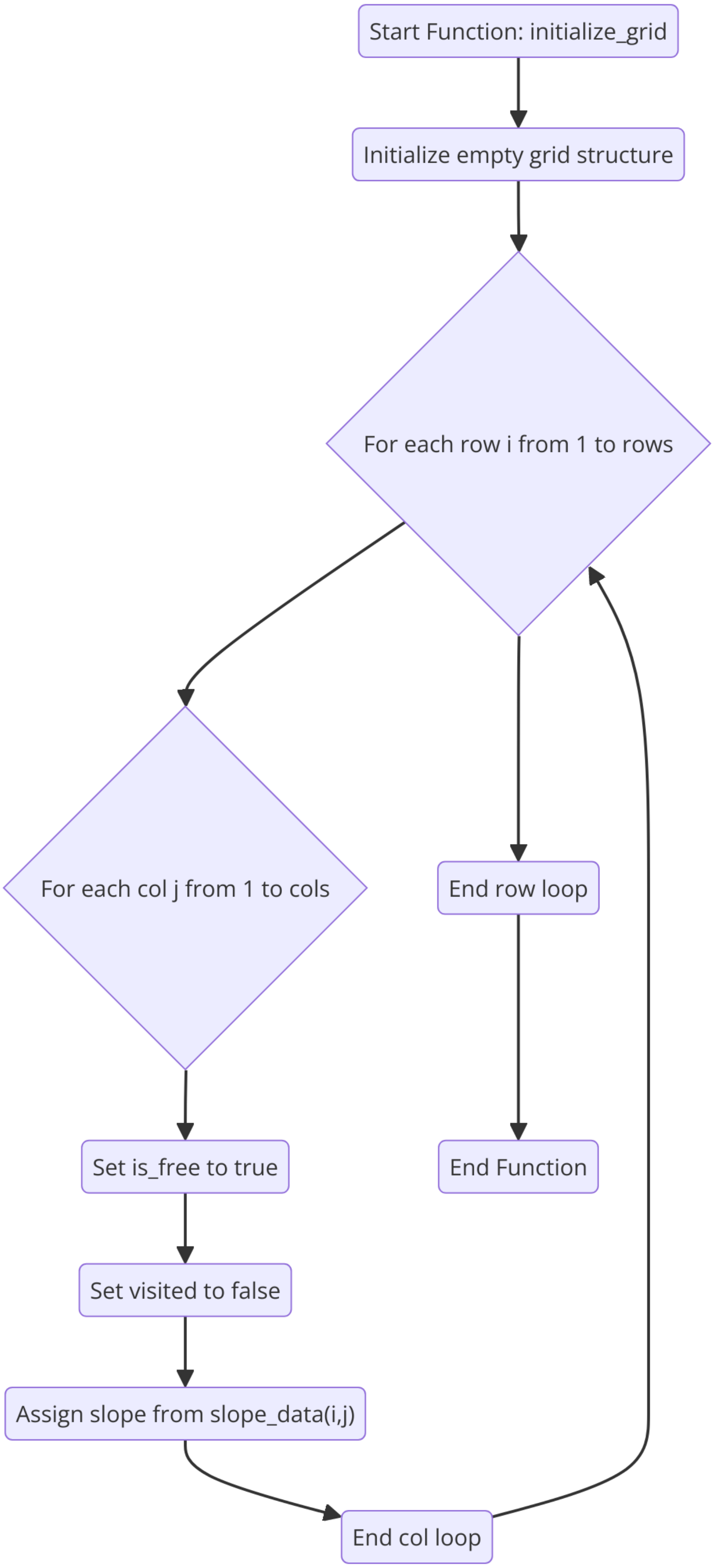

Figure 1. *3D Terrain Grid Initialization*

Cost Function Design

The cost function plays a crucial role in guiding the robot through the terrain in the most efficient manner. In our approach, the cost function accounts for three main factors:

1. Path Length (L): The distance traveled by the robot.
2. Effort (E): The energy consumed, which depends on the slope.
3. Revisit Penalty (G): A penalty for visiting the same cell multiple times.

These components are combined into a single objective function as:

$$J = \alpha L + \beta E + \gamma G$$

Equation 1

Where α, β and γ are weighting factors that prioritize different aspects of the cost.

This function is designed to balance three critical factors in the robot's motion planning process: the distance traveled, the energy expenditure due to terrain slope, and additional environmental factors that may impact the robot's performance. Each term in the equation represents a specific cost component, weighted by parameters α, β and γ which allow for fine-tuning the robot's behavior based on the mission's priorities.

- **L (Distance/Path Length)**: This term represents the total distance traveled by the robot. Minimizing the path length is important for reducing both time and energy consumption in scenarios where the terrain is relatively flat or when rapid completion of the mission is critical. The weighting factor α controls how much emphasis is placed on minimizing the path length in the overall cost. A larger α encourages the robot to prioritize shorter paths.
- **E (Slope/Effort Due to Slope)**: This term accounts for the energy required to navigate varying slopes in the terrain. Steeper slopes lead to increased energy consumption, which must be factored into the cost function. The weighting factor β adjusts the sensitivity of the cost to changes in slope. A higher β value will cause the robot to avoid steep inclines, favoring paths that require less energy to traverse, even if they may be longer.
- **G (Environmental/Other Costs)**: This term represents additional costs associated with environmental factors such as terrain roughness, obstacles, or hazardous areas. These factors may significantly impact the robot's safety and efficiency. The parameter γ\gammaγ determines the influence of these factors in the overall cost. Higher γ values cause the robot to prioritize avoiding risky or challenging terrain.

The cost function is evaluated at each decision point, where the robot must choose its next move. The function ensures that the robot selects the move that minimizes the total cost J, taking into account the distance to the next cell, the slope of the terrain, and any environmental constraints. By adjusting the values of α, β and γ the planner can be tailored to emphasize different aspects of the mission, such as energy efficiency, safety, or speed. This cost function provides a flexible and robust framework for autonomous navigation in challenging environments. By adjusting the weighting factors, the algorithm can be tailored to prioritize energy efficiency, safety, or speed based on the specific requirements of the mission. In all cases, the robot can balance these competing factors to achieve optimal performance while traversing complex terrains.

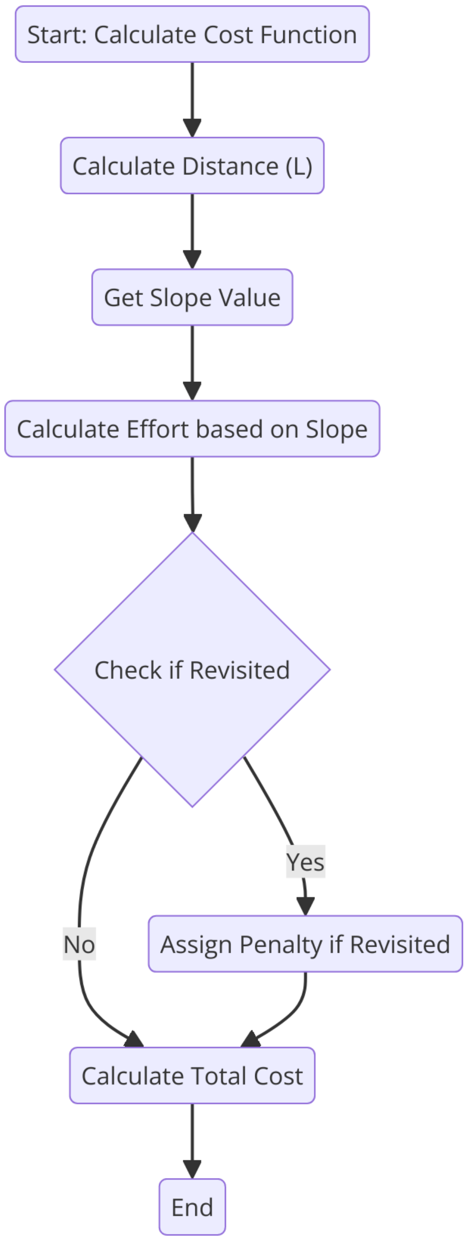

Figure 2. *Cost Function Calculation*

Boustrophedon Path Planning in 3D Terrain

The traditional boustrophedon algorithm can be extended to consider slope when making movement decisions. The robot systematically covers the grid by sweeping back and forth, but at each step, it evaluates the cost of moving to neighboring cells based on slope and other factors.

Pseudo Code: *Boustrophedon Motion Planning with Slope Consideration*

function boustrophedon_coverage(grid, slope, start)

initialize visited grid;

current_position = start;

while not all cells visited

neighbors = get_neighbors(current_position, grid);

```
        best_next_move = find_min_cost_move(neighbors, slope, visited);
        current_position = best_next_move;
        mark_cell_visited(current_position);
    end
    display "Coverage complete!";
end
```

Results and Discussion

The final section involves running simulations to evaluate the effectiveness of the proposed algorithm. We can create different test scenarios with varying terrain slope profiles and compare the performance in terms of path length, energy consumption, and number of grid cells visited.

Table 1 summarizes the performance of the proposed motion planning algorithm across various terrain scenarios. The key metrics evaluated include total path length, total effort (considering slope and energy consumption), number of visited cells, and the proportion of revisited cells, among others. These metrics provide a comprehensive comparison of the algorithm's ability to efficiently navigate challenging terrains while minimizing energy expenditure. In each case, the robot successfully avoided steep inclines, demonstrating the effectiveness of the slope-weighted cost function in balancing path length and energy efficiency. The results highlight the trade-offs between minimizing travel distance and conserving energy, which are crucial for autonomous robots operating in uneven environments.

Table 1. *Performance Metrics of Slope-Aware Terrain Scenarios*

Map	Total Path Length(cm)	Total Effort	Visited Cells	Revisited Cells	Max Slope (Degree)	Average Slope	Total Visit Time (ms)
One Peak	400	475	400	0	30	1.875	400
Three Peaks	400	529	400	0	30	3.225	400
Five Peaks	400	676	400	0	45	6.9	400
Random Map 1	400	804.6690765	400	0	19.99187087	10.11672691	400
Random Map 2	400	769.5749502	400	0	19.9863208	9.239373755	400

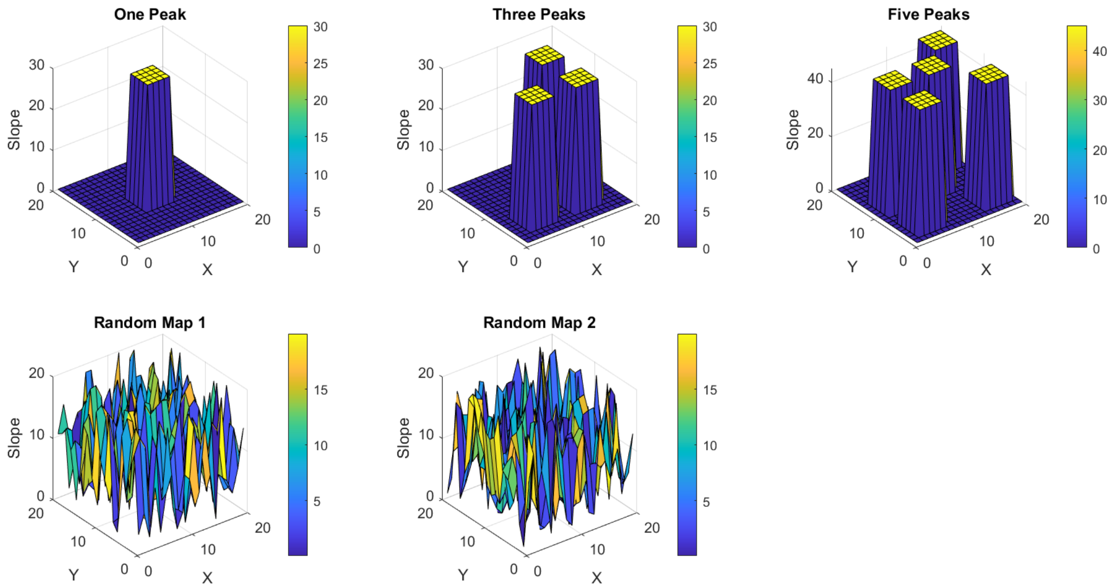

Figure 3. *3D Terrain Models Used for Slope-Aware Motion Planning Simulations*

Performance Metrics Calculation

In the simulation of robot motion over a 3D terrain grid, several performance metrics are essential for evaluating the efficiency and effectiveness of the algorithm. The following sections outline the mathematical formulas used to calculate these metrics:

- ***Total Path Length***

The Total Path Length metric represents the total distance the robot travels over the grid. Since the Boustrophedon algorithm ensures that every cell is visited, the total path length is calculated as:

$$Total\ Length = N_{rows} \times N_{cols}$$

Equation 2

Where:

- N_{rows} is the number of rows in the grid.
- N_{cols} is the number of columns in the grid.

This value represents the total number of cells the robot traverses during the coverage.

- ***Total Effort***

Total Effort refers to the energy expended by the robot as it traverses the grid. This energy depends on the slope of the terrain: steeper slopes require more energy. The total effort is calculated as:

$$Total\ Effort\ = \sum_{i=1}^{n}\left(1 + \frac{slope(i)}{10}\right)$$

Equation 3

Where

- n is the is the total number of cells visited by the robot.
- $slope(i)$ represents the slope value of the i-th cell.

- $1 + \frac{slope(i)}{10}$ represents the cost of moving through the cell, where the slope increases the energy cost.

- ***Visited Cells***

The Visited Cells metric indicates how many unique cells the robot visits at least once during its operation.

- ***Revisited Cells***

Revisited Cells refers to the number of cells visited more than once by the robot. This metric is important in evaluating the efficiency of the coverage algorithm, as ideally, each cell should only be visited once.

- ***Max Slope***

The Max Slope metric captures the steepest slope the robot encounters during its traversal. This value is critical for understanding the most challenging terrain the robot must navigate.

- ***Average Slope***

The **Average Slope** provides a measure of the average terrain difficulty the robot encounters over its entire path. It is computed as the mean slope of all cells visited by the robot.

- ***Total Visit Time***

Total Visit Time measures the total time the robot spends traversing the grid. In this case, we assume that the robot spends a constant amount of time in each cell. Therefore, the total visit time is directly proportional to the total number of cells visited,

Where the Total Path Length is the total number of cells traversed by the robot.

Example Path Context Under Steep Terrain

In the simulation, the robot's path was planned using a slope-weighted cost function, which prioritized energy efficiency by avoiding steep slopes. As illustrated in Figure 4, the robot did not always follow the shortest path, especially in areas where steep inclines were present. Instead, the algorithm extended the path to bypass regions with higher slopes, resulting in a longer overall trajectory.

In Figure 4, the red path represents the actual trajectory taken by the robot, while the black-highlighted areas indicate regions of steep slopes. It is evident that, rather than climbing directly through steep regions, the robot opted for lower-slope routes, even though this increased the overall path length.

This demonstrates the effectiveness of the slope-weighted cost function in ensuring that the robot minimizes energy consumption in hilly or uneven terrains. The decision to extend the path rather than climb directly up steep inclines reflects the algorithm’s ability to intelligently balance the trade-off between minimizing distance and reducing energy expenditure.

In real-world applications, such behavior is crucial for autonomous robots operating in energy-constrained environments, where efficiency in both distance and energy usage is critical. The ability to make such trade-offs highlights the flexibility of the algorithm and its adaptability to challenging terrain conditions.

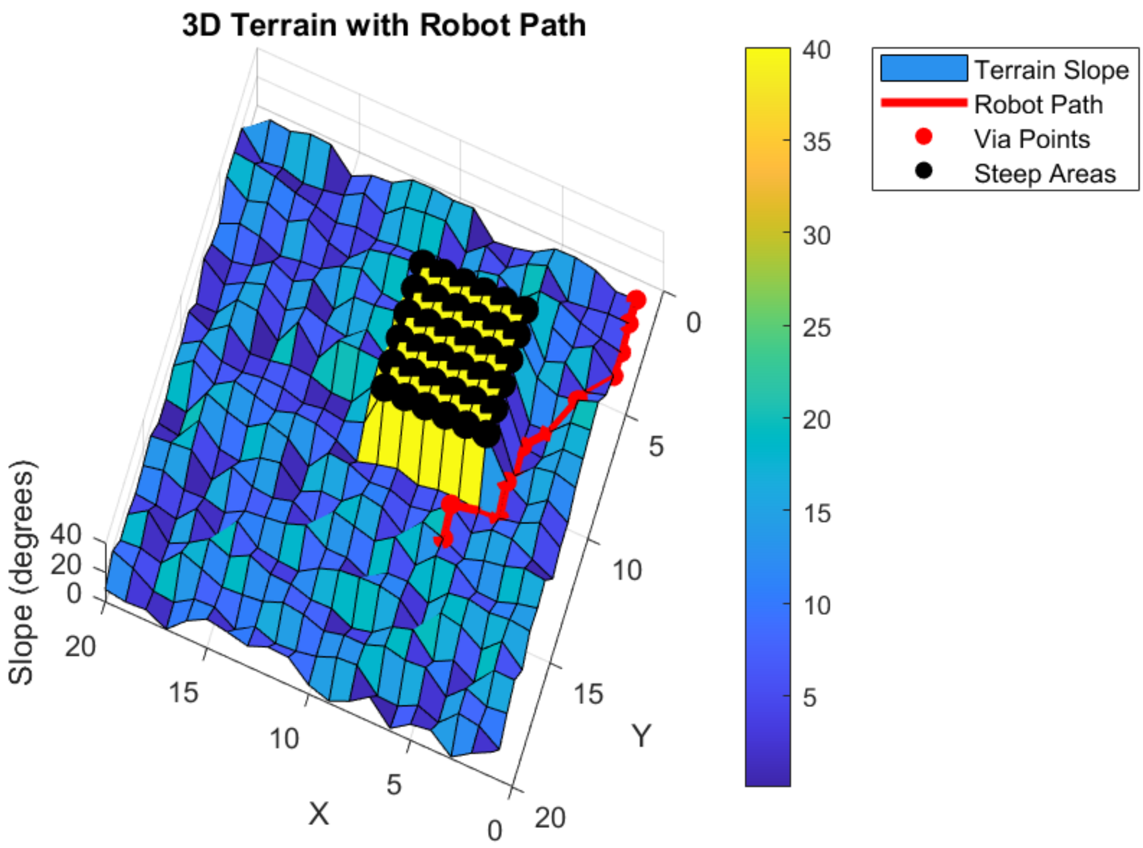

Figure 4. *3D Terrain with Slope-Aware Robot Path*

Figure 4 shows the robot’s slope-aware path (red line), where it deliberately avoids steep areas (marked in black) to minimize energy expenditure. The terrain's slope is represented by the color gradient, with steeper slopes shown in yellow.

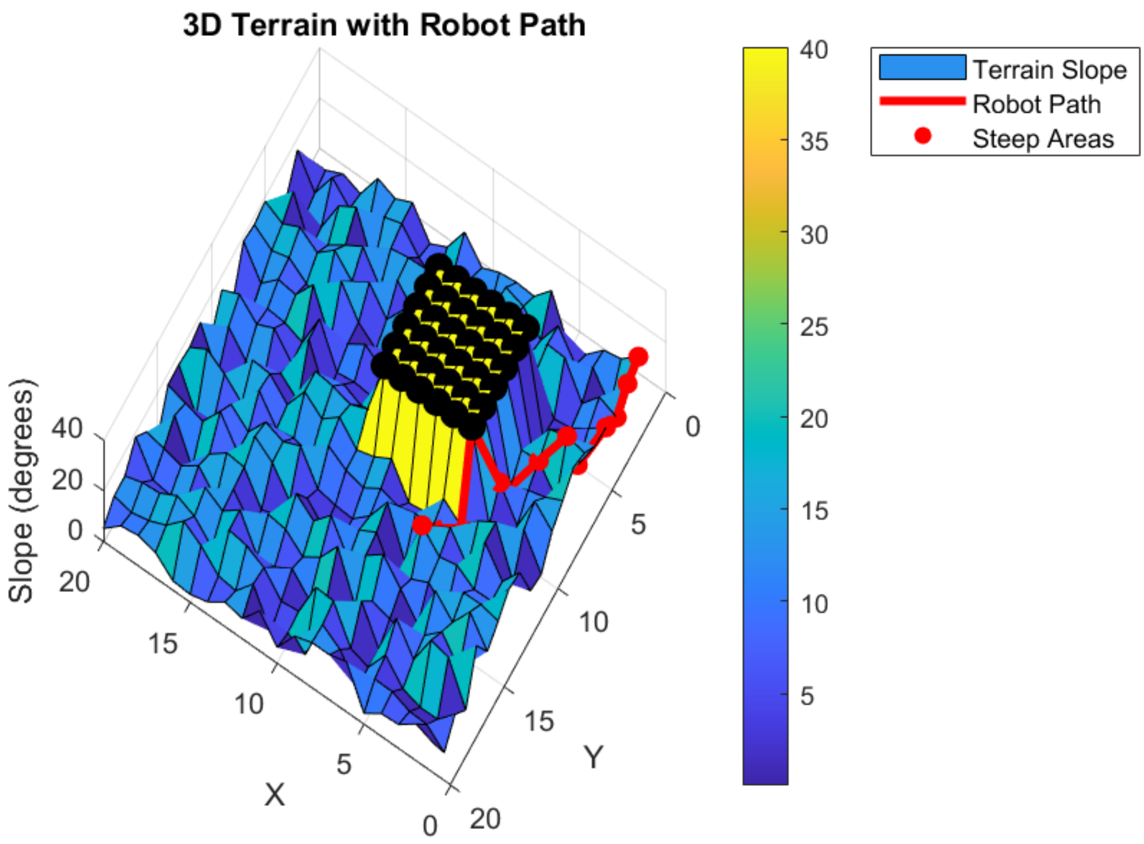

Figure 5. *3D Terrain with Standard Robot Path*

Figure 5 depicts the standard robot path (red line), where the robot does not prioritize avoiding steep areas. The path includes via points (red dots), and the terrain slope is visualized with the color gradient, indicating steep regions in yellow.

Conclusion

In this work, we presented a motion planning algorithm that integrates slope awareness into the path optimization process, focusing on balancing path length with energy efficiency in 3D terrain environments. By prioritizing slope as a significant factor in the cost function, we demonstrated that the robot can avoid steep inclines, even if it means extending its overall path. This trade-off between distance and energy expenditure is crucial in real-world scenarios where energy conservation is vital, particularly for mobile robots operating in uneven and challenging terrains. The simulation results show that the robot successfully navigates the terrain, avoiding high-energy cost paths while ensuring complete coverage of the grid. The robot's ability to extend its path to avoid steep slopes was clearly illustrated, confirming that the slope-weighted cost function provides a more efficient path when energy expenditure is a priority.

Through detailed analysis and visualization, we demonstrated how the robot's path deviates from the shortest route to avoid energy-intensive regions. This behavior emphasizes the flexibility and adaptability of the proposed algorithm in handling terrain variations, making it highly suitable for applications such as autonomous exploration, search and rescue operations, and agricultural tasks in outdoor environments.

Future Work

While the current approach effectively minimizes energy expenditure by avoiding steep slopes, there are several avenues for future research to enhance the capabilities of this motion planning system. Some potential areas of improvement and exploration include:

- Incorporating Dynamic Terrain Conditions: Future iterations of this work could consider terrain that changes dynamically over time, such as shifting sand or mud, which could alter the robot's path planning strategy. Real-time terrain analysis could be integrated to adapt the path continuously based on updated environmental data.

- Multi-Objective Optimization: Although this work focused on minimizing energy consumption by balancing path length and slope, future research could explore multi-objective optimization approaches. These could include additional factors such as time constraints, terrain roughness, and environmental hazards, making the algorithm even more robust for real-world applications.

- Path Planning for Multi-Robot Systems: Extending the current approach to handle multi-robot coordination could open up possibilities for cooperative coverage of large terrains. The coordination of multiple robots while avoiding steep slopes and minimizing overlap could be an exciting direction for future work.

- Testing on Physical Robots in Real-World Environments: While our current study was conducted in a simulated environment, future work should focus on deploying this algorithm on physical robots in real-world terrains. Field experiments could provide valuable insights into the practical challenges and performance of the proposed method in actual conditions, leading to further refinement of the system.

- Improving Computational Efficiency: As the terrain size and complexity increase, the computational demands of real-time path planning can become significant. Optimizing the algorithm's performance, perhaps through parallel computing techniques or heuristic methods, could enhance its scalability for larger, more complex environments.

References

[1] Lamini C, Benhlima S, Elbekri A. (2018). Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning. *Procedia Computer Science*. 127: 180-189.

[2] Dijkstra EW. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*. 1(1): 269-271.

[3] Samperi K, Hawes N. (2014). Improving the Generation of Rapidly Exploring Randomised Trees (RRTs) in Large Scale Virtual Environments Using Trails. In: Mistry M, Leonardis A, Witkowski M, Melhuish C, editors. *Advances in Autonomous Robotics Systems*. Cham: Springer. vol. 8717, pp. 221-233.

[4] Ishigami G, Nagatani K, Yoshida K. (2007). Path Planning for Planetary Exploration Rovers and Its Evaluation Based on Wheel Slip Dynamics. *Proceedings of the IEEE International Conference on Robotics and Automation*. Rome, Italy, pp. 2361-2366.

[5] Ferguson D, Stentz A. (2006). Field D*: An Interpolation-Based Path Planner and Replanner. *Proceedings of the 19th International Conference on Robotics and Automation*.

[6] Sun H, Yang G, Zhang L. (2017). Energy-Efficient Path Planning for Solar-Powered UAVs in Uncertain Environments. *IEEE Transactions on Control Systems Technology*. 25(1): 258-269.

[7] Damm ER, Gregory JM, Lancaster ES, Sanchez FA, Sahu DM, Howard TM. (2023). Terrain-Aware Kinodynamic Planning with Efficiently Adaptive State Lattices for Mobile Robot Navigation in Off-Road Environments. *Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Detroit, MI, USA.

[8] Belter D, Łabęcki P, Skrzypczyński P. (2015). Adaptive Motion Planning for Autonomous Rough Terrain Traversal with a Walking Robot. *Journal of Field Robotics*. 33(3): 337-357.

[9] Choset H. (2001). Coverage for Robotics: A Survey of Recent Results. *Annals of Mathematics and Artificial Intelligence*. 31(1): 113-126.

[10] Bajracharya S, Maimone MW, Helmick D. (2008). Autonomy for Mars Rovers: Past, Present, and Future. *Computer*. 41(12): 44-50.

[11] Hrabar I, Goričanec J, Kovačić Z. (2021). Towards Autonomous Navigation of a Mobile Robot in a Steep Slope Vineyard. *Proceedings of the 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*. Opatija, Croatia, pp. 1119-1124.

[12] Liu Z, Wang Q, Yang B. (2022). Reinforcement Learning-Based Path Planning Algorithm for Mobile Robots. *Wireless Communications and Mobile Computing*. 2022: 1-10.

[13] Deb K, Pratap A, Agarwal S, Meyarivan T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 6(2): 182-197.

[14] Samaniego F, Sanchis J, Garcia-Nieto S, Simarro R. (2019). Smooth 3D Path Planning by Means of Multiobjective Optimization for Fixed-Wing UAVs. *Electronics*. 9(1): 51.