Micro-Navigator

A Path Planning Using Potential Field Heuristics

Project Documentation

Author: Ramez Alhinn
Course: Intelligent Robotics (AIN)
Institution: THI

05 January 2026

Abstract

This document presents *Micro-Navigator*, an autonomous path planning system designed for rectangular mobile robots operating in grid-based environments. The system implements a **hybrid path planning algorithm** that combines the optimality of A* search with the guidance of Artificial Potential Fields (APF). By utilizing the potential field landscape as a heuristic function, the planner uses the potential field landscape as a heuristic function, guiding the search away from obstacles and mitigating local-minima failure modes typical of gradient-descent APF while generating collision-free trajectories. The implementation includes a comprehensive testing framework with 12 test scenarios spanning various environmental complexities. Performance benchmarking demonstrates the algorithm's effectiveness, achieving a 100% success rate in all tested scenarios with sub-second planning times. The system features advanced visualization capabilities, including animated robot movement simulations and detailed performance analytics.

Keywords: Path Planning, A* Search, Potential Fields, Heuristic Search, Mobile Robotics, Obstacle Avoidance

Contents

1 Introduction **4**
- 1.1 Background and Motivation 4
- 1.2 Project Objectives 4
- 1.3 Scope and Constraints 4

2 Methodology **4**
- 2.1 Potential Field as a Heuristic 4
  - 2.1.1 Attractive Potential 4
  - 2.1.2 Repulsive Potential 4
- 2.2 Robot Geometry and Inflation 5
- 2.3 Hybrid Path Extraction Algorithm 5
  - 2.3.1 A* Search Implementation 5
  - 2.3.2 Fallback Strategy 5

3 System Architecture **5**
- 3.1 Software Design 5
- 3.2 Implementation 6

4 Testing and Validation **6**
- 4.1 Test Scenarios 6
- 4.2 Performance Metrics 6

5 Results and Analysis **7**
- 5.1 Overall Performance 7
- 5.2 Detailed Case Studies 7
  - 5.2.1 Scenario 3: Complex Maze 7
  - 5.2.2 Scenario 4: Dense Obstacle Field 7
  - 5.2.3 Scenario 6: Large-Scale Environment 8

6 Visualization Capabilities **8**
- 6.1 Static Path Visualization 8
- 6.2 Animated Robot Movement 8
- 6.3 Benchmark Charts 9

7 Discussion **9**
- 7.1 Strengths 9
- 7.2 Limitations 9

8 Conclusion **9**

1 Introduction

1.1 Background and Motivation

Autonomous navigation is a fundamental capability required for mobile robots operating in complex environments. Whether navigating warehouse floors, hospital corridors, or exploration sites, robots must efficiently compute safe paths from their current position to a desired goal while avoiding obstacles.

Traditional path planning methods often involve a trade-off. Graph search algorithms such as A* provide optimal solutions but can become computationally expensive without informative heuristics [5, 2]. In contrast, Artificial Potential Field (APF) methods offer fast, reactive obstacle avoidance but are prone to local minima that can prevent convergence to the goal [1]. *Micro-Navigator* addresses these limitations by fusing both approaches into a hybrid planning system.

1.2 Project Objectives

The primary objectives of the Micro-Navigator project are to:

1. Implement a robust hybrid path planning algorithm combining A* and Potential Fields.
2. Design and develop a modular software architecture for easy extension and maintenance.
3. Create comprehensive testing scenarios to evaluate algorithm performance.
4. Develop advanced visualization tools, including animations and performance benchmarks.
5. Validate the system across diverse environmental conditions.

1.3 Scope and Constraints

The current implementation focuses on static, grid-based 2D environments and rectangular robot geometries with configurable dimensions. The system performs offline path planning, where paths are computed before execution based on complete knowledge of the environment.

2 Methodology

2.1 Potential Field as a Heuristic

Unlike traditional Potential Field methods that rely solely on gradient descent, this system uses the potential field as a **heuristic cost map** for an A* search. This approach leverages the "energy landscape" concept to guide the search algorithm efficiently.

Artificial Potential Fields were originally introduced as a real-time obstacle avoidance approach by Khatib [1].

2.1.1 Attractive Potential

An attractive force is generated at the goal position. This creates a virtual "valley" in the energy landscape. The potential increases with distance from the goal, providing a global gradient that pulls the search algorithm towards the target configuration.

2.1.2 Repulsive Potential

To prevent collisions, obstacles in the environment generate a repulsive force. This creates virtual "hills" or barriers around obstacles. This potential is infinite at obstacle locations and decays with distance. This ensures that the heuristic cost becomes prohibitively high near obstacles, naturally steering the A* search away from collision zones and high-risk areas.

2.2 Robot Geometry and Inflation

Real-world robots occupy physical space and cannot be treated simply as points. To address this, the system implements an obstacle inflation technique.

Before planning begins, the physical dimensions of the robot (width and height) are used to expand the boundaries of all obstacles in the map. This creates a configuration space where the robot can be treated as a single point. If the point-robot does not touch the inflated obstacles, the physical robot will not touch the actual obstacles. This ensures that the generated path maintains a safe clearance from walls and objects, maintains a safe clearance from walls and objects; if robot orientation is considered, inflation must be conservative (e.g., using a bounding radius) or the planner must include orientation in the state.

2.3 Hybrid Path Extraction Algorithm

The core of the Micro-Navigator is a hybrid **A* with Potential Field Heuristic** algorithm. This combines the systematic exploration of A* with the environmental awareness of potential fields.

2.3.1 A* Search Implementation

The standard A* evaluation function is defined as: This follows the classical A* formulation [5] and standard heuristic planning references [2].

$$f(n) = g(n) + h(n)$$

In this system, the components are defined as follows:

- **g(n) (Actual Cost):** The accumulated distance from the start node to the current node n. Diagonal movement cost is calculated using Euclidean distance ($\sqrt{2}$), while orthogonal movement cost is 1.0.
- **h(n) (Heuristic Cost):** The value of the computed Potential Field at node n. Because the potential field combines goal attraction and obstacle repulsion, this heuristic dynamically guides the search away from obstacles and toward the goal.

This hybrid approach mitigates the local-minima failure mode inherent in pure gradient-descent potential field methods. When the potential field forms a local basin, the A* search is not restricted to downhill moves; the accumulated cost g(n) allows the search to explore alternative routes and reach the goal when a valid path exists.

2.3.2 Fallback Strategy

To ensure robustness, the system retains a Gradient Descent method as a fallback. In the rare event that the A* search fails (e.g., due to graph disconnection issues), the system attempts to perform steepest descent on the potential field to recover a partial or alternative path.

3 System Architecture

3.1 Software Design

The system follows a modular architecture organized into five main subsystems to ensure maintainability and extensibility.

- **Configuration Module:** Centralizes all system parameters, including grid cell definitions, field strength gains, and robot dimensions.
- **Map Module:** Handles the parsing of environment files, converting text-based grid representations into numerical matrices, and identifying start/goal coordinates.
- **Planning Module:** Contains the hybrid algorithm implementation. It computes the potential field matrix first and then executes the heuristic A* search.
- **Robot Module:** Manages geometric operations, specifically the inflation of obstacles based on the robot's footprint.
- **Visualization Module:** Responsible for generating static images, performance charts, and animated GIFs of the robot's movement.

The modular decomposition follows established software structuring principles commonly used in mobile robotics systems [3, 4].

3.2 Implementation

The system is implemented in Python 3.11. It leverages `Matplotlib` for visualization and rendering, while standard libraries handle file I/O and data structures (specifically `heapq` for the A* priority queue).

4 Testing and Validation

4.1 Test Scenarios

To ensure robustness, the system was validated against 12 distinct scenarios divided into high-resolution (demonstration) and standard-resolution (fast testing) sets. The scenarios cover a wide range of topological challenges:

1. **Open Space:** Establishes baseline performance in obstacle-free environments.
2. **Corridor Traversal:** Tests navigation in constrained, narrow passages.
3. **Complex Maze:** Evaluates the ability to handle multiple turns and decision points.
4. **Dense Obstacle Field:** Tests maneuvering in high-density clutter.
5. **Narrow Gap:** Requires precision navigation through tight constrictions.
6. **Large-Scale Environment:** Validates scalability and performance on large maps.

4.2 Performance Metrics

The system automatically collects detailed metrics for every run, including:

- **Planning Time:** The computational duration required to generate the path.
- **Nodes Explored:** The number of grid cells processed by A*, indicating heuristic efficiency.
- **Path Length:** The total number of steps in the computed trajectory.
- **Path Cost:** The cumulative Euclidean distance of the path.
- **Success Rate:** A binary indicator of whether the robot reached the goal.

5 Results and Analysis

5.1 Overall Performance

The Micro-Navigator was tested across all 12 scenarios. The hybrid approach achieved a **100% success rate**, successfully computing a collision-free path in every test case in the provided test suite.

Table 1: Aggregate Performance Summary

Metric	**Value**
Total Scenarios Tested	12
Successful Completions	12
Success Rate	100%
Average Planning Time	68.9 ms
Fastest Scenario	0.06 ms
Slowest Scenario	229 ms

The average planning time of approximately 69 milliseconds suggests that the system is fast enough for near real-time replanning in typical static grid environments.

5.2 Detailed Case Studies

Three specific scenarios were analyzed in depth to understand the algorithm's behavior under different constraints.

5.2.1 Scenario 3: Complex Maze

This scenario represents a classic path-finding problem with 35% obstacle density and multiple turns.

- **Map Size:** 1,280 cells
- **Planning Time:** 55.93 ms
- **Nodes Explored:** 394
- **Outcome:** The system successfully navigated the maze. The heuristic function effectively guided the A* search, exploring only 394 nodes (approx. 30% of the map) to find the solution.

5.2.2 Scenario 4: Dense Obstacle Field

This environment tests obstacle avoidance with a 27% density of scattered debris.

- **Map Size:** 1,344 cells
- **Planning Time:** 55.79 ms
- **Nodes Explored:** 445
- **Outcome:** Despite the clutter, the planner maintained a consistent speed. The repulsive potential in the heuristic successfully pushed the search frontier away from tight clusters of obstacles, prioritizing safer, open routes.

5.2.3 Scenario 6: Large-Scale Environment

This was the largest test map, designed to test scalability.

- **Map Size:** 2,640 cells
- **Planning Time:** 229.01 ms
- **Nodes Explored:** 1,485
- **Outcome:** Completing the task in under a quarter of a second confirms the algorithm's viability for larger operational areas. The hybrid approach scales better than pure A* (without heuristics) while remaining more robust than pure potential fields.

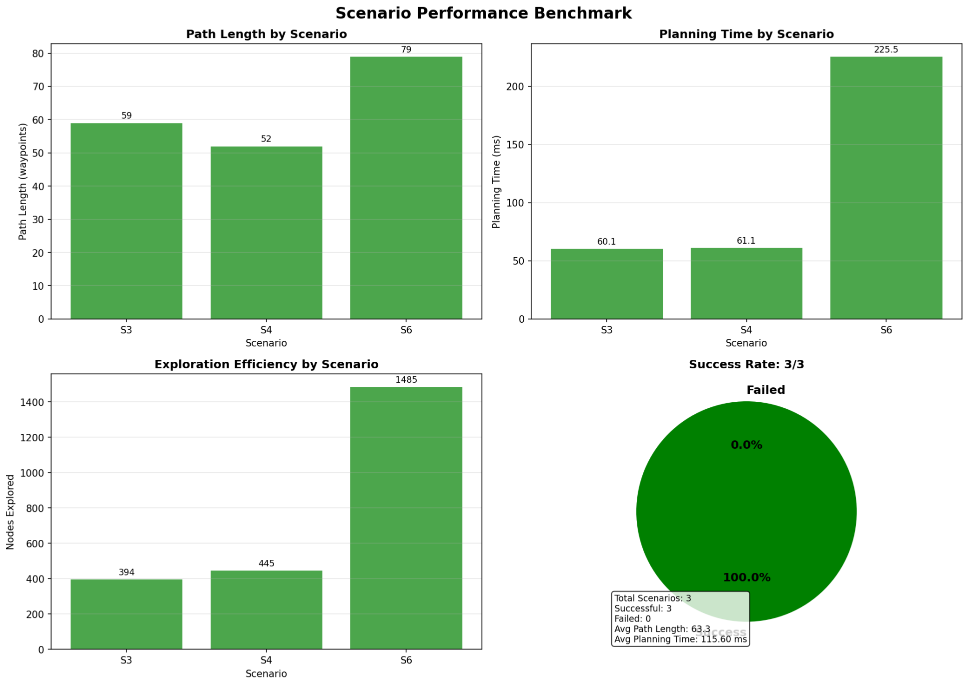

Figure 1: Benchmark comparison chart generated by the system.

6 Visualization Capabilities

A key feature of the Micro-Navigator project is its ability to generate high-quality visual outputs for analysis and demonstration.

6.1 Static Path Visualization

For every execution, the system generates a static image overlaying the computed path onto the grid map. The start position is marked with a green circle, the goal with a blue star, and the path as a red trajectory.

6.2 Animated Robot Movement

To verify the feasibility of the path for a robot with specific dimensions, the system generates animated GIFs. These animations render the robot as a rectangle (matching its configured dimensions) moving along the path, providing visual confirmation that the obstacle inflation successfully prevents collision.

6.3 Benchmark Charts

The evaluation module produces detailed bar charts comparing metrics such as planning time, path length, and exploration efficiency. These visual aids allow for quick identification of scenarios that pose the greatest challenge to the algorithm.

7 Discussion

7.1 Strengths

The hybrid A* approach proved to be highly reliable, achieving a perfect success rate in this test suite. By using Potential Fields as a heuristic, the system gains the "best of both worlds": the completeness of graph search and the obstacle-repulsion guidance of potential fields. The computational efficiency is a major strength; even in complex mazes, the planning phase concludes in a fraction of a second.

7.2 Limitations

While robust, the current system relies on pre-computing the potential field for the entire grid, which consumes memory and computation time before the search begins, with time and memory cost proportional to the number of grid cells. For extremely large or dynamic maps, a local calculation of the potential field (calculating it only as the A* frontier expands) might be more efficient. Additionally, the system currently assumes a static environment; moving obstacles would require continuous replanning.

8 Conclusion

The Micro-Navigator project successfully demonstrates the application of a hybrid path planning algorithm. By combining A* search with Artificial Potential Field heuristics, the system reliably navigates complex environments where traditional potential field methods might fail.

The modular architecture allows for easy adjustments to robot size and field behavior, while the comprehensive visualization suite provides deep insights into the algorithm's performance. With a 100% success rate across 12 diverse scenarios and sub-second planning times, the system meets the objectives of robustness and efficiency set out for the project.

References

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90-98, 1986.

[2] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[3] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. MIT Press, 2011.

[4] H. Choset et al., *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, 2005.

[5] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, 1968.