Navigational algorithms evaluation and benchmarking

Mohamed Osama Idries

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Mathematics

Artificial Intelligence and Robotics Group
Department of Computing and Communication Technologies
School of Engineering, Computing and Mathematics
Oxford Brookes University

Supervised by :
Dr. Tjeerd v. Olde Scheper and Dr. Matthias Rolf

September 2022

Acknowledgements

First and foremost, I would like to acknowledge and give my heartfelt thanks to my wife *Lourdes* for her unlimited patience, love, and support through the years. You have enjoyed my success as your own; none of this would be possible without you.

I would also like to thank my supervisors, Dr Tjeerd v. Olde Scheper and Dr Matthias Rolf, for their continuous support, wisdom, and advice. Your guidance and advice motivated me and created many memorable and enjoyable moments.

I would like to acknowledge my gratitude to my friends whose companionship has supported me through this academic endeavour, Gurkirt Singh, Kamil Wyczynski, Indroneel Chatterjee, Mireya Munoz Balbontin, Bedour Alshaigy, Meenakshi Singh, and Noemi Dreksler.

I want to convey my heartfelt appreciation and gratitude to Blackbox and, in particular, Nienke Hensbroek and Ian Turner for their kindness, support, encouragement, and understanding when undertaking and writing my research project.

Finally, my eternal gratitude to my family. The bedrock on which everything is built and without whom none of my success would be possible.

Abstract

One of the fundamental problems in mobile robotics is navigating unexplored environments safely and efficiently. Efforts to address this issue are classified into three categories: reactive-based approaches, which make instantaneous decisions; map-based approaches, involving grid or topological representations; and learning-based approaches. Evaluating and comparing approaches is essential to better understand them, particularly in how they perform in different problem environments and in relation to each other. This information serves to guide the development of further approaches, highlight problem environments, and provide a clear mapping between approaches and environments. However, current comparative studies within a single category have been limited by the existence of a degree of similarity between the different approaches. There has not yet been a comparative framework across different categories in navigational robotics. Thus, this work aims to develop an evaluation method to compare a variety of different approaches in the same environment to achieve a better understanding of navigational algorithms. To this end, a framework has been proposed that simulates these approaches in a common set of problem environments and evaluates them with the same set of metrics to compare their effectiveness and efficiency. The most common reactive and map-based approaches are implemented and a generic, precise, and empirical way to evaluate their performance to the set of environments they are in and compared to the other different approaches is demonstrated. The resulting analysis shows that methods like RRT* don't improve on the RRT when benchmarked and the evaluation of the problem areas of the Potential Field approach led to the development of the novel Pheromone Potential Field approach. This work opens the doors to more in-depth research into benchmarking across the different navigational categories in static and dynamic environments, which will result in a better understanding and significantly impact the future development of navigational approaches. This research is a step toward dynamic navigational planners that match the different approaches to a set of problems or environments.

Contents

List of Figures

List of Tables

Nomenclature

Approaches

A*: A Dijkstra-based heuristic path search algorithm

FOV: Field Of Vision

PPF: Pheromone Potential Field Pheromone Field

PF: Potential Field

PRM: Probabilistic Road Map approach

RRT: Rapidly Exploring Random Tree approach

Table Abbreviations

Time Sc: Computational Time Score

Path Sc: Path Length Score

Smooth Sc: Path Smoothness Score

Std: Standard Deviation

Vis Sc: Visibility Score

Framework Specific

Environment A map type defined by a set of rules

Map A defined representation of an environment

Experiment A benchmark run including a set of approaches and a set of maps

Trial A map spawned with robot and goal location and a set of approaches

Run Execution of an approach in a trial configuration

Chapter 1

Introduction

Whilst the field of robotics can arguably be dated back to the time of the ancient Egyptians using water clocks, where human figurines were used to strike the hour bells, the first autonomous mobile robots were built in the 1940s. This was when William Grey Walter built Elma and Elsie [1]. These two autonomous mobile robots moved toward the light and avoided obstacles along the way. This field of mobile robotics has developed significantly, having introduced various navigational approaches able to traverse unknown environments.

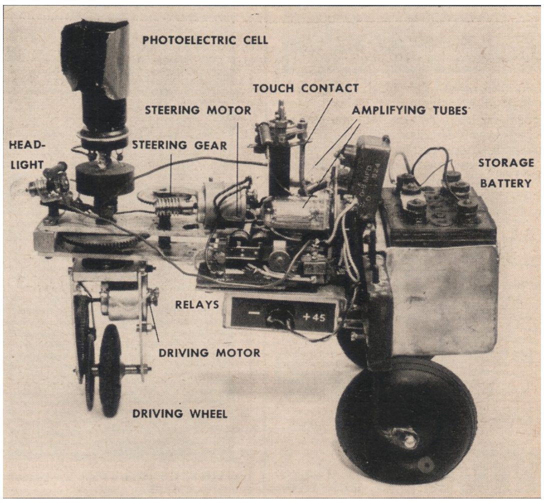

Figure 1.1: This figure shows the first autonomous robot. This light and touch sensitive robot uses these sensors to traverse environments whilst avoiding obstacles.

Current systems allow for extended real-world navigation through complex environments. Whilst these systems have advanced the field of robotics significantly; much work remains to be done in evaluating the approaches. The evaluation of approaches is a necessary step that allows for quantifiable measures of an approach's strengths and weaknesses. These measures aid in the further development of an approach and in determining the optimal application. A complication with evaluation frameworks is the extreme variety of different approaches. This is particularly evident in the goal-directed navigation of mobile robotics in unexplored environments. Current frameworks lack metrics and environments, and research contributions do not utilise the benchmarking frameworks to evaluate their approaches against a standard set of metrics and various maps.

It is against this background that this research aims to answer the following question:

Can we evaluate and quantify the performance of different navigational approaches in unknown environments with a common framework?

This research project developed an evaluation methodology that aims to answer this question. The methodology combines the relevant evaluation characteristics and metrics in path planning and extends it to navigational approaches. This work documents the information relevant to the development of this research and the results that demonstrate the achievement of the objectives. The first chapter introduces the motivation behind the development of this methodology, the objectives, and the structure of this work.

1.1 Motivation

Mobile robots exploring and navigating through unrevealed areas is a challenging research field covering many different areas. Motion planning (also sometimes called path planning), is a problem area where an autonomous vehicle finds the most optimal sequence of valid configurations to reach a goal state whilst avoiding obstacles. Researchers have proposed different approaches ranging from simple reactive strategies based on Brownian [2] or Braitenberg [3] vehicles to artificial learning-based strategies using Genetic Algorithms. The different approaches generate a variety of paths depending on their configuration on what they are optimised for, as seen in Figure 1.2. This figure shows four different paths; an invalid path *(as seen with the red path colliding with an object)*, a non-optimal path *(as seen with the purple wall following path)*, and two paths optimised for path length *(the blue path)* and path length with consideration to a certain object clearance *(the brown path)*.

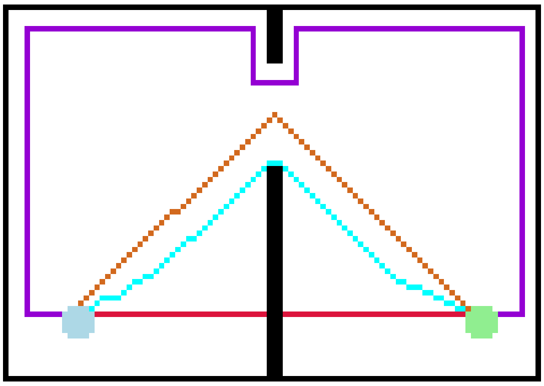

Figure 1.2: This figure shows several paths from the starting configuration marked with a grey square to the goal location marked with the green square. The red path is an invalid path as it collides with an obstacle. The purple wall follower, whilst valid does generate a very long sub-optimal path towards the goal location. The remaining two paths are more optimal solutions depending on the requirements. The blue generates the shortest path possible whilst the brown path sacrifices a bit of the path length for path smoothness and to keep a distance from the obstacles.

As more approaches are being proposed that allow mobile robots to manoeuvre increasingly complex environments autonomously, the importance of evaluating them grows. In the literature on mobile robotic path planners, it is common practice to compare the proposed novel approach against a limited number of algorithms, often only the approach it aims to improve on. These comparisons are performed predominantly in explored environments and highlight only the advancements made without presenting any of the limitations. Performance metrics and results differ depending on the state of exploration in the environment. This is evident from the findings produced by a recent comparison paper in explored environments [4] when compared to the findings in this research project. This motivated the current research project to evaluate different types of navigational approaches in a common framework. Furthermore, this evaluation is performed on maps that represent different problem environments to get a more optimal representation of performance.

Currently, there are multiple research projects aimed at benchmarking the different path planners in a common benchmarking framework [5] [6] [7]. Whilst some frameworks [8] present a mechanism for demonstrating navigation through unexplored environments [9], their capabilities could be improved in two key aspects:

- **Maps** vary significantly between frameworks and do not have a well-defined purpose [7]. As the maps are key to the evaluation efficiency of approaches, this research proposes that frameworks should have:

   1. a map that covers a specific key problem in a simplistic form e.g. a corridor
   2. a set of maps that cover this key problem in an increasingly more complex manner e.g. a tunnel that is a series of branching corridors
   3. maps that combine different key problems e.g. a dungeon with tunnels and rooms
   4. maps that represent close to or real-life scenarios

- **Evaluation methodologies** are limited to the export of metrics that focus on the success of an approach. However, achieving a goal alone is not the perfect metric because there are other desirable properties that might be of more value. With the ever-increasing efficiency of approaches, the importance of other factors increases, and this needs to be represented in any benchmarking framework. This, combined with the ability to quantify the success of an approach, will aid in the development of further approaches.

This motivated this research project to address the aforementioned limitations. The proposed methodology allows for such evaluation to be conducted within this framework or it could be added to any of the existing path planning frameworks to extend their capabilities in benchmarking navigational approaches.

1.2 Aim and contributions

The main aim of this work is to develop a benchmarking framework that addresses the limitation and the research question set forth in the previous section. Such a framework should cover the following objectives:

1. Define the metrics needed to evaluate and compare navigational approaches in unknown environments.

2. Define a set of maps to cover a range of common problem areas.

3. Define an evaluation methodology that evaluates the efficiency of different approaches and can be extended to different requirements, e.g. safety-critical systems.

4. Define a set of tools that evaluate a given approach on a set of maps under a given evaluation methodology

5. Define a single meaningful score value for any given navigation approach

This research has successfully achieved these objectives by contributing the following:

- Literature review: Chapter 2 evaluates the existing literature and examines the approaches, metrics, maps, and evaluation methodologies used. Whilst certain frameworks exist for robot exploration this research contributes the first navigation framework that generates a benchmarking score from several key metrics.
- Evaluation methodology: Section 3.5 presents an evaluation methodology that can evaluate different approaches across a set of metrics and approaches in unknown environments. This framework is used in Chapter 4 to evaluate the performance of the Potential Field approach. The results led to the development of the novel Pheromone Potential Field approach. Furthermore, Chapter 5 evaluated the performance of commonly used approaches and generate a single meaningful score.
- Maps: A set of maps representing common problem areas in navigational robotics are discussed in Chapter 3 and documented in the Appendix 6.3.

1.3 Selection of references

The sources for this research project are divided into two key aspects; the path planning approaches and their benchmarking frameworks. Sources on path planning were included when:

- the source was the original novel contribution, e.g. Dijkstra's algorithm [10]
- the source was an improvement on the novel contribution e.g. A* approach [11]
- the source detailed the measurement in which the improvement was determined
- the source critically analysed an approach and reflected on the limitations of an approach e.g. the limitations of potential fields [12]

When evaluating the sources for different contributions made to benchmarking the path planning approaches, the following criteria were observed:

- the methodology was not limited to similar types of approaches i.e. limited to only learning based-approaches or restricted to specific representations of environments or data structures.
- the framework was recent and ideally still maintained

The main aim of these sources is to reflect on the development of approaches in path planning and the metrics used in individual contributions and benchmarking frameworks to determine this improvement. IEEExplore and Google Scholar are the initial starting point for sources. If the source did not meet the criteria above, the evaluation of both the references and citations were used as points to explore further sources that provided the information required. It is preferred in the case of the benchmarking sources that they are recent and actively contribute to evaluating the state-of-the-art and validating the originality of the contribution of this research.

1.4 Structure of this work

This chapter introduced the research project with its aims and objectives. The remaining chapters are organised as follows:

- **Chapter 2 - Overview of navigational approaches:** addresses the first object by presenting the literature review on path planning approaches and evaluation frameworks. The first section covers the most commonly used global and local path planners. These planners are presented with a specific consideration as to how they have advanced and the methodologies used to determine these advancements. The second section initially aims to evaluate what would constitute an effective evaluation methodology. This is followed by a review of the different evaluation methodologies and frameworks that currently exist.

- **Chapter 3 - Benchmarking framework development:** addresses the remaining objectives by presenting the framework developed in this research to evaluate the different navigational approaches. After discussing the framework's necessary characteristics, this chapter presents the different problem environments considered and the maps that embody them. This is followed by a documentation of the representative approaches and their implementations. Finally, this chapter concludes by detailing the metrics used, the experimental setup, and the evaluation methodology.

- **Chapter 4 - Evaluating Potential Fields :** presents the results of evaluating the performance of the potential field method. The second section presents the novel Pheromone Potential Field approach. These approaches are detailed in implementation and evaluated in their performance in accordance with the relevant parameter ranges, the different problem environments, and metric scores. This chapter compares the two approaches demonstrating the capabilities of the framework in performance analysis and the improvement of the Pheromone Potential Field over the traditional Potential Field methods.

- **Chapter 5 - Benchmarking navigational approaches:** compares the performance of several representative path planning methods using the framework. This chapter aims to benchmark these approaches quantitatively, using the developed framework. The results of these benchmarks are presented and discussed here. The experimental results are also assessed to determine the optimal approach for each set of problem environments.
- **Chapter 6 - Conclusion and future work :** concludes the research findings by discussing the outcomes of this project. This chapter also presents the possible future extensions for this work with a specific focus on the dynamic switch algorithms that would utilise the findings of Chapter 5 to navigate an environment possibly more efficiently than a single approach could.
- **Appendix - Environments :** This appendix includes the documentation of all the static and dynamic problem environments used in this research with a discussion on the possible extensions for these maps to cover more known navigational problems.

1.5 Scientific output

This thesis resulted in the following resources:

- **Framework implementation:** Source code for this framework is publicly available at `https://github.com/MoIdriez/Benchmarking`.
- **Initial work:** The initial work that led to the proposed framework was presented at TAROS, 2019 [13].
- **Generated maps:** are also publicly released along with the framework implementation.

Chapter 2

Overview of navigational approaches

This chapter examines the literature on mobile robotic navigation, and the evaluation of such approaches necessary for this research's development. It addresses the first objective of this research: the investigation and definition of the metrics needed to evaluate and compare the performance of different navigational approaches in unknown environments.

Due to the nature of this project, it is important to progress through several main background aspects, all of which are explored in this research. First, an introduction and overview of the research in mobile robotics path navigation is given, demonstrating the different types of approaches in navigation and the advancements made throughout the past couple of decades. This is followed by a section reviewing the mechanism used to evaluate the success of an approach and the advancements made compared to other methods. This is accomplished by closely examining the metrics used within the different evaluation approaches to determine the success and advancement of a navigational approach. This is followed by a discussion of the methodologies of comparisons used in the literature alongside a review of the current evaluation and benchmarking frameworks that exist. This chapter is concluded with a section discussing the outcomes of the literature review.

The aim of conducting this review is to identify:

- the main approaches used in mobile robotics
- the key metrics that provide information on the approach and its effectiveness
- the modelling methodologies used to evaluate the approaches

The resulting approaches, metrics, and methodologies are incorporated to develop the benchmarking framework.

2.1 An introduction to navigational approaches

Mobile robotic navigation is one of the fundamental fields in robotics. The research aims to enable a mobile robot to navigate safely from a start position to a goal position. Navigation itself involves an interaction of multiple systems drawing from separate, extensive research fields in perception, map building, localization, motion control, and path planning [14]. Over the past several decades, approaches addressing this problem have evolved from simple obstacle avoidance systems to complex systems allowing for multi-robot exploration [15], swarms [16], human aware navigation [17] and search and rescue [18] to name a few.

More specifically, unexplored path navigation involves the essential step of allowing a mobile robot to navigate a previously unexplored environment from a start state to a goal state whilst avoiding obstacles. Efficient approaches save time and energy usage and aid in more robust mobile robotic systems [19].

Mobile robotic path planning and navigation strategies are classified by the availability of prior knowledge of the environment [19, 20]. These classifications are global path planning and local path planning. For global path planning approaches, the mobile robot requires prior knowledge of the obstacle and the goal positions, whereas, in local path planning, this information is not required. In navigation, the goal is known. However, the environment and its obstacles may be unknown. Therefore this work divides the approaches depending on the importance of an internal representation of the environment. Each division discusses the most prominent approaches, how they have advanced, and the mechanisms used to determine this advancement.

2.1.1 Global navigational approaches

In global navigational approaches, there is a significant reliance on the overall environment model. These approaches frequently adapt and recalculate routes with the availability of new information. In this section, we will be looking at the advancements made in the most popular approaches within this category.

Dijkstra's algorithm is arguably the most popular navigational path planner originally conceived by Dijkstra [10]. This graph traversal algorithm finds the shortest path from any starting configuration to any node by creating a tree from a starting node to all other points in the graph. It utilizes the weighted edges, often distance, to keep track of the short distance from any node to the starting point. The most common improvement is the A* algorithm [11] which uses a goal-directed heuristic that allows for finding an optimally efficient path [21]. Other improvements include incremental search [22], bidirectional search [23], and the utilization of the Delaunay triangulation for surface modeling [17]. Comparison attempts by Permanaet al. [24] and Zarembo [25] show the overall efficiency of the Dijkstra-like approaches in terms of path length, computational time, and memory usage. Due to their proven effectiveness as a path planning approach in deterministically finding the shortest path in explored maps, the A* approach is used as a baseline in this research.

Genetic algorithms are artificial-intelligence-based approaches inspired by natural selection. Whilst the computer simulations in evaluation can be dated back to the works of Barricelli [26], the genetic algorithm was firstly proposed by Holland [27]. In a genetic algorithm, all the possible solutions are encoded as chromosomes and referred to as the initial population. Determined by a fitness function, members of the initial population compete in a joint process involving crossover, mutation, and selection. Liu et al. [28] introduced a genetic-algorithmic approach to dynamic path planning. Their approach was evaluated mostly on convergence speed and computational time. Liu identified in their contribution the importance of comparing to more approaches, in more complex environments, and with more metrics. A better evaluation was done by Qu et al. [29], which proposed an improvement with the co-evolutionary strategy. This improvement was determined by measuring the shortest path length, computational time, and speed of convergence to an optimal path on 3 different known and static map structures. Recently Bacchin et al. [30], developed a human aware navigational approach using genetic algorithms.

Rapidly exploring random trees is a randomized path planning algorithm proposed by LaValle [31] that takes a random free sample and connects this sample to the nearest point on the tree within a predetermined range. The reduction in the number of connections leads to better performance of the PRM. These methods have been improved by adding progressive search [32], the introduction of goal-directed heuristics [33], combined with frontier selection [34], and bi-directional tree growth [35] among others. Comparisons between the many RRT approaches are limited to evaluations with one or two other similar approaches. Noreen et al. [36] noted in their survey of the different approaches that a performance analysis in different problems would be beneficial.

A recent contribution compared the results of A*, GLS, RRT and PRM in a simulated environment [4]. The evaluations were done on explored maps measuring computational time and path distance. The results from this evaluation show an increased processing time for the A* when compared to the other algorithm. However, results from the current research project, presented both in Section 5.3 and Appendix 6.3 show that when exploring unknown environments, this processing time is similar between approaches. This indicates the importance of evaluating approaches on explored and unexplored maps. Most research efforts have been on explored maps which motivated this research to further contribute to evaluating the performance of approaches on unexplored maps.

2.1.2 Local path planning

Probabilistic road map is a local approach, initially introduced by Kavraki et al. [37], that takes random free samples and connects this sample to the nearest k neighbours in a predetermined range. This generates an undirected road map graph which can be utilized by any graph search approach to determine the shortest path. The developments in the basic PRM approach come in the form of efforts on how to best sample the space [38] [39], what local planners to use, when to connect the nodes [40], and what pre and post-processing perform well [41] [42]. The first step toward creating a comparative framework was developed by Geraerts and Overmars [43]. In their contribution, several PRM approaches were evaluated on their collision checking, sampling and node addition. All performances were measured on a single metric of time. This research project expands on their choice of experimental environments (see Section 3.3).

The Artificial Potential Field initially proposed by Khatib [44] is a popular path planning approach. The robot is affected by two types of forces; the attractive gravitational force originating from the target that pulls the robot towards it and a repulsive force pushing it away from other objects. Whilst this approach allows for the robot to navigate from a start state to the goal state whilst avoiding obstacles, it does have some well-known shortcomings [12] with the biggest issue being local optimal solutions which have also been the subject of focus for most further research. Initial approaches [45] [44] were often reactive and had shortcomings in clearing closely situated obstacles which affected the success rate. Furthermore, as the robot moved, drastic changes in forces could be calculated that would result in unstable and non-smooth paths. To remedy these issues and the inherent issue of local minima, further research has focused on the detection of, and escape from, local minima. Focusing primarily on this issue, Ge and Cui [46] proposed that for dynamic systems, the continuous movement of the robot might help to overcome local minima. Simple approaches like random movement and wall following can be used to accomplish this. Vadakkepat [47] proposed a new approach named Evolutionary Artificial Potential Field that utilizes genetic algorithms to fine-tune the potential field functions and introduces a quantity named escape force to escape local minima resulting in a higher success rate. A swarm intelligence-based approach using similar potential fields was proposed by Dorigo et al. [48]. This approach, inspired by the foraging behavior of ants, has been used with several heuristic methods [20] [49] [50] to explore and path plan environments.

2.2 Comparing evaluation frameworks

The advancements in any scientific field are built on the competition between different research groups and the exchange and evaluation of results and ideas [51]. In the field of mobile robotic path planning and navigation, this would require the benchmarking of different approaches under an identical evaluation methodology on a well-defined set of environments. This section looks at the evaluation methodologies and frameworks that benchmark approaches.

Michel et al. [52], claim that RoboCup [53] should be recognized as a reference benchmarking in robotics. RoboCup, Robotics Soccer World Cup, is an annual robotic competition where robots compete in a game of football. I find myself aligned with the works of Balaguer et al. [54] in support of Baltes et al. [55] that robotic competitions do not suffice in providing a generic framework for performance evaluation due to their inherent nature and scope. This also puts a spotlight on a general trend of literature in mobile robotics benchmarking and performance evaluation, often being specific to certain algorithm types and/or addressing specific issues.

Baltes et al. [55], whilst critical of the effectiveness of mobile robotic competitions, do suggest several numerical metrics that can be seen as a step toward quantifiable results. Caltes et al. [56] provides a common framework adding the notion of time which in turn was extended by the work of Basilico and Amigoni [57] adding the power consumption.

A key problem with finding an optimal navigational solution for distinct environments is the ability to evaluate the different approaches with their vastly different features. The selected metrics must not be biased and apply to a wide range of approaches in distinctly different environments.

Amato et al. [58] investigated the effect of different distance metrics on probabilistic roadmap methods. Their research showed that a good choice is the Scaled Euclidean distance metric with some deviations depending on the environment. Gao et al. [59] however, argue that path planners primarily focused on the shortest distance generate inadequate paths that do not consider smoothness or the distance of the path from obstacles.

Morales et al. [60] proposed that to allow for adaptive planners, metrics should not only be calculated on a global level but also on a node and region level. This allows for the introduction of more complex metrics, among others, visibility and environment estimation, that are key for adaptive planners.

Currently, there are attempts of benchmarking frameworks for socially-aware robots [61] [62], dynamic environments [63] [64], and multi-robot systems [65] [66] among others. There most notable motion planning libraries, with some benchmarking capabilities, are ROS [8], OMPL [67], and MoveIt [5]. There are also several attempts made with OpenRAVE [6], PathBench [7], and OOPS [68] that are noteworthy but haven't had recent contributions made to them. The contributions are discussed here.

The Robot Operating System (ROS) [8] is the standard in robotics. It provides different motion planning algorithms and environments for a variety of different types of robots. It acts as a base for many benchmarking frameworks. Moll et al. [69] is one of the frameworks extending on ROS with the Planner Arena. This benchmarking framework proposes to standardise many aspects of benchmarking, including the extension of the OMPL and the generalisation of the input and output file formats.

The Open Robotics and Animation Virtual Environments, (OpenRave) [6] is a high-level scripting framework that focuses on real-world robotics, including simulations, visualizations, planning, and control.

The Online Open-Source Programming System for Motion Planning (OOPS) [68] is an online framework for comparing different types of motion planning approaches on a common set of maps. It provides implementations of common approaches with some benchmarking capabilities.

PathBench [7] extends motion planning benchmarking with native support for machine learning-based approaches. This framework utilizes maps developed by Sturteven et al. [70].

The Open Motion Planning Library (OMPL) [67] is a motion planning library that also has benchmarking capabilities. Whilst commonly used, it is limited to sampling-based approaches only. A similar type of library is the

MPK [71], but this library is not discussed as it does not extend beyond the capabilities of OMPL, and it is not actively used or maintained.

MoveIt [**5**] is a framework that combines both ROS and OMPL that can be extended easily for new motion planning approaches. Cohen et al. [72] introduced a benchmarking framework as part of the MoveIt [5] project. In this work, they benchmark motion planning algorithms in robotic arm manipulation. Evaluations were done according to the following metrics:

- *Computational time:* several times were measured including path planning, simplifications, and post-processing
- *Path length:* the length of the path as total distance
- *Smoothness of path:* where α is the angle between two consecutive segments of the path containing n number of segments. The smoothness is calculated as $k = \frac{1}{n}\sum_{i=2}^{n}\alpha_i^2$
- *Clearance:* average minimum distance the path took from obstacles
- *Success rate:* the percentage of successful attempts a motion planning approach has within a specified time frame

Another noteworthy contribution is the effort by Sturtevant et al. [70] to create a general set of maps to be used. Whilst this includes an extensive set of maps there is no clear correlation to a navigational characteristic/problem. Therefore in this research, we have opted to use simplified abstractions of some of those maps that focus on specific problems. This framework could however be easily extended to include the entire general set of maps.

2.3 Closing statement

This chapter presented the relevant research done in the progression of navigational approaches and the methods to evaluate them. This review concludes that whilst several frameworks exist that evaluate the performance of mobile robotic path planning there are numerous limitations to them. These limitations can be summarised as follow:

- **maps:** there is no standard to the maps used across the frameworks. The choice of maps to be evaluated needs to be justified and represent specific problems in navigational robotics.

- **metrics:** the frameworks do not produce a single overall measure to determine the efficiency of an approach under a specific condition.

- **approaches:** the frameworks can be limited to the evaluation of a specific approach type.

This research project aims to combine the various evaluation methodologies and metrics scattered across the different frameworks within a single framework to address these limitations. This framework is also complemented with a set of maps that addresses common navigational problems.

Chapter 3

Benchmarking navigational approaches

This chapter presents the proposed framework and evaluation methodology to benchmark the performance of navigational mobile robotic approaches. In developing this framework, the limitations identified in the literature review conducted in the previous chapter have been considered. The efforts in this evaluation framework are divided into two; the first part simulates the execution of a navigational approach on a mobile robot in a selected problem environment, and the second part evaluates the results of the first part and accordingly benchmarks the performance of the navigational approach.

Section 3.1 presents the characteristics of this framework that are necessary to address the limitations identified in the literature review. This is followed by section 3.2 presenting an overview of the simulator. Section 3.3 presents the different problem environments the simulator utilises to run the navigational approach. The environments are presented in this chapter and documented in detail in Appendix 6.3. Section 3.4 documents the implementation of the different navigational approaches used in this research project. Finally, section 3.5 discusses the evaluation of an approach by detailing the metrics and the evaluation methodology. These sections address the remaining objectives of this research.

3.1 Characteristics of the proposed framework

The limitations identified in the literature review motivated the development of an evaluation framework. This framework needs specific characteristics to provide more insight into how certain navigational approaches perform on particular challenges compared to other techniques. These characteristics are as follows:

- **The framework offers an unbiased and large number of environments.** This characteristic was achieved by using many static maps alongside procedurally generated maps. The maps address specific issues in navigational approaches or problem environments, not the navigational approaches themselves.

- **The framework can evaluate different types of navigational approaches in identical conditions.** This is a fundamental characteristic of the framework, which was also taken into consideration with the robot development. This was achieved by allowing the robot to execute a navigational approach's plan, whether that is a list of nodes in a graph for more complex path planning approaches, or a single location, as is the case in more reactive approaches. The separation created by this between the mobile robot and the approach also ensures that the robot is not biased towards a certain approach.

- **Metrics gathered are useful on a single run, per navigational approach evaluation, and in comparison with other navigational approaches.** This research aims to aid in the general advancement of the field. It is believed, therefore, that metrics must be meaningful per individual run level for algorithmic development, across all the different environments for validation, and in comparison with other navigational approaches to compare overall efficiency.

3.2 Simulator overview

The simulator is a library that fundamentally takes a map and a navigational approach as input and performs the simulated movement of a mobile robot in the given environment. For these simulations, we use an omnidirectional robot that can step from each point in the grid to the neighbouring grid points. This robot is presumed to have object recognition and classification which enables it to detect walls and the goal once it is in its sensory range. As the robot navigates through the environment towards the goal, it creates an internal map that the navigational approaches can utilise to make informed decisions. It also holds all the capabilities for measuring time, the usage of resources, and a mechanism to verify whether it is stuck at local minimum. This mechanism works independently from any method implemented with an external tool, that may be appropriate for a given approach. The simulator has two main components; map generation and approach execution.

The map generator is a module that generates a range of 2D maps. These static and procedural maps cover a range of different problem environments from simple 50 by 50 arbitrary unit obstructing walls to a 200 by 200 unit procedurally generated network of tunnels for the robot to navigate. This original map is never made available to the navigational approach but is used to determine the complexity of the map. The robot is aware of its starting location and the goal location. Whilst the mobile robot navigates this unknown environment, it generates an internal map of its surroundings allowing the navigational approach to make more informed decisions. This module is covered in more detail in section 3.3.

The approach executive is the module responsible for benchmarking the approaches. The simulator executes an experiment given a set of maps and approaches. An experiment consists of trials. At each trial, the framework starts by obtaining an environment, generating the random or pseudo-random spawns of the robot and goal location, and calculating the baseline for this trial. The baseline is determined by running an A* approach on the trial's fully explored environment. This step also determines the validity of the setup by ensuring connectivity (which is especially important in the case of generated maps) and provides the necessary information for calculating the path length and time metrics. The next step is executing all sets of approaches under the trial configuration but with an unknown environment. An experiment can instruct the framework to execute an environment several times. In this case, for each time, a new trial is generated with the new robot and goal locations. Figure 3.1 shows a visualisation of this process.

A run is an execution of an approach under a specific trial configuration. In every run, the robot's navigation toward the goal location is a step-wise iterative mechanism. The robot, at every step, requests the next step from the navigation approach. This next step is calculated at each iteration with reactive approaches, e.g. the Potential Field approach. However, other approaches generate a path consisting of a list of steps the robot follows, e.g. the RRT approach. As the robot navigates through the environment, it builds an in-

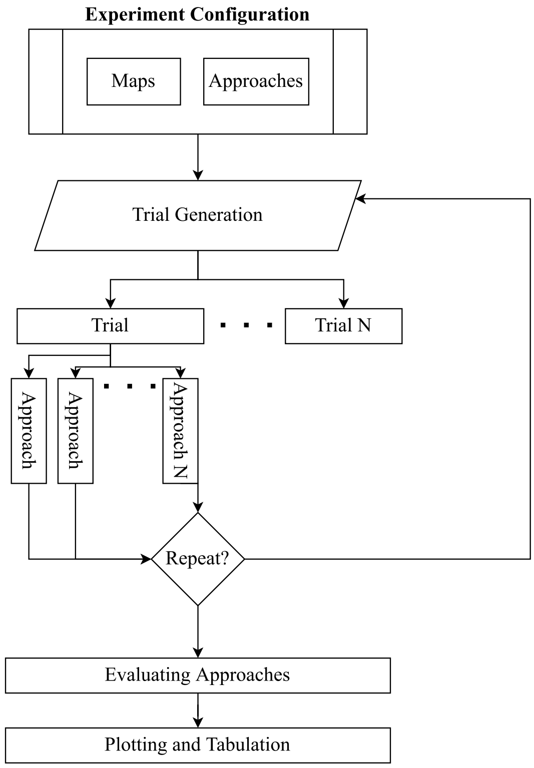

Figure 3.1: This figure visualises the steps the framework goes through when executing an experiment.

ternal map of the explored areas. The navigational approaches can utilise this exploration map to make informed decisions and alter the paths, e.g. to avoid obstacles and improve path quality. This navigation toward the goal location continues until it reaches the goal, gets stuck, hits max iteration, or hits max time.

3.3 Problem environments

This section presents the environments used by the framework for evaluating the different navigational approaches. The environments represent common problems in navigational mobile robotics. The static maps evaluate the approach's performance on object avoidance, path obstruction, narrow corridors, local minima, and a combination of the aforementioned. The dynamic ones increase the complexity of a specific problem or combine several different types of challenges. Furthermore, these dynamic maps provide non-deterministic maps for evaluation. An environment is a two-dimensional grid map with an outer layer wall. Within each environment, we randomly or pseudo-randomly spawn a mobile robot and a goal. Within the map, we spawn a set of predetermined or procedurally generated objects that the robot must avoid whilst navigating towards the goal location. The environments are divided into static and procedurally generated.

3.3.1 Static Environments

The first type of environment is the traditional static environment. These environments aim to present a common navigational challenge in an isolated environment. Unless expressly specified otherwise, maps are a grid of 50 x 50 arbitrary units with an outer border of an artificial wall. The robot and the goal location occupy one grid location at any time. Each type of environment is split into three levels of complexity, creating a total of 18 maps that are evaluated. The figures in this sub-section show the static environments, the possible robot spawn locations (marked in blue), and the possible goal spawn locations (marked in green).

Wall Environment In these maps, the robot and goal are spawned on opposite sides of a wall. The maps range from a wall blocking only 20% of the width of the map to a wall blocking 70% of the available space. These relatively simple maps provide insight into essential object avoidance. Furthermore, the higher complexity levels generate local minima problems for particular approaches.

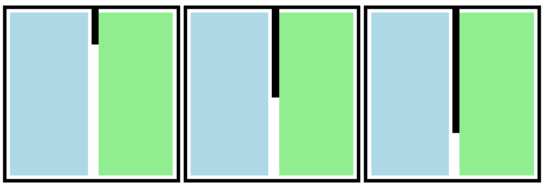

Figure 3.2: This figure shows the three Wall maps. These maps increase in complexity from left to right. The robot and goal spawn randomly in the blue and green rectangle, respectively.

Plank Pile Environment These maps feature increased complexity compared to the previous environments. Rows of small walls separate the robot and goal spawn regions. With every level of complexity, the map's size and the number of walls increase. The map's sizes are; 50 x 50 units, 100 x 100 units, and 200 x 200 units. These maps aim to evaluate the performance of the approaches with a large number of objects in an increasing size environment.

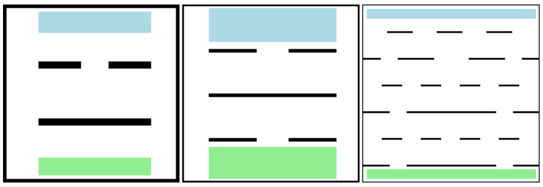

Figure 3.3: This figure shows the three Plank Pile maps. These maps increase in complexity from left to right. The robot and goal spawn randomly in the blue and green rectangle, respectively.

Corridor Environment To explore further the challenge of navigating through narrow environments, these environments split the robot and goal spawn areas with a narrow corridor of an extended length. Initially, the corridor is a straight corridor connecting the two areas. The complexity of the corridors grows with each map's complexity level. The first two complexity levels extend some of the challenges of the previous environments in a different form. The final complexity level explores in an obvious way how approaches deal with local minima problems as the method would need to move away from the goal for a while to be able to reach it.

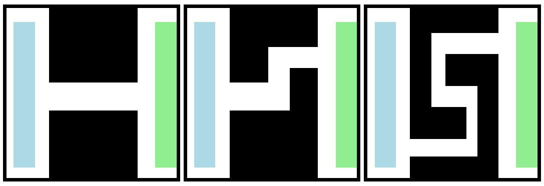

Figure 3.4: This figure shows the three Corridor maps. These maps increase in complexity from left to right. The robot and goal spawn randomly in the blue and green rectangle, respectively.

Bug Trap Environment In these classic map types, we further explore the effects of local minima problems. In these maps, the robot spawns inside a U-shaped 3-wall obstacle. The goal spawns on the opposite side of the middle wall. The robot would need to first navigate out of this "trap" before reaching the goal. The complexity of this task is made more difficult by narrowing the open side of this obstacle.

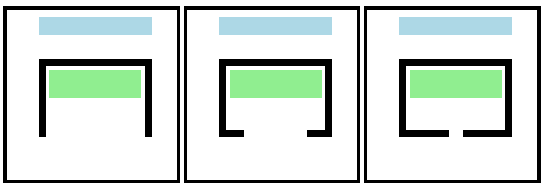

Figure 3.5: This figure shows the three Bug Trap maps. These maps increase in complexity from left to right. The robot and goal spawn randomly in the blue and green rectangle, respectively.

Room Environment These maps aim at providing abstractions of human-like environments. The complexity grows from straightforward room-like environments to larger maps more closely resembling maze-like environments to combine all the characteristics of previous environments. The map's sizes are; 50 x 50 units, 100 x 100 units, and 200 x 200 units.

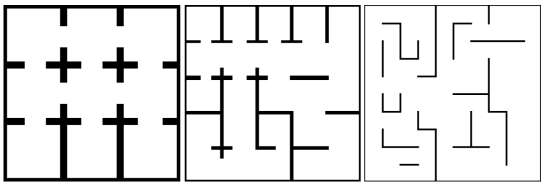

Figure 3.6: This figure shows the three Room maps. These maps increase in complexity from left to right. The robot and goal spawn randomly anywhere on the map with at least 150 units in between them.

3.3.2 Generated Environments

In this novel map generation method the obstacle and tunnel environments are procedurally generated. These maps are pseudo-randomly generated maps so as not to allow for predictable path planning. These maps are larger in size, being 200 x 200 units. The robot and goal spawn randomly across the map with at least 150 steps in between them. The step count and connectivity are ensured with the baseline step of the trial configuration. If the baseline step can not connect the goal and robot location on the fully explored map, the map is re-generated.

Obstacle Environments For these maps, we generate a random amount of obstacles of different sizes within a specified range. The obstacles are permitted to stack on top of each other to create interesting environments with odd shapes. The map's complexity starts from a sparse environment to maps cluttered with obstacles. The main aim of these environments is to evaluate the performance of an approach as environments get progressively restrictive. Table 3.1 shows the quantity of 3x3 - 5x5 units obstacle used for setting up the 5 different obstacle maps. The effect this has on the average free cells and average path length is measured by generating 100 valid maps. Figure 3.7 shows five sample environments with the different complexity levels.

Obstacle Count	Average Free Cells	Average Path Length
70 - 75	86.90 %	163.40
190 - 195	69.95 %	162.00
330 - 335	51.79 %	185.10
450 - 455	38.23 %	198.80
510 - 515	32.59 %	217.10

Table 3.1: This table details the parameters for the 5 obstacle maps and the average proportion of free cells. The average path length of the baseline approach also increases with every complexity level

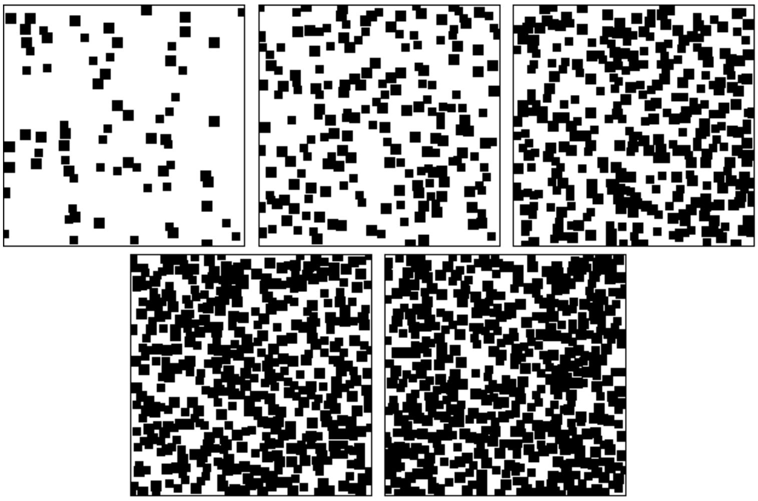

Figure 3.7: This figure shows five Obstacle maps. Starting from the left, these maps increase in complexity. The robot and goal spawn randomly anywhere on the map with at least 150 units in between them

Tunnel Environments These maps are generated by range-controlled random room and corridor sizes. Initially, a room is placed that is followed by a corridor in any of the cardinal directions. We connect a room to that corridor and repeat the process till we reach the level of complexity needed for the map. Rooms and corridors are permitted to overlap to create non-deterministic shapes and maps. Table 3.2 show the parameters used to generate the five tunnel maps. Note the increased path length with the decreased number of rooms and corridors. Figure 3.8 shows five sample tunnel environments where the complexity increases with a reduced number of pathways.

Rooms Count	Average Free Cells	Average Path Length
110 - 155	42.79 %	164.00
81 - 85	31.31 %	175.80
61 - 65	26.57 %	191.90
41 - 45	19.85 %	191.50
21 - 25	11.14 %	207.40

Table 3.2: This table details the parameters for the 5 tunnel maps. The effect this configuration has on the average path length is also demonstrated.

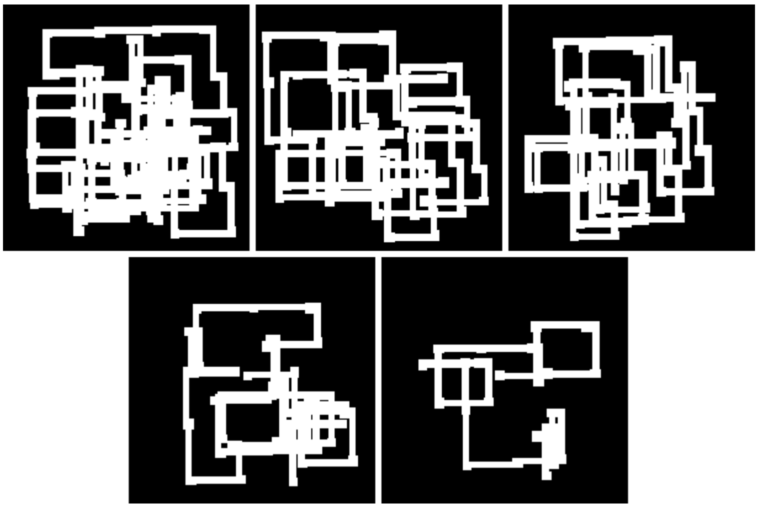

Figure 3.8: This figure shows the five Tunnel maps. Starting from the left, these maps increase complexity due to fewer connections between tunnels. The robot and goal spawn randomly anywhere on the map with at least 150 units in between them

3.4 Navigational approaches

This section documents the navigational approaches used in the experiments and details their implementation. To fulfill the characteristics and requirements needed for the framework, these approaches are significantly different approaches, ranging from purely reactive approaches to graph traversal-based approaches. There are also examples of base approaches and improvements on these approaches that allow this framework to quantify the degree of improvement, if any. This framework could also be used to evaluate a trained or online machine-learning mobile robotic approach. Future work will include the ability to train on this framework and allow for mobile robotic systems.

3.4.1 A* approach

This extension of Dijkstra's algorithm [10] [11] uses heuristics to guide the search for the shortest path. This approach generates a path from the starting point to the goal location using the map information it has available. This plan is followed by the robot till it reaches a point where the plan cannot be followed due to the existence of an object. The navigational approach uses the updated map of the environment obtained from the navigation up until that point, to develop a new plan from that location to the goal location. This incremental approach is followed until the robot reaches the goal or reaches any of the end conditions specified. These end conditions are: being stuck, reaching the maximum step count, or reaching the maximum time allowed.

The navigational approach for generating a path from the robot's current position to the goal location is as follows:

Algorithm 3.1 A* algorithm

This algorithm maintains two lists; an *open* and a *visited* list.
The *open* list contains all the nodes that are visited and need expanding.
The *visited* list contains all the nodes that are visited and expanded.

```
open ← robot location
while open has items do
    current ← lowest value node from open
    successors ← neighbouring cells to current
    for each successor in successors
        if successor == goal; break
        if successor is in visited with lower cost; continue
        if successor is in open
            if successor cost is lower than in open:
                update open cost with successor cost
        else
            add successor to open
    add current to visited
if current != goal:
    exception                         // start and goal locations not connected
```

Note:
The lowest value node = cost from the start (distance between steps) + cost till the end (euclidean distance to goal).

3.4.2 Potential field approaches

This navigational approach [44] generates repulsive and attractive force from obstacles and the goal location, respectively. These forces guide the robot from the start state to the goal state whilst avoiding obstacles. For this framework, the base Potential Field approach was implemented. Furthermore, the improved Pheromone Potential Field has been proposed and implemented. These two approaches have been studied and compared more in-depth in Chapter 4.

Potential Field The basic potential field navigational approach generates its next step by considering all the obstacles within a specific range and the attractive force from the goal at any range. The default algorithm is as follows:

Algorithm 3.2 Potential Field algorithm

This approach calculates the next step based on the current location and surroundings.

```
while current != goal do
    attractive ← dir * constant * (1 / distance)                      // Note 1
    repulsive ← 0
    for each obstacle point in max range
        repulsive add dir * constant * (max range / distance)         // Note 2
    current ← current + attractive + repulsive                         // Note 3
```

Note 1: The direction is from the robot to the goal.
Note 2: The direction is from the obstacle to the robot.
Note 3: 1 unit step in any direction.

Pheromone Potential Field This extension of the basic potential field algorithm [44] drops pheromones as it navigates through an environment. This approach, inspired by the way ants move in nature, instructs the robot to drop artificial pheromones at locations he has visited. However, contrary to the attractive nature of pheromones for ants, these pheromones act as a source of a repulsive force for the robot itself that guides the robot away from local minima and locations it has visited. The more frequently a robot visits a location, the more prominent the pheromones and thus the repulsive force. The navigational algorithm looks as follows:

Algorithm 3.3 Pheromone Potential Field algorithm

This algorithm maintains a **pheromones** list with all nodes visited and the strength of these pheromones

```
while robot location != goal do
    att ← dir * constant * (1 / distance)
    rep ← 0
    for each obstacle point in obstacle range
        rep += dir * constant * (obstacle range / distance)
    pher ← 0
    for each pheromone point in pheromone range
        pher += dir * constant * strength * (pheromone range / distance)
    next step ← current + attr + rep + pher
    if next step == obstacle point
        next step = current
        pheromones add next step or increase strength if exists
    else
        pheromones add current or increase strength if exists
    robot location = next step
```

3.4.3 Rapidly exploring random tree approaches

This navigational approach [31] grows a tree from the robot's location by gathering random samples and connecting them to the nearest point on the tree where possible. Once the tree reaches the goal, the shortest path along the nodes of the tree from the starting point to the goal node is the path. This work has been extended into RRT* [73] where each node has the travel cost needed to get to this point from the start. When a new node is added, the lowest cost node, as opposed to the shortest distance node, is selected. Finally, the RRT* also checks the nodes neighbouring the new node to check whether any re-wiring of those nodes would decrease the cost of that node.

Algorithm 3.4 RRT algorithm

```
current node ← (robot location, null)                                   // Note 1
nodes ← current node
while current node distance to goal ¿ goal range do                     // Note 2
    random location ← get random location within growth range of any node
in nodes                                                                // Note 2
    current node ← (random location, nearest node)
    nodes += current node
goal node = (goal location, current node)
path ← from goal node get all parent locations till null                // Note 3
```

Note 1: A Node is a combination of a location and link to the parent node
Note 2: Straight line between points must be unobstructed
Note 3: This is a recursive method that returns a parent location and goes up a level

For the RRT* we can replace Line 6 with the re-wiring mechanism

Algorithm 3.5 RRT* Rewiring Nodes

```
1: neighbours ← all nodes within growth range from random location
2: current node ←(random location, lowest cost neighbours node)
3: nodes += current node
4: for each node in neighbours
5:     if current node cost < node parent cost
6:         node ← (node location, current node)
```

3.5 Evaluation methodology

This section presents the second part of the framework, the evaluation methodology. This model provides insight into the navigational approaches by benchmarking the performance of the each approach in accordance with the metrics generated. Metrics are measured for each run and aggregated over the whole experiment. The evaluation methodology tracks the success, path length, path smoothness, time, and visibility metrics. These are an aggregation of non-approach-specific metrics used in the most notable existing evaluation frameworks [8] [5] [67]. This section introduces each metric and how they are measured. All metrics score a value between 0 and 1.

Success: this metric measures whether the robot successfully navigated from the start location to the goal location within the specified restrictions like obstacle avoidance and below the specified maximum time and iterations. The success rate s_r is the average success rate across all map types, where s_i is the average success rate on map type i:

$$s_r = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{3.1}$$

Path Length: this metric measures the path length efficiency. The path length for any given run is the accumulated euclidean distance between every two steps along the path travelled by the mobile robot. This length is evaluated in a range from an upper boundary r_{mx} to a lower boundary r_{mn} that is generated based on the baseline's path length in the same trial. This metric is configured to rate the baseline approach at an 0.9 score, allowing more efficient methods to score higher.

$$r_{mx} = b_p k \tag{3.2}$$

$$r_{mn} = b_p - \frac{r_{mx} - b_p}{9} \tag{3.3}$$

$$p_l = \frac{1}{n} \sum_{i=1}^{n} \frac{p_i - r_{mx}}{r_{mn} - r_{mx}} \tag{3.4}$$

Where:

b_p is the total steps the baseline performed on this configuration

k is a constant used to calculate the max range

p_i is the total steps the approach performed on configuration i

p_l is the path length score

Note: *that the upper boundary r_{mx} generates a score of 0 and the lower boundary r_{mn} generates a score of 1 allowing efficient approaches to score higher.*

Path Smoothness: this metric measures the cumulative angle between each step segment. As drastic changes in directions are sub-optimal in robotic movement, this measure of rotational movements provides insight into the smoothness of the path generated by the navigational approach. This metric has a direct effect on the power consumption of a mobile robot. The metric is calculated as follows:

$$p_s = \frac{1}{n} \sum_{i=2}^{n} \alpha_i^2 \tag{3.5}$$

where

n is the total steps the baseline performed on this configuration

α_i is the angle between segment $i - 1$ and i

p_s is the path length score

Time: this metric measures the execution time of the approach, including the planning time. The time metric has proven to be a limiting factor for approaches with extended planning periods. Chapter 5 discusses this metric's effect in more detail.

Visibility: this metric measures the level of obstruction to the field of view the robot experiences as it navigates through the environment. This clearance measure provides insight into the environment and the path choices of the navigational approach, i.e. moving close to walls. This measure is vital for safety-critical systems.

Algorithm 3.6 Visibility algorithm

This method calculates the visibility around the robot at any given moment in time where
start is the robot's location
length is the length of the field of view

```
fov ← default
for each angle in field of vision
    end ← (x : start_x + length ∗ cos(angle), y : start_y + length ∗ sin(angle))
    fov add line(start, end)
visibility ← default
for each line in fov
    points ← get points on line                    // algorithm below this one
    visibility += points count till obstacle
visibility \= fov count
```

Algorithm 3.7 Points on line

This method returns all *points* on a line where:
x1 = line's start point x value, *y1* = line's start point y value
x2 = line's end point x value, *y2* = line's end point y value

```
if x2 - x1 == 0                                          // Vertical
    inc ← y2 > y1 ?1 : −1
    for (y=y1; y2 > y1 ? y <= y2 : y >= y2 ;y += inc)
        if x1 >= 0 and y >= 0
            points add (x1, y)
else                                                     // Slope
    m ← (y2−y1)/(x2−x1)
    if (y2-y1) > (x2-x1)
        inc ← y2 > y1 ?1 : −1
        for (y=y1; y2 > y1 ? y <= y2 : y >= y2 ;y += inc)
            nx = (y−y1)/m + x1
            if nx >= 0 and y >= 0
                points add (nx, y)
    else
        inc ← x2 > x1 ?1 : −1
        for (x=x1; x2 > x1 ? x <= x2 : x >= x2;x += inc)
            ny = (x - x1) m + y1
            if x >= 0 and ny >= 0
                points add (x, ny)
```

The benchmark score is a singular value assigned to an approach to rate the performance of that approach for the given environment(s). This score is calculated from the success rate, path length, path smoothness, visibility, and time scores.

This score combines the path length and visibility rating with a single weight to reflect the trade-off between the metrics. The visibility or the safety distance from obstacles comes at the cost of path length. This associated path weight p_w indicates the importance of one or over the other. The other two measures are each associated with their own weights and then normalized all with each other.

The most important measure is still the success rate. This is why the success rate is multiplied by the results of the previous calculations to act as a limiter for the final score.

$$x = s_r \frac{p_l p_w + v(1 - p_w) + t_t t_w + p_s p_{sw}}{1 + t_w + p_{sw}} \tag{3.6}$$

where:

s_r is the success rate

p_l is the path length

p_s is the path smoothness

v is the visibility rate

t is the time

t_w is the time weight

p_w is the path weight

p_{sw} is the path smoothness weight

3.6 Chapter Summary

This chapter presents the benchmarking framework, which encompasses the evaluation methodology, maps, and navigational techniques. The aim of the methodology is to evaluate the performance of navigational approaches in identical conditions. It achieves this performance evaluation by assessing the success rate, path length, path smoothness, visibility and time metrics. The approach is benchmarked relative to a baseline and results in scores on each metric and an overall benchmarking score. The environments used to assess the different approaches are detailed in this chapter and in Appendix 6.3. The following chapters will utilise this framework's capabilities to:

- discover the optimal hyper-parameters for an approach
- evaluate the improvement of two similar approaches e.g. the Potential Field and the Pheromone Potential Field

- benchmark all the navigational approaches mentioned in this chapter

Chapter 4

Evaluating Potential Field Approaches

Potential fields are one of the most fundamental methods in mobile robotic path planning, navigation, and exploration. Due to this importance, an evaluation of the Potential Field navigation approach is shown from the perspective of benchmarking them. This chapter evaluates a basic Potential Field and demonstrates the quantifiable improvement of the novel Pheromone Potential Field approach to the Potential Field approach. Initially, a more in-depth look of the Potential Field will be taken, with a focus on the stability of the algorithm by exploring and demonstrating the range of successful and practical parameters. This is followed by an analysis of its performance in various environments. The second part of this chapter introduces the Pheromone Potential Field approach. A similar analysis is performed and demonstrated. The two approaches are then evaluated using the evaluation framework in different environments and the results are presented.

4.1 Potential Field Parameter Evaluation

Fundamentally, a potential field [12] is an approach that utilizes a multitude of forces to navigate an environment and avoid obstacles. The attractive force is a gravitational-like pull that draws the robot towards the goal location which can be written as follow:

$$F_a = \vec{\Theta_{rg}} c \frac{1}{d_{rg}} \tag{4.1}$$

where:

F_a is the attractive force between the robot and the goal

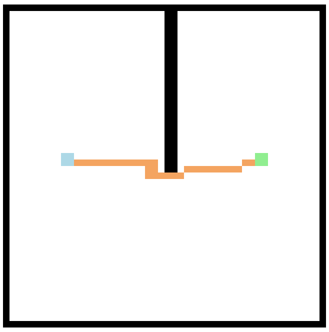

Figure 4.1: This figure shows a Potential Field approach navigating from the grey start location towards the green target location whilst avoiding the wall separating them.

d_{rg} is the distance between the robot and the goal

c is the attractive force constant

Θ_{rg} is the direction from the robot to the goal

The repulsive force pushes the robot away from an undesired location and can be written as follow:

$$F_{ri} = \begin{cases} \vec{\Theta_{ri}} k(\frac{d_{max}}{d_{ri}}), & d_{ri} < d_{max} \\ 0, & d_{ri} \geq d_{max} \end{cases} \tag{4.2}$$

where:

Θ_{rg} is the direction from a robot to a location i

F_{ri} is the repulsive force between the robot and a location i

k is the repulsive force constant

d_{max} is the maximum distance the repulsive force has an effect

d_{ri} is the distance between the robot and a location i

The sum of all the repulsive forces results in the total repulsive force F_r which is then summed with the attractive force to produce:

$$F = F_a + F_r \tag{4.3}$$

Figure 4.1 shows the navigation of the mobile robot in an environment with the Potential Field approach where the robot and goal are separated by a wall.

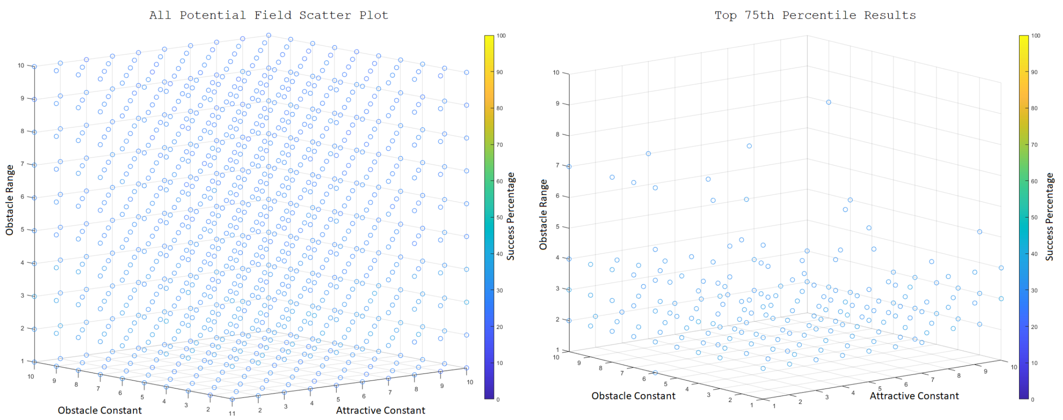

Figure 4.2: The left plot shows the average success rate of the Potential Field for a range of parameter combinations. The results show that the average success rate falls below 30%. The right plot filters the results to the highest quartile of success rate, which demonstrates a clear preference for low obstacle ranges.

Research into potential fields has demonstrated significant limitations in this approach [12]. Furthermore, the present study has also discovered that this approach suffers from having a limited range of usable parameter values, if a high rate of successful navigation for any given environment is desired. This section aims at providing more insight into this alongside establishing the optimal set of parameters for performance evaluation. The optimal parameter set for these environments is used in the experiments of this chapter and the following.

To evaluate the effective parameter range, an experiment was conducted that performs parameter sweeps across all the different environments. For this experiment, the static environments were utilized, and the robot and the goal locations were pseudo-randomly spawned in accordance with the map configurations specified in Section 3.3.1. These trials explored the effective parameter ranges of the d_{max}, k, and c parameters over the range of 1 to 10 with a step of 1.

Figure 4.2 presents the average success rate of a combination of parameters on the potential field approach across all the different environments. The results demonstrate that whilst most parameter combinations can lead to *a* result, none score better than 38%. Analyzing the top quartile of results on success rate the Potential Field approach is more optimal in low obstacle range settings. The results shown in Table 4.1 demonstrate that this approach has an overall low success rate of 28.02%. Grouping the approaches by map type shows that there is clear evidence that the effectiveness of this approach is strongly tied to the environment. Grouping the trials by the parameter combinations shows that

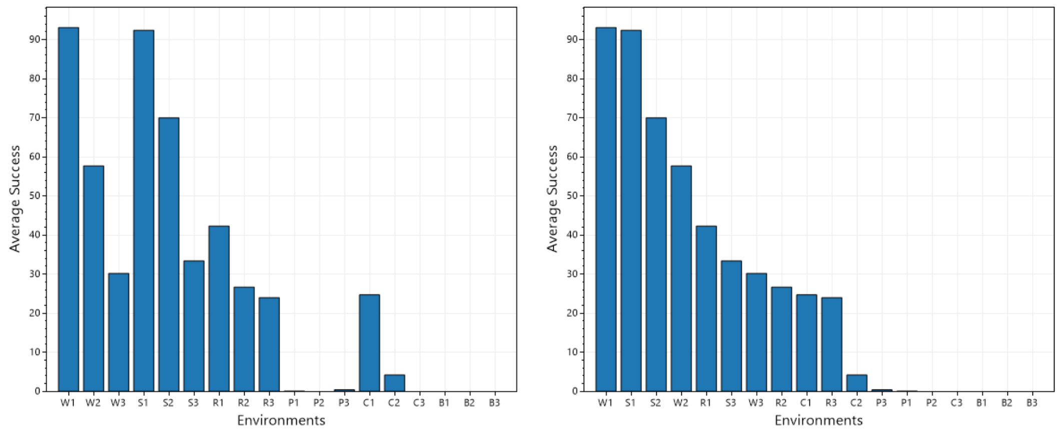

Figure 4.3: This left plot shows the average success rate of the Potential Field across the different map types grouped by the map types. The right plot orders the results by the average success rate on that map. The numbers stand for the complexity level of a map type with 3 being the most complex. Maps are abbreviated as follow; W: Wall, S: Slit, R: Room, P: PlankPile, C: Corridor, B: Bugtrap.

the approach produces similar results across a variety of maps. Furthermore, excluding the level one maps which contain minimal obstacles the results remain similar, further showing that even in difficult environments, the environment itself plays a larger role than the parameters.

Potential Field	Average success rate: **28.03%**					
	Std	Min	25%	50%	75%	Max
By map type	31.90	0.00	1.83	17.8	44.27	95.02
By parameter set	2.75	21.39	26.11	27.50	29.44	37.50
By parameter set - excl. level one maps	2.75	12.50	18.75	20.83	22.92	29.17

Table 4.1: This table shows the Potential Field parameter run average success rates when grouped by map type and parameter set. The final row shows the grouping by parameter set excluding the level one complexity maps

Figure 4.3 presents the results from the average success rate of the potential field in each distinct environment. The left plot has the maps of similar types grouped together whilst the other plot has them ordered by success. These results demonstrate that this approach has increased difficulties with the increase of objects in the environment ultimately leading to little success with maps designed around local minima e.g. the plank pile, and bug trap maps. Whilst no single parameter combination has a success rate over 38% there are environments that average over 90% success rate further demonstrating the importance of the environment over the parameter combination.

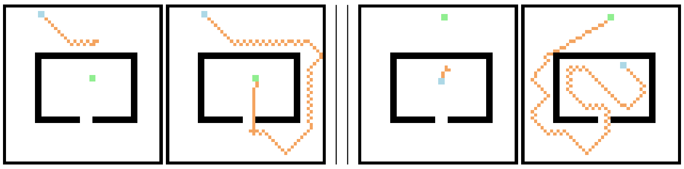

Figure 4.4: This figure shows two trials where the Potential Field (on the left of each trial) and the Pheromone Potential Field (on the right of each trial) attempt to navigate a bug trap map from the grey starting location towards the green target location whilst avoiding the walls separating them.

4.2 Pheromone Potential Field Parameter Evaluation

The primary problem with potential fields as demonstrated in the previous section is the state of equilibrium that the robot reaches local minima. To overcome the local minima problem of the potential field approach, a new force is introduced that attempts to push it out of this sub-optimal solution toward the goal [74]. In this research, we present a novel approach that is inspired by the way ants navigate between the nest and their food source [75].

The Pheromone Potential Field field is an extension of the potential field algorithm in which the robot continuously drops artificial pheromones on the path it has travelled. These pheromones are represented by non-obstacle point locations on the map and increase in strength when the robot traverses over them. Unlike ants, these pheromones act as a minor repulsive force motivating the robot to avoid locations it has previously visited, indirectly instilling exploratory behaviour. More importantly, if a robot were to be stuck in a local minimum the increase in the strength of the pheromones would generate an escape force pushing the robot out of this trap and onwards.

Figure 4.4 shows the result of a trial performed on the bug trap map between the potential field and the pheromone potential field. The robot spawning on the northern part of the map aims to get to the goal location marked with the green square in the centre of the map. Whilst the potential field reaches equilibrium early on, the pheromone potential field approach is pushed to explore away from the local minima, eventually making its way to the goal location.

The pheromone potential field adds a new force called the pheromone force. This force is a sum of all the repulsive forces generated from the pheromones in the range of the robot. This force adds three more parameters to the basic potential field being: the range in which pheromones affect the robot, the pheromone strength increase with each traverse, and a constant. Similar to the

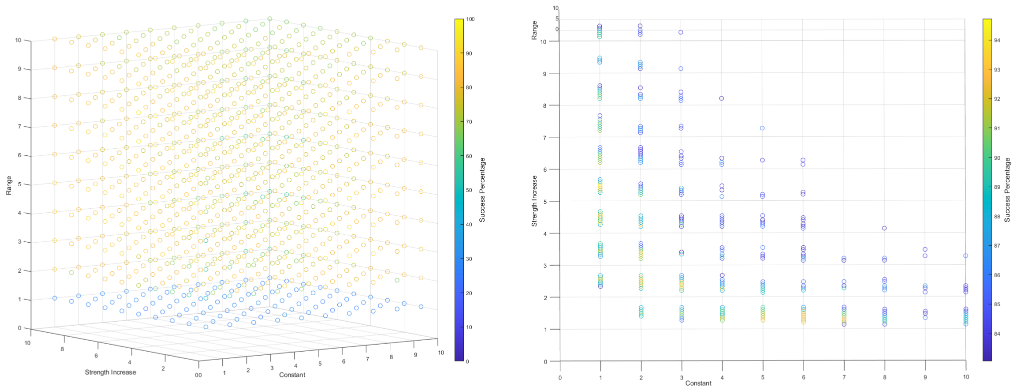

Figure 4.5: This left plot shows the average success rate of the Pheromone Potential Field for a range of parameter combinations. The results show a significant improvement in success rate compared to the Potential Field approach, especially on higher ranges. The right plot is focused on the relationship between the pheromone's strength increase and the constant.

previous section, an experiment on the Pheromone Potential Field approach was performed that evaluates the effective range of parameters across all the different environments.

Figure 4.5 presents the average success rate of trials grouped by the three parameters. These results demonstrate that this approach

- has a clear preference for longer pheromone inclusion ranges
- has a relationship qualitatively similar to inverse proportional hyperbolic relationship, between the strength increase and constant parameter
- a significant improvement in the success rate when compared to the Potential Field approach

Table 4.2 shows the resulting average success rate grouped by map types and parameters. Whilst the environment is still an important factor, this importance has been reduced with the Pheromone Potential Field approach. This surmise is inferred from the drop in standard deviation when grouped by map type and the relatively high success rate at the 25th percentile. A similar observation can be seen in the 25th percentile when grouping by parameter combination indicating that whilst there are some combinations that result in a low average success rate the majority of parameters do perform well. The low rating parameter groups are the low range pheromone settings as shown in Figure 4.5.

Pheromone Field	Average success rate: **74.05%**					
	Std	Min	25%	50%	75%	Max
By map type	19.95	29.68	66.38	78.52	84.08	98.67
By parameter set	16.76	23.61	71.94	78.33	83.89	94.72
By parameter set - excl. level one maps	17.63	15.42	64.17	71.67	78.33	93.75

Table 4.2: This table shows the Pheromone Potential Field parameter run average success rates when grouped by map type and parameter set. The final row shows the grouping by parameter set excluding the level one complexity maps

4.3 Benchmarking the approaches

In this section, we aim to evaluate the performance of the Potential Field approach and the Pheromone Potential Field approach with the optimal parameter set combinations obtained from the previous section. This experiment is conducted in static environments where both approaches are run alongside the baseline approach to calculate the performance. The previous two sections discussed the potential field success rate. This section will look into the path length metric, the path smoothness metric, the visibility metric, and the performance score.

Path length metric Figure 4.6 show the path lengths generated by both approaches where the dotted lines show all the runs and the solid lines show all the successful runs. The maximum path length permitted is 1000 units which is achieved at the 25th percentile of path lengths for all Potential Field runs. This in combination with the low path length at successful runs highlights that the Potential Field approach works best in short ranged navigation where the goal is nearby. By contrast, failure from path length is only encountered near the 100th percentile for all Pheromone Potential Field runs, indicating that the majority of navigation attempts succeed. Furthermore, the Pheromone Potential Field has a low overall path length but is still able to achieve results close to the maximum path length permitted demonstrating the ability to explore till the goal is achieved.

Path Smoothness Evaluation Figure 4.7 shows the smoothness of the paths generated by the approaches. The Potential Field approach across all runs scores significantly worse than the Pheromone Potential Field approach, but successful runs have comparable or better results than the Pheromone Potential Field approach.

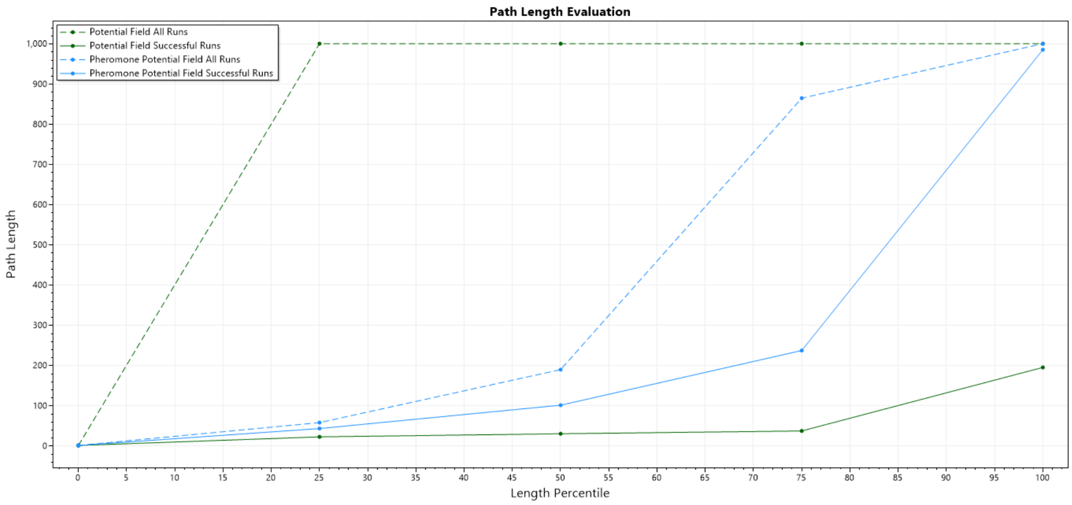

*Figure 4.6: **Lower is better.** This figure shows the path lengths the Potential Field and Pheromone Potential Field approaches generate. The dashed lines represent all runs whilst the filled lines represent all successful runs. Note that the Potential Field , with an average success rate of **28.03%**, is restricted to successful runs with paths under 200 units and in the first quartile of length. However, the Pheromone Potential Field , with an average success rate of **74.05%**, demonstrates successful solutions across all lengths of paths available.*

Visibility metric Figure 4.8 shows the visibility metric of both approaches. The visibility metric measures the amount of free space within the robot's FOV there is at every step and averages the values for each approach. Both approaches score similarly on this metric with the successful run having a higher visibility score. This indicates that higher visibility scores are not just a measure of safety, but could also be an indication of likely navigational success.

4.4 Closing statement

In this chapter, the benchmarking framework is used to evaluate and benchmark two similar approaches individually and in comparison to each other. Here, the Potential Field algorithm was evaluated and benchmarked using the framework presented in the previous chapter. Whilst the results confirmed known limitations [12] and a low success rate, they also demonstrated the importance of considering short object ranges to increase the success rate of this approach. This chapter also presented the Pheromone Potential Field approach, which introduced a pheromone force to the Potential Field approach. This novel approach demonstrated significant improvements on the basic Potential Field approach across all metrics except path smoothness.

The framework for evaluating these approaches demonstrates in a quantitative measure the improvements the Pheromone Potential Field approach has

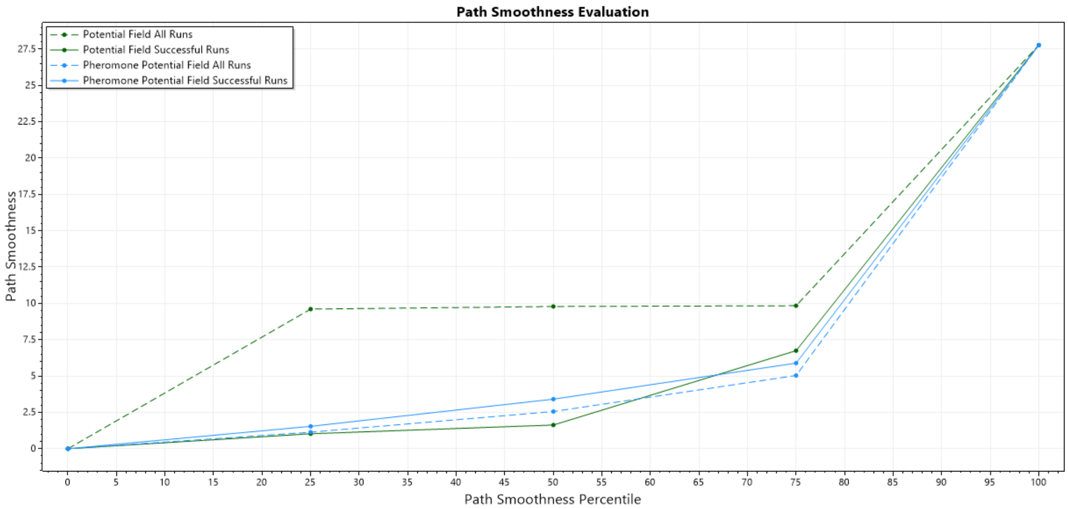

Figure 4.7: ***Lower is better.*** *This figure shows the smoothness of the paths Potential Field and Pheromone Potential Field approaches generate. The dashed lines represent all runs whilst the filled lines represent all successful runs. Note that the Potential Field approach across all runs scores significantly worse than the Potential Field . However, successful runs have comparable or better results than the Potential Field .*

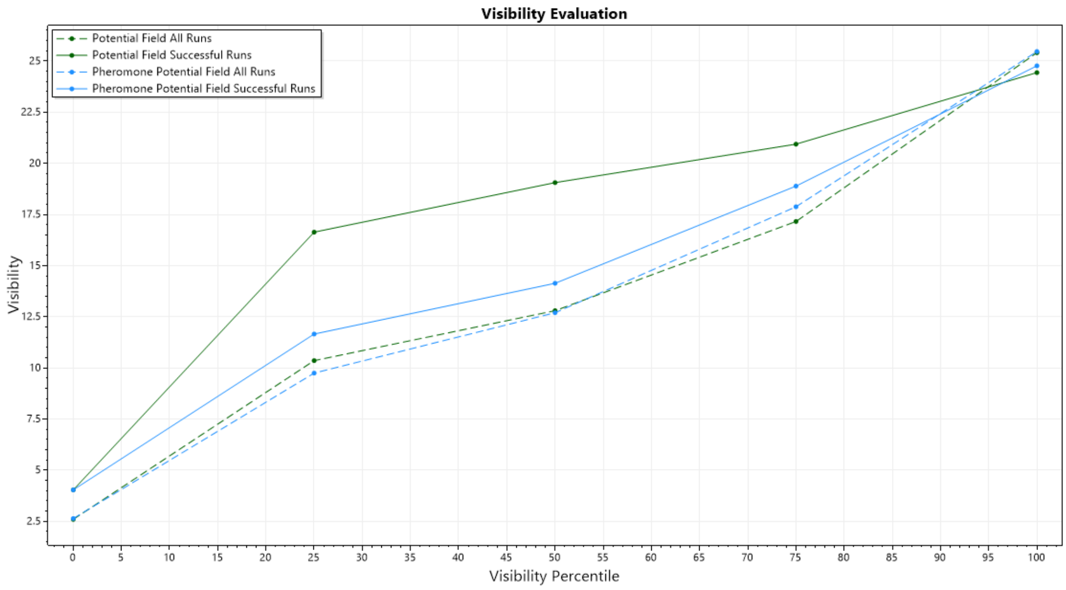

Figure 4.8: ***Higher is better.*** *This figure shows the average visibility around the robot the Potential Field and Pheromone Potential Field approaches generate. The dashed lines represent all runs whilst the filled lines represent all successful runs. Note that for both approaches the successful runs have a higher visibility score. This indicates that higher visibility scores are not just a measure of safety, but could also be an indication of likely navigational success.*

on the Potential Field approach. This framework also highlighted several shortcomings of the Pheromone Potential Field approach. Improvements to this novel approach will focus on addressing the path smoothness and the relative lower success rates in plank pile and corridor environments.

These findings demonstrated that the framework has similar parameter tuning and metric evaluation capabilities to other frameworks like ROS [8]. However, this framework extends on these frameworks by indicating the exact problem environments and metrics to address for improvements alongside several beneficial characteristics of the approaches.

The following chapter will look to extend this work by evaluating the performance of the Potential Field approaches alongside an assortment of different approaches.

Chapter 5

Benchmarking navigational approaches

The previous chapter evaluated the performance of a navigational approach in isolation and comparison with a similar type of approach. This evaluation of the Potential Field and Pheromone Potential Field approaches demonstrated the framework's capabilities in quantitatively determining performance improvement.

This chapter follows up on these capabilities by evaluating a more extensive set of approaches with respect to their performance within specific problem environments. Furthermore, this chapter expands the evaluation to generated environments along with the static environments utilized so far. Here, the results from the proposed framework show the significant success rate of A* but also demonstrate this approach's shortcomings in the visibility metric.

The first section details the framework's experimental setup used to generate the results discussed in this chapter. Section 5.2 presents the comparison between the base RRT approach with the improved RRT* approach. Section 5.3 benchmarks the different navigational approaches across all the environments and discusses the findings. The chapter concludes with section 5.4 that summarises the findings.

5.1 Framework's experimental setup

This section documents the experimental conditions in which the framework evaluated the efficiency of the navigational approaches across all the environment types. The approaches explored in this experiment are the Potential Field, Pheromone Potential Field, A*, RRT, and RRT* (See section 3.4). The framework implemented certain constraints for these experiments to demonstrate more realistic scenarios. This is especially important in the larger and cluttered maps where approaches converge very slowly to a solution making it impractical for real-world usage. Table 5.1 shows these parameters and fail conditions. Alongside the static maps presented in section 3.3.1 this chapter also utilizes the dynamically generated maps presented in section 3.3.2. The configuration details for these maps and sample-generated environments are shown in their respective sections.

Parameter Name	Value	Short Description
Robot FOV	30	the length of the robot's field of view
Max Step	1000	maximum amount of steps the robot can perform
Max Time	10,000	maximum amount for all stages of the navigation in ms
Stuck Condition	10/40	10 number of similar steps to points in any 40 consecutive steps
Trials per configuration	100	the frequency an approach is run on each map type
Time Weight	0.75	the time weight used to calculate the approach's score
Path Weight	0.75	the path weight used to calculate the approach's score
Path Smoothness Weight	0.25	the path smoothness weight used to calculate the approach's score

Table 5.1: This table details the framework's parameters.

The experiment conducted in this framework consists of 100 trials per map type. At each trial, the robot and goal spawn locations were either random or pseudo-random per the configuration details. Firstly, the framework determined a baseline by running an A* algorithm[1] on the explored map. This determined the connectivity of the map from the robot's starting location to the goal location in the generated maps and enabled the performance calculations for the path length and execution duration. After the framework performed

[1] This approach differs from the incremental A* approach used to explore unknown environments in it that has a fully explored map. Therefore, it is closer to a graph traversal than actual navigation.

the baseline calculations, it evaluated each approach under the same configuration. Unlike the baseline calculations, the map is unknown for each method. Each method only knows the robot and the goal location at the starting point. The results from these trials are discussed in this chapter, with the full results available in Appendix 6.3.

5.2 Comparison of the RRT approaches

The first notable results from the experiment's results, as noted in Table 5.2, showed that the RRT approach scored ever so slightly higher than the RRT* approach. These results prompted a further investigation into the nature of these results, of which the findings are presented here. This section aims to evaluate the performance of the RRT* approach compared to its base RRT approach and benchmark this improvement.

Approach	Score	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
RRT	**0.67**	0.82	0.83	**0.88**	**0.88**	**0.49**
RRT*	0.66	**0.83**	**0.87**	0.87	0.81	**0.49**

Table 5.2: The performance difference between the RRT and the RRT approach. All metrics: higher the better.*

Table 5.2 demonstrates that the two approaches have similar scores overall. The RRT* outperforms the RRT in both the success rate and the path score whilst being surpassed in both the path smoothness score and the time score. The time score is the metric with the highest difference between the two approaches, and the speed of the RRT approach relative to the RRT* is why it scored higher. The difference between the approaches is that the RRT* performs two additional operations: the best neighbour operation and the tree rewiring operation. These operations lead to improved and shorter paths but also lead to increased computational time [76]. This additional time is noticeable both in the overall performance (seen in Table 5.2) and the failure scores (seen in Table 5.3. Table 5.3 shows where the different approaches failed. The table shows whether the approach failed by hitting the maximum step count, or the maximum duration permitted. These results show that both approaches fail about half of the generated environments' runs due to the time or step count restrictions. However, the RRT* approach almost exclusively fails in time.

All Map Types				
Approach	Step Fail	Time Fail	Total Fail	Total Runs
RRT	179	315	494	2800
RRT*	1	479	480	2800
Static Map Types				
Approach	Step Fail	Time Fail	Total Fail	Total Runs
RRT	2	0	2	1800
RRT*	0	0	0	1800
Generated Map Types				
Approach	Step Fail	Time Fail	Total Fail	Total Runs
RRT	177	315	492	1000
RRT*	1	479	480	1000

Table 5.3: This table presents the failed runs for the RRT and the RRT approach showing whether an approach failed due to time constraints or the maximum amount of steps.*

Figure 5.1 shows that in the navigation of static environments, the two approaches score similarly, with the RRT* outperforming the RRT approach slightly in all maps except for the RoomThree map. When evaluating the results of the generated environments, as presented in figure 5.2, one can note that both approaches struggle more in larger cluttered obstacle maps. However, the RRT is relatively consistent in the tunnel environment and outperforms the RRT* significantly in the higher complexity tunnel maps.

These findings indicate that the RRT* performs better and generates shorter paths within an acceptable time frame in smaller and less cluttered environments. As maps grow more extensive and increasingly limiting in open space, the effectiveness of this approach decreases significantly to the point where the RRT outperforms it. These results suggest that the RRT* is more favourable in large open space areas with decreased clutter. As the average free space decreases below 70% in obstacle environments and 30% in tunnel environments, the RRT approach is a more optimal solution. The resulting framework analysis can be utilized in exciting ways discussed in more detail in the following chapter.

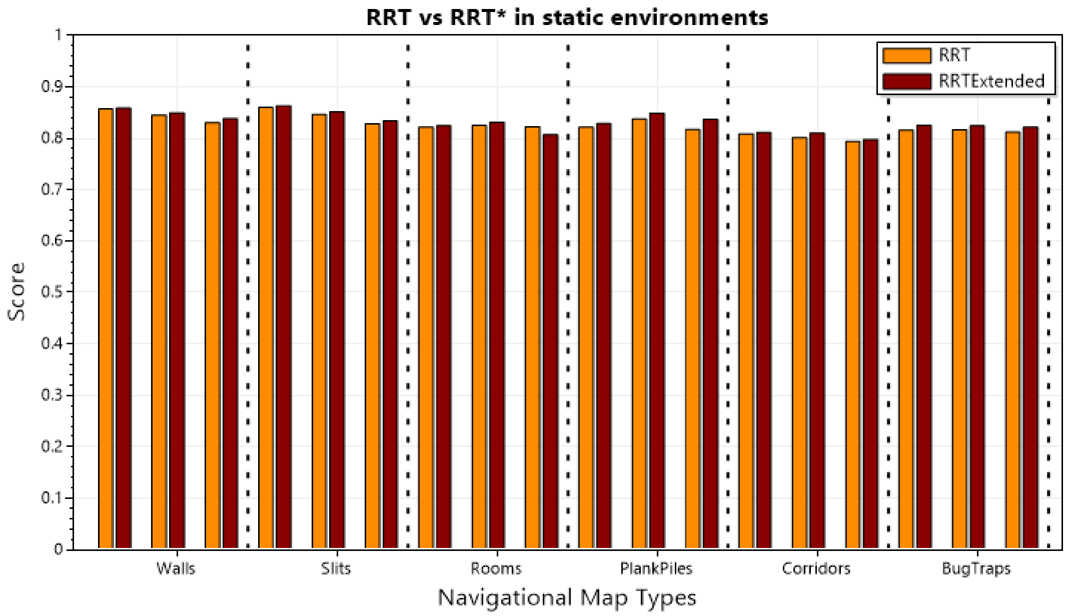

Figure 5.1: This figure shows the results of the RRT and the RRT approaches in the static environments. Each type of environment is separated by a vertical dotted line. Each couple of bars represents one complexity level within the type of environment increasing in difficulty to the right.*

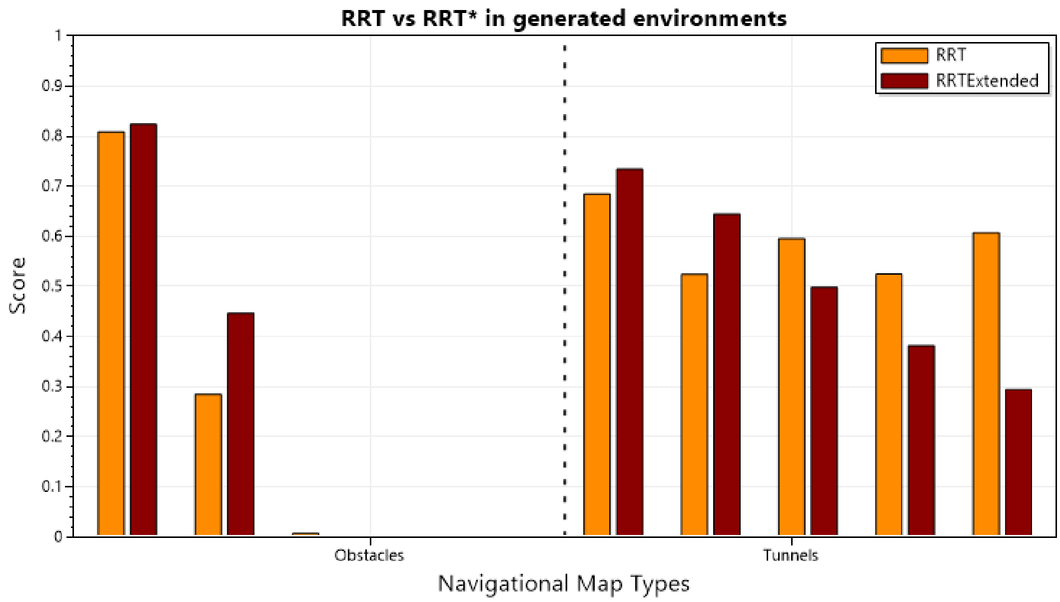

Figure 5.2: This figure shows the results of the RRT and the RRT approaches in the dynamically generated environments. The area on the left holds the obstacle maps whilst the area on the right holds the tunnel maps. Each couple of bars represents one complexity level within the type of environment increasing in difficulty to the right.*

5.3 Performance of the navigational approaches

This section will present the results of the performance of each approach and in comparison with each other. The full results on each map type are presented in Appendix 6.3. Table 5.4 presents the scores of each approach across all the static and generated environments. Table 5.5 and Table 5.6 present the results of the static and generated environment respectively. The comparison between the approaches is first highlighted under two metrics: the success rate and the visibility score. This is due to the interesting trends observed under these two metrics.

Approach	Score	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
Base Line	0.84	1.00	0.90	0.91	0.90	0.45
A*	**0.82**	**0.99**	0.88	**0.91**	**0.89**	0.42
RRT	0.67	0.82	0.83	0.88	0.88	0.49
RRT*	0.66	0.83	0.87	0.87	0.81	0.49
Potential Field	0.16	0.18	**0.90**	0.83	**0.90**	**0.59**
Pheromone Field	0.62	0.76	0.86	0.84	**0.89**	0.48

*Table 5.4: All metric results on **All** environments. Each metric is in the range of [0.0-1.0] averaged across all maps. Higher is better.*

Approach	Score	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
Base Line	0.85	1.00	0.90	0.91	0.90	0.51
A*	**0.84**	**1.00**	**0.89**	**0.90**	**0.90**	0.49
RRT	**0.83**	**1.00**	0.85	0.87	**0.89**	0.53
RRT*	**0.83**	**1.00**	0.88	0.87	0.88	0.51
Potential Field	0.24	0.29	**0.90**	0.82	**0.90**	**0.57**
Pheromone Field	0.75	0.91	0.86	0.85	**0.89**	0.52

*Table 5.5: All metric results on **Static** environments. Note due to the exceptionally low score of the Potential Field approach its results are excluded from the metric high score.*

5.3.1 Performance under environments and metrics

Table 5.5 and Figure 5.3 show the results of all approaches in static environments. These results, with the exception of the Potential Field approach, score roughly the same. However, when evaluating the scores in the generated environments, as shown in Table 5.6 and Figure 5.4, A* maintains its dominance as complexity grows. Similarly, RRT performs better than RRT* as environment

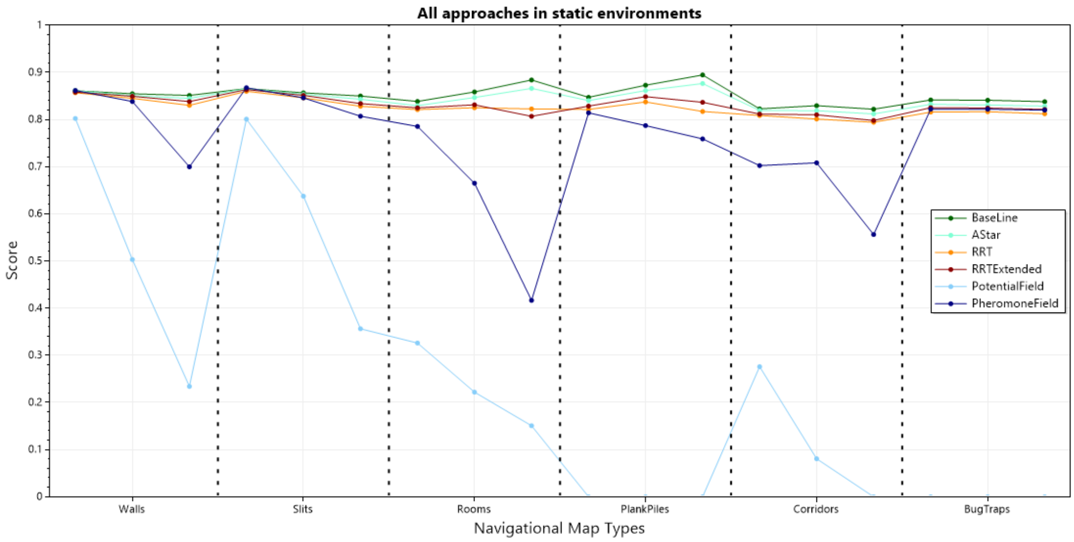

Figure 5.3: This figure shows the results for all the approaches in the static environments. Each type of environment is separated by a vertical dotted line. The complexity level within the type of environment increases in difficulty to the right.

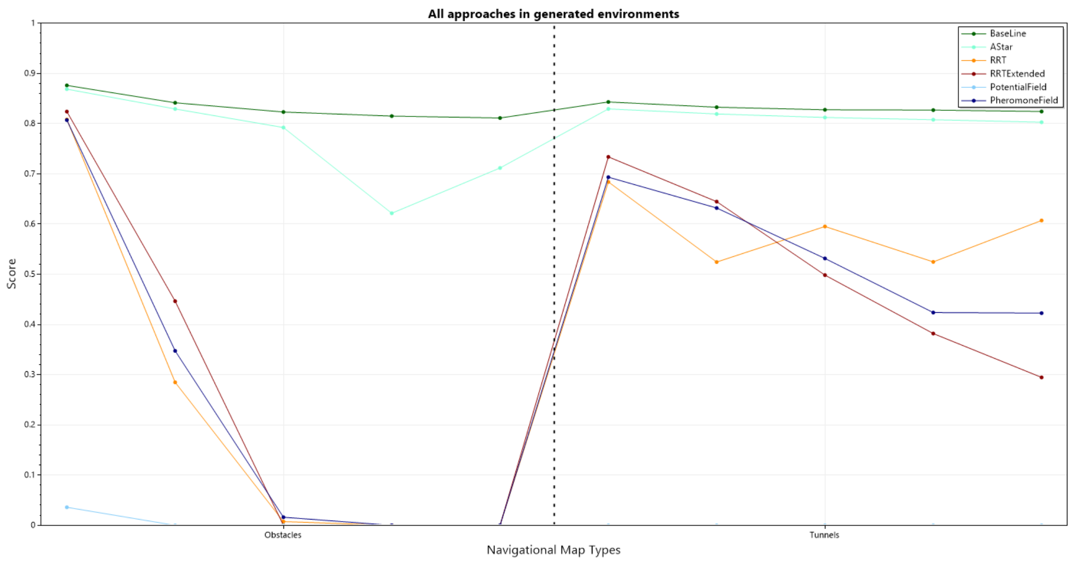

Figure 5.4: This figure shows the results for all the approaches in the generated environments. Each type of environment is separated by a vertical dotted line. The complexity level within the type of environment increases in difficulty to the right.

Approach	Score	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
Base Line	0.83	1.00	0.90	0.92	0.90	0.34
A*	**0.79**	**0.97**	**0.87**	**0.92**	0.88	0.31
RRT	0.40	0.51	0.79	0.89	0.87	**0.42**
RRT*	0.38	0.52	**0.86**	0.87	0.64	**0.42**
Potential Field	0.004	0.004	0.90	0.95	0.90	0.84
Pheromone Field	0.39	0.48	**0.86**	0.81	**0.90**	0.40

*Table 5.6: All metric results on **Generated** environments. Note due to the exceptionally low score of the Potential Field approach its results are excluded from the metric high score.*

complexity increases. Most notably, Pheromone Potential Field always performs better than the Potential Field approach. Combining all the information and evaluating across all environments, the results in Table 5.4 show that A* is the most successful algorithm. This success is closely followed by the RRT , RRT* , and the Pheromone Potential Field approaches. The Potential Field approach has performed significantly worse. These results show the usefulness of the generated environments in the proposed framework to bring performance insights on the different approaches under various degrees of difficulty.

Even though the success rate is a valid metric to test the efficacy of an approach under a given experimental setup, other metrics such as 'Visibility score' are also crucial from a safety perspective. At the same time, it is essential to note that other metrics are only accumulated when a run of an approach succeeds. As a result, if an approach fails many times, the sampling rate also falls, which could make other metrics skewed towards success under easy trials. This is where static environments are suitable for evaluating other metrics because most of (more than 90%) the trials succeed in all the approaches except in Potential Field as shown in Table 5.5.

We can observe from Table 5.5 that A* has the lowest visibility score among all the approaches, which indicates that A* paths are too close to walls and other obstacles. This does generate specific concerns with safety-critical applications. The RRT approach would be a more optimal solution in these scenarios. Even though Potential Field shows the best visibility score, it is of little use because its success rate is too low. However, the same cannot be said about other metrics where A* is consistently among the top performers, with the rest of the approaches remaining competitive.

5.3.2 Approach-wise analysis

In this section, trends peculiar to a specific approach are noted. The following analysis highlights the use case of any given approach based on the above results.

The A* approach performs significantly better than the other approaches in the overall score metric. This iterative A* approach generates a new plan every time a previous plan could not be followed due to the existence of an obstacle as it explores an environment. This score is primarily due to the significant success rate of this approach across the different environments. However, when evaluating individual metrics, the other approaches have comparable or better scores, with this approach specifically performing poorly on the visibility score. This poor score indicates that this approach often moves close to obstacles and is biased towards the shortest path. This suggests that if time and visibility scores are essential for a specific application, it is better to look at the performance of other approaches in a given environment.

The RRT and RRT* have similar performance ratings primarily due to the limitations discussed in the previous section. The results indicate that the RRT* has a better success rate by a small margin with shorter paths. The apparent limitation is the computational time where this approach scores worse than the RRT even in environments where it does reach the goal location. However, RRT and RRT* have outstanding visibility scores across all environments. Hence, such methods can be beneficial if we do not have strict computational limitations.

The Potential Field and Pheromone Potential Field have significantly different results. The Potential Field approach suffers from a meagre success rate which is the primary driver behind the low benchmarking score. The individual metrics score high due to the Potential Field only being successful in simple and almost unobstructed environments. However, the extremely low computational cost of this approach is worth noting. The Pheromone Potential Field has a comparable score to the RRT and the RRT* , with the key difference being computational time efficiency. This difference is particularly evident in the generated environments where the Potential Field approaches outperform the RRT* approach. Due to efficient computational time, environments with low obstacle counts and high visibility are best suited for the Pheromone Potential Field approach. In particular, the Pheromone Potential Field outperforms the baseline in these WallOne and SlitOne maps.

The following section will make recommendations based on the above results on how to choose different approaches based on the complexity of the environment and the importance of desired metrics.

5.4 Summary of findings

This chapter demonstrates the effectiveness of the proposed framework. It allows various experimental setups throughout the different approaches and evaluates them using the same evaluation methodology. This section will look into the results presented in this chapter. These results are summarised, and based on these results, this section will make recommendations based on how to choose different approaches based on the complexity of the environment and the importance of desired metrics.

The findings of this chapter are as follows:

- **Similar score of RRT and RRT* .** The experimental results showed that whilst the RRT* has a slightly improved success rate and path score, the time requirement of this approach counteracts this. These findings conclude that when choosing between these two approaches, the time requirement in smaller and uncluttered maps is negligible and therefore, the RRT* approach is the more optimal solution. As the average free space decreases below 70% in obstacle environments and 30% in tunnel environments, the RRT approach is a more optimal solution.

- **The A* approach is the best approach in this experiment.** This approach scored near baseline scores on all metrics. The limitations are in the form of the time requirements and visibility score when it comes to larger, more complex environments. This approach can serve as the base when navigating unknown environments except for in open and uncluttered environments where Potential Field algorithms will outperform.

- **Higher complexity obstacle maps** suffered from the lowest success rates of all different map types. More optimal approaches are needed for these types of environments.

These findings demonstrate the framework's ability to produce a far greater aggregation of information and evaluation of approaches than the other comparable frameworks in unexplored environments. It extends the common capability of finding optimal approaches while also highlighting the complex problem environments that need further research efforts. The results can also be used by adaptive planners to switch between approaches dependent on the environment type they are in. A good example of this is the effect of free space on the optimality of the RRT and the RRT* approaches.

Chapter 6

Conclusions and Future Research Directions

This thesis presented an evaluation methodology and framework to benchmark the performance of mobile robotic navigation in unknown environments. These contributions were used to find optimal parameter sets for a range of approaches, evaluate the performance of a method in isolation, and evaluate the performance when compared to similar and different techniques. This research project addressed the limitations in current frameworks by evaluating navigational approaches with all important metrics across various approaches. The contributions made in this research can be used to improve new algorithms and highlight deficiencies in existing approaches.

The final chapter starts by summarising the results and achievements of this research and concludes with a section on recommendations for future research directions.

6.1 Summary of contributions of the thesis

6.1.1 Navigational Environments

This research project describes a set of static environments that address specific common problems in navigational robotics. These problem environments can be summarised as simple obstacles, narrow corridors, traps, and the abstraction of human-like maps. The static environments increasingly become larger, more complex, and restrictive in requirements. This complexity introduces common issues in mobile robotic navigation like narrow constrictions, local minima, and safety clearance.

Furthermore, procedurally generated maps that extend the human-like maps and cluttered obstacle maps are introduced. The generated obstacle environments increase in complexity from an average free cell rate of 86% to a rate of 32.59%. The results from the environments show that navigational approaches, with the exception of A* struggled, with average free cell rates below the 52%.

All maps are detailed and documented in this research project, with recommendations for future maps in the following section.

6.1.2 Pheromone Potential Field

Initial experimentation with the Potential Field approach in this research project's benchmarking framework emphasised certain limitations. These observed limitations motivated the development of the Pheromone Potential Field approach. The Pheromone Potential Field approach is an extension of the Potential Field approach where this approach deploys pheromones along its navigated path to avoid local minima and install a goal-directed exploration behaviour.

The experimental results have shown that this added behaviour significantly improved the success rate of this approach from an average of 28.03% through the Potential Field approach to 74.05%. Furthermore, this approach has comparable scores with the RRT and RRT* approaches. The Pheromone Potential Field approach was also shown to be the most optimal approach in a few environments.

The evaluation methodology also demonstrated several limitations with this approach. Whilst comparable in scores, this approach suffers from a lower average success rate when compared to the A* , RRT , and RRT* approaches. Improvements to this approach would focus on addressing the limitations in problem environments CorridorThree and RoomThree which are narrower environments, with a larger focus on delayed rewards in goal distance.

6.1.3 Evaluation Methodology

This research presented a method to benchmark the navigational approach. A framework combines the map generation and approach execution elements to evaluate the performance of these algorithms. The achieved benchmarking score evaluates the performance of an approach across several key metrics, including the path length, path smoothness, visibility rating, computational time, and success rate. The introduction of path smoothness is a metric to achieve paths reflecting different optimality criteria than shortest path length. Shortest path length criteria may lead to supposedly-optimal paths that introduce sudden changes of direction contrasting sharply with the smooth constant change of direction desired for real-world applications. The visibility rating metric looks into the robot's average clearance from obstacles. This average distance from obstacles is critical in applications with safety concerns. This score can be adapted with weights to better represent the needs for safety-critical and resource-scarce systems. The evaluation methodology challenges common assumptions that the RRT approach is improved by the RRT* . These findings highlight the importance of benchmarking navigational robotic approaches in a general framework.

This research project proposed a novel benchmarking evaluation methodology that combines all evaluated metrics in a comparable score. This score is accompanied by an aggregation of information that enables the indication of exact problem environments and metrics. The results lead this project to the introduction of a novel approach due to the clear indication of both success and shortcomings.

6.2 Future Research Directions

In this section, we discuss potential prominent research directions for future research projects based on the results reported in this thesis.

6.2.1 Integration with Robotic Frameworks

The literature review identified several key mobile robotic frameworks, of which ROS [8] [9] is the most commonly used. The integration with this framework would be beneficial to aid in the usage of this evaluation methodology in future research contributions.

6.2.2 Approaches inclusion

This framework has an iteration-based mechanism that advances the simulation in single steps, directed by the results from the navigational path planner. It would be appropriate to expand on the current mechanism to evaluate the performance of learning-based approaches and multi-robotic systems. This allows for a wider variety of approaches and would generate a more accurate and universal benchmark. A possibility would be to include the navigational mechanism in this framework and the learning-based mechanisms in PathBench [7] to expand on the MoveIt [5] framework. Furthermore, the author agrees with Moll et al. [69] that the file formats must be standardised to allow for community-wide usage.

6.2.3 Environment Improvements

The results from the benchmarking in Chapter 4 and 5 indicate the significant effect of the environment on navigation performance. An essential research direction would include more maps that address specific or common navigational problems. Future research should aim to expand this underdeveloped area with systematically generated maps that address specific concerns within the field of mobile robotics like dynamic environments, difficult terrain, large environments with resource-constrained robots, and aquatic-based exploration. Furthermore, metrics that better represent mobile robotics' resource usage are needed to benchmark these new proposed environments properly.

6.2.4 Adaptive motion planners

With the introduction of more environments and navigational approaches, the author believes that the framework can establish a correlation between an environment and its optimal approach. An adaptive planner that can recognise environments and utilise the most efficient approach for that environment could lead to a more efficient navigation overall than any one navigational approach. This research contributes efforts towards this in Chapter 5 with an example threshold being presented in Section 5.2 for switching between the RRT and RRT* approaches.

6.3 Final Summary

This research study presented a novel benchmarking framework that evaluated the performance of mobile robotic navigational approaches. This evaluation inspired the development of the novel Pheromone Potential Field approach that addressed several of the shortcomings found in the Potential Field approach. Furthermore, this study contributed several static and procedurally problem-specific environments.

The field of mobile robotics has a need for standardised benchmarking and evaluation for contributions. Future work aims to address this shortcoming by allowing all branches of navigational mobile robotics to be evaluated on this framework.

References

[1] G. Walter, "The living brain," *Ltd, London*, 1953.

[2] F. Martinez, E. Jacinto, and D. Acero, "Brownian motion as exploration strategy for autonomous swarm robots," in *2012 IEEE international conference on robotics and biomimetics (ROBIO)*, pp. 2375–2380, IEEE, 2012.

[3] V. Braitenberg, *Vehicles: Experiments in synthetic psychology.* MIT press, 1986.

[4] A. Muhammad, N. R. H. Abdullah, M. A. Ali, I. H. Shanono, and R. Samad, "Simulation Performance Comparison of A*, GLS, RRT and PRM Path Planning Algorithms," in *2022 IEEE 12th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 258–263, IEEE, 2022.

[5] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[6] R. Diankov and J. Kuffner, "OpenRAVE: A Planning Architecture for Autonomous Robotics," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34*, vol. 79, 2008.

[7] A.-I. Toma, H.-Y. Hsueh, H. A. Jaafar, R. Murai, P. H. Kelly, and S. Saeedi, "Pathbench: A benchmarking platform for classical and learned path planning algorithms," in *2021 18th Conference on Robots and Vision (CRV)*, pp. 79–86, IEEE, 2021.

[8] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[9] R. L. Guimarães, A. S. d. Oliveira, J. A. Fabro, T. Becker, and V. A. Brenner, "ROS navigation: Concepts and tutorial," in *Robot Operating System (ROS)*, pp. 121–160, Springer, 2016.

[10] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, pp. 287–290, 2022.

[11] 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.

[12] Y. Koren, J. Borenstein, *et al.*, "Potential field methods and their inherent limitations for mobile robot navigation.," in *ICRA*, vol. 2, pp. 1398–1404, 1991.

[13] M. O. Idries, M. Rolf, and T. V. O. Scheper, "Exploration: Do we need a map?," in *Annual Conference Towards Autonomous Robotic Systems*, pp. 476–479, Springer, 2019.

[14] P. Raja and S. Pugazhenthi, "Optimal path planning of mobile robots: A review," *International journal of physical sciences*, vol. 7, no. 9, pp. 1314–1320, 2012.

[15] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[16] B. Pang, Y. Song, C. Zhang, H. Wang, and R. Yang, "A swarm robotic exploration strategy based on an improved random walk method," *Journal of Robotics*, vol. 2019, 2019.

[17] C. Pérez-D'Arpino, C. Liu, P. Goebel, R. Martín-Martín, and S. Savarese, "Robot navigation in constrained pedestrian environments using reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1140–1146, IEEE, 2021.

[18] A. Davids, "Urban search and rescue robots: from tragedy to technology," *IEEE Intelligent systems*, vol. 17, no. 2, pp. 81–83, 2002.

[19] H.-y. Zhang, W.-m. Lin, and A.-x. Chen, "Path planning for the mobile robot: A review," *Symmetry*, vol. 10, no. 10, p. 450, 2018.

[20] B. Patle, A. Pandey, D. Parhi, A. Jagadeesh, *et al.*, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582–606, 2019.

[21] S. Russell and P.Norvig, "Artificial intelligence: A modern approach," *Prentice Hall Upper Saddle River, NJ, USA*, vol. 90, pp. 33–48, 2002.

[22] A. Stentz, "Optimal and efficient path planning for partially known environments," in *Intelligent unmanned ground vehicles*, pp. 203–220, Springer, 1997.

[23] W. Pijls and H. Post, "Yet another bidirectional algorithm for shortest paths," tech. rep., 2009.

[24] S. Permana, K. Bintoro, B. Arifitama, A. Syahputra, and et al., "Comparative analysis of pathfinding algorithms A*, Dijkstra, and BFS on Maze Runner game," *IJISTECH (International J. Inf. Syst. Technol., vol. 1, no. 2, p. 1*, 2018.

[25] I. Zarembo and S. Kodors, "Pathfinding algorithm efficiency analysis in 2d grid," in *ENVIRONMENT. TECHNOLOGIES. RESOURCES. Proceedings of the International Scientific and Practical Conference*, vol. 2, pp. 46–50, 2013.

[26] N. Barricelli, *Symbiogenetic Evolution Processes Realized by Artificial Methods*. 1957.

[27] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[28] F. Liu, S. Liang, and D. X. Xian, "Optimal path planning for mobile robot using tailored genetic algorithm," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 12, no. 1, pp. 1–9, 2014.

[29] H. Qu, K. Xing, and T. Alexander, "An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots," *Neurocomputing*, vol. 120, pp. 509–517, 2013.

[30] A. Bacchin, G. Beraldo, and E. Menegatti, "Learning to plan people-aware trajectories for robot navigation: A genetic algorithm," in *2021 European Conference on Mobile Robots (ECMR)*, pp. 1–6, IEEE, 2021.

[31] LaValle, S. M, *et al.*, "Rapidly exploring random trees: A new tool for path planning," 1998.

[32] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[33] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2997–3004, IEEE, 2014.

[34] Y. A. Prabowo, B. R. Trilaksono, E. M. Hidayat, and B. Yuliarto, "Utilizing a rapidly exploring random tree for hazardous gas exploration in a large unknown area," *IEEE Access*, vol. 10, pp. 15336–15347, 2022.

[35] J. J. Kuffner and S. M. LaValle, "RRT-connect: An Efficient Approach to Single-Query Path Planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2, pp. 995–1001, IEEE, 2000.

[36] I. Noreen, A. Khan, , and Z. Habib, "Optimal Path Planning using RRT* based Approaches: A Survey and Future Directions," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, 2016.

[37] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[38] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, "MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 2, pp. 1024–1031, IEEE, 1999.

[39] V. Boor, M. H. Overmars, and A. F. V. D. Stappen, "The Gaussian Sampling Strategy for Probabilistic Roadmap Planners," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 2, pp. 1018–1023, IEEE, 1999.

[40] A. A. Ravankar, A. Ravankar, T. Emaru, and Y. Kobayashi, "Hpprm: hybrid potential based probabilistic roadmap algorithm for improved dynamic path planning of mobile robots," *IEEE Access*, vol. 8, pp. 221743–221766, 2020.

[41] M. Garber and M. C. Lin, "Constraint-based Motion Planning Using Voronoi Diagrams," in *Algorithmic Foundations of Robotics V*, pp. 541–558, Springer, 2004.

[42] R. Wein, J. P. V. D. Berg, and D. Halperin, "The visibility– Voronoi complex and its applications," *Computational Geometry*, vol. 36, no. 1, pp. 66–87, 2007.

[43] R. Geraerts and M. H. Overmars, "A comparative study of probabilistic roadmap planners," in *Algorithmic foundations of robotics V*, pp. 43–57, Springer, 2004.

[44] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*, pp. 396–404, Springer, 1986.

[45] e. a. J. Borenstein, Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE transactions on robotics and automation*, vol. 7, no. 3, pp. 278–288, 1991.

[46] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous robots*, vol. 13, no. 3, pp. 207–222, 2002.

[47] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*, vol. 1, pp. 256–263, IEEE, 2000.

[48] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[49] S. Li, W. Su, R. Huang, and S. Zhang, "Mobile robot navigation algorithm based on ant colony algorithm with a heuristic method," in *2020 4th International Conference on Robotics and Automation Sciences (ICRAS)*, pp. 28–33, IEEE, 2020.

[50] S. Kumar, K. K. Pandey, M. K. Muni, and D. R. Parhi, "Path planning of the mobile robot using fuzzified advanced ant colony optimization," in *Innovative Product Design and Intelligent Manufacturing Systems: Select Proceedings of ICIPDIMS 2019*, pp. 1043–1052, Springer, 2020.

[51] G. Fontana, M. Matteucci, and D. G. Sorrenti, "The rawseed proposal for representation-independent benchmarking of slam," in *Workshop on experimental methodology and benchmarking in robotics research (RSS 2008)*, 2008.

[52] O. Michel, Y. Bourquin, and J.-C. Baillie, "Robotstadium: Online humanoid robot soccer simulation competition," in *Robot Soccer World Cup*, pp. 580–590, Springer, 2008.

[53] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup: The robot world cup initiative," 1995.

[54] B. Balaguer, S. Balakirsky, S. Carpin, and A. Visser, "Evaluating maps produced by urban search and rescue robots: lessons learned from robocup," *Autonomous Robots*, vol. 27, no. 4, pp. 449–464, 2009.

[55] J. Baltes, "A benchmark suite for mobile robots," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, vol. 2, pp. 1101–1106, IEEE, 2000.

[56] D. Calisi, L. Iocchi, and D. Nardi, "A unified benchmark framework for autonomous mobile robots and vehicles motion algorithms (movema benchmarks)," in *Workshop on experimental methodology and benchmarking in robotics research (RSS 2008)*, 2008.

[57] N. Basilico and F. Amigoni, "On evaluating performance of exploration strategies for autonomous mobile robots," in *Proceedings of the performance evaluation and benchmarking for intelligent robots and systems workshop at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2008)*, 2008.

[58] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo, "Choosing good distance metrics and local planners for probabilistic roadmap methods," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 1, pp. 630–637, IEEE, 1998.

[59] X. Gao, Y. Ou, Y. Fu, Z. Li, D. Dai, L. Cui, Y. Zhou, and W. Guo, "A novel local path planning method considering both robot posture and path smoothness," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 172–178, IEEE, 2013.

[60] M. Morales, *Metrics for Sampling-Based Motion Planning.* PhD thesis, PhD thesis, Texas A&M University, 2007.

[61] Y. Gao and C.-M. Huang, "Evaluation of socially-aware robot navigation," *Frontiers in Robotics and AI*, vol. 8, p. 420, 2022.

[62] J. Holtz and J. Biswas, “Socialgym: A framework for benchmarking social robot navigation,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11246–11252, IEEE, 2022.

[63] L. Kästner, R. Carstens, H. Zeng, J. Kmiecik, T. A. Le, T. Bhuiyan, B. Meinardus, and J. Lambrecht, “Arena-rosnav 2.0: A development and benchmarking platform for robot navigation in highly dynamic environments,” *arXiv preprint arXiv:2302.10023*, 2023.

[64] F. Grzeskowiak, D. Gonon, D. Dugas, D. Paez-Granados, J. J. Chung, J. Nieto, R. Siegwart, A. Billard, M. Babel, and J. Pettré, “Crowd against the machine: A simulation-based benchmark tool to evaluate and compare robot capabilities to navigate a human crowd,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3879–3885, IEEE, 2021.

[65] S. Schaefer, L. Palmieri, L. Heuer, N. van Duijkeren, R. Dillmann, S. Koenig, and A. Kleiner, “Towards reliable benchmarking for multi-robot planning in realistic, cluttered and complex environments,”

[66] Z. Yan, L. Fabresse, J. Laval, and N. Bouraqadi, “Metrics for performance benchmarking of multi-robot exploration,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3407–3414, IEEE, 2015.

[67] I. A. Sucan, M. Moll, and L. E. Kavraki, “The open motion planning library,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.

[68] E. Plaku, K. E. Bekris, and L. E. Kavraki, “OOPS for Motion Planning: An Online, Open-source, Programming System,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3711–3716, IEEE, 2007.

[69] M. Moll, I. A. Sucan, and L. E. Kavraki, “Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization,” *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 96–102, 2015.

[70] N. R. Sturtevant, “Benchmarks for grid-based pathfinding,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 2, pp. 144–148, 2012.

[71] I. Gipson, K. Gupta, and M. Greenspan, "MPK: An Open Extensible Motion Planning Kernel," *Journal of Robotic Systems*, vol. 18, no. 8, pp. 433–443, 2001.

[72] B. Cohen, I. A. Şucan, and S. Chitta, "A generic infrastructure for benchmarking motion planners," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2012)*, pp. 589–595, IEEE, 2012.

[73] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104, no. 2, 2010.

[74] A. Azzabi and K. Nouri, "An advanced potential field method proposed for mobile robot path planning," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 11, pp. 3132–3144, 2019.

[75] S. Goss, S. Aron, J.-L. Deneubourg, and J. M. Pasteels, "Self organized shortcuts in the argentine ant," *Naturwissenschaften*, vol. 76, no. 12, pp. 579–581, 1989.

[76] I. Noreen, A. Khan, and Z. Habib, "A Comparison of RRT, RRT* and RRT*-Smart Path Planning Algorithms," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 16, no. 10, p. 20, 2016.

All Maps

Static Maps This section documents all the static maps. Each map is in the following format:

==

Map Name: W:*Map Width* H:*Map Height*
Robot Rectangle: X:[*Robot MinX*-*Robot MaxX*] — Y:[*Robot MinY*-*Robot MaxY*]
Goal Rectangle: X:[*Goal MinX*-*Goal MaxX*] — Y:[*Goal MinY*-*Goal MaxY*]

——————————————-

foreach *O* in *Obstacles*
Obstacle Rectangle: X:[*O MinX*-*O MaxX*] — Y:[*O MinY*-*O MaxY*]

==

WallOne: W:50 H:50
Robot Rectangle: X:[2-23] — Y:[2-47]
Goal Rectangle: X:[27-47] — Y:[2-47]

——————————————-

Obstacle Rectangle: X:[25-26] — Y:[0-10]

==

WallTwo: W:50 H:50
Robot Rectangle: X:[2-23] — Y:[2-47]
Goal Rectangle: X:[27-47] — Y:[2-47]

——————————————-

Obstacle Rectangle: X:[25-26] — Y:[0-25]

==

WallThree: W:50 H:50
Robot Rectangle: X:[2-23] — Y:[2-47]
Goal Rectangle: X:[27-47] — Y:[2-47]

——————————————-

Obstacle Rectangle: X:[25-26] — Y:[0-35]

==

SlitOne: W:50 H:50

Robot Rectangle: X:[2-23] — Y:[2-47]

Goal Rectangle: X:[27-47] — Y:[2-47]

—————————————-

Obstacle Rectangle: X:[25-26] — Y:[0-7]

Obstacle Rectangle: X:[25-26] — Y:[40-48]

==

SlitTwo: W:50 H:50

Robot Rectangle: X:[2-23] — Y:[2-47]

Goal Rectangle: X:[27-47] — Y:[2-47]

—————————————-

Obstacle Rectangle: X:[25-26] — Y:[0-14]

Obstacle Rectangle: X:[25-26] — Y:[33-48]

==

SlitThree: W:50 H:50

Robot Rectangle: X:[2-23] — Y:[2-47]

Goal Rectangle: X:[27-47] — Y:[2-47]

—————————————-

Obstacle Rectangle: X:[25-26] — Y:[0-21]

Obstacle Rectangle: X:[25-26] — Y:[26-48]

==

RoomOne: W:50 H:50

Robot Rectangle: X:[-] — Y:[-]

Goal Rectangle: X:[-] — Y:[-]

—————————————-

Obstacle Rectangle: X:[0-5] — Y:[16-17]

Obstacle Rectangle: X:[0-5] — Y:[32-33]

Obstacle Rectangle: X:[16-17] — Y:[0-5]

Obstacle Rectangle: X:[16-17] — Y:[12-21]

Obstacle Rectangle: X:[16-17] — Y:[28-49]

Obstacle Rectangle: X:[12-21] — Y:[16-17]

Obstacle Rectangle: X:[12-21] — Y:[32-33]

Obstacle Rectangle: X:[32-33] — Y:[0-5]

Obstacle Rectangle: X:[32-33] — Y:[12-21]

Obstacle Rectangle: X:[32-33] — Y:[28-49]

Obstacle Rectangle: X:[28-37] — Y:[16-17]

Obstacle Rectangle: X:[28-37] — Y:[32-33]

Obstacle Rectangle: X:[45-49] — Y:[16-17]

Obstacle Rectangle: X:[45-49] — Y:[32-33]

==

RoomTwo: W:100 H:100
Robot Rectangle: X:[-] — Y:[-]
Goal Rectangle: X:[-] — Y:[-]

——————————————-

Obstacle Rectangle: X:[0-8] — Y:[20-21]
Obstacle Rectangle: X:[0-8] — Y:[40-41]
Obstacle Rectangle: X:[0-21] — Y:[60-61]
Obstacle Rectangle: X:[15-26] — Y:[20-21]
Obstacle Rectangle: X:[15-26] — Y:[40-41]
Obstacle Rectangle: X:[15-26] — Y:[80-81]
Obstacle Rectangle: X:[20-21] — Y:[0-21]
Obstacle Rectangle: X:[20-21] — Y:[35-86]
Obstacle Rectangle: X:[40-41] — Y:[0-21]
Obstacle Rectangle: X:[40-41] — Y:[35-81]
Obstacle Rectangle: X:[35-46] — Y:[20-21]
Obstacle Rectangle: X:[35-46] — Y:[40-41]
Obstacle Rectangle: X:[40-51] — Y:[80-81]
Obstacle Rectangle: X:[40-61] — Y:[60-61]
Obstacle Rectangle: X:[60-61] — Y:[0-21]
Obstacle Rectangle: X:[60-61] — Y:[60-98]
Obstacle Rectangle: X:[55-66] — Y:[20-21]
Obstacle Rectangle: X:[60-81] — Y:[40-41]
Obstacle Rectangle: X:[60-81] — Y:[80-81]
Obstacle Rectangle: X:[80-81] — Y:[0-21]
Obstacle Rectangle: X:[80-98] — Y:[60-61]
==
RoomThree: W:200 H:200
Robot Rectangle: X:[-] — Y:[-]
Goal Rectangle: X:[-] — Y:[-]

——————————————-

Obstacle Rectangle: X:[20-21] — Y:[40-81]
Obstacle Rectangle: X:[20-21] — Y:[100-121]
Obstacle Rectangle: X:[20-21] — Y:[140-161]
Obstacle Rectangle: X:[40-41] — Y:[20-61]
Obstacle Rectangle: X:[40-41] — Y:[100-121]
Obstacle Rectangle: X:[60-61] — Y:[40-61]
Obstacle Rectangle: X:[60-61] — Y:[120-141]
Obstacle Rectangle: X:[80-81] — Y:[0-81]
Obstacle Rectangle: X:[80-81] — Y:[140-199]

Obstacle Rectangle: X:[100-101] — Y:[20-61]

Obstacle Rectangle: X:[120-121] — Y:[120-161]

Obstacle Rectangle: X:[140-141] — Y:[0-21]

Obstacle Rectangle: X:[140-141] — Y:[60-121]

Obstacle Rectangle: X:[160-161] — Y:[120-181]

Obstacle Rectangle: X:[20-41] — Y:[20-21]

Obstacle Rectangle: X:[100-121] — Y:[20-21]

Obstacle Rectangle: X:[120-181] — Y:[40-41]

Obstacle Rectangle: X:[40-61] — Y:[60-61]

Obstacle Rectangle: X:[60-81] — Y:[80-81]

Obstacle Rectangle: X:[100-141] — Y:[100-101]

Obstacle Rectangle: X:[20-41] — Y:[120-121]

Obstacle Rectangle: X:[140-161] — Y:[120-121]

Obstacle Rectangle: X:[60-81] — Y:[140-141]

Obstacle Rectangle: X:[20-61] — Y:[160-161]

Obstacle Rectangle: X:[100-141] — Y:[160-161]

Obstacle Rectangle: X:[40-61] — Y:[180-181]

==

PlankPileOne: W:50 H:50

Robot Rectangle: X:[10-41] — Y:[2-7]

Goal Rectangle: X:[10-41] — Y:[43-47]

——————————————-

Obstacle Rectangle: X:[10-21] — Y:[16-17]

Obstacle Rectangle: X:[30-41] — Y:[16-17]

Obstacle Rectangle: X:[10-41] — Y:[32-33]

==

PlankPileTwo: W:100 H:100

Robot Rectangle: X:[15-86] — Y:[2-20]

Goal Rectangle: X:[15-86] — Y:[80-97]

——————————————-

Obstacle Rectangle: X:[15-41] — Y:[25-26]

Obstacle Rectangle: X:[60-86] — Y:[25-26]

Obstacle Rectangle: X:[15-86] — Y:[50-51]

Obstacle Rectangle: X:[15-41] — Y:[75-76]

Obstacle Rectangle: X:[60-86] — Y:[75-76]

==

PlankPileThree: W:200 H:200

Robot Rectangle: X:[5-195] — Y:[5-15]

Goal Rectangle: X:[5-195] — Y:[185-195]

————————————————-

Obstacle Rectangle: X:[28-56] — Y:[30-31]

Obstacle Rectangle: X:[84-112] — Y:[30-31]

Obstacle Rectangle: X:[140-168] — Y:[30-31]

Obstacle Rectangle: X:[0-20] — Y:[60-61]

Obstacle Rectangle: X:[40-80] — Y:[60-61]

Obstacle Rectangle: X:[120-160] — Y:[60-61]

Obstacle Rectangle: X:[180-198] — Y:[60-61]

Obstacle Rectangle: X:[22-44] — Y:[90-91]

Obstacle Rectangle: X:[66-88] — Y:[90-91]

Obstacle Rectangle: X:[110-132] — Y:[90-91]

Obstacle Rectangle: X:[154-176] — Y:[90-91]

Obstacle Rectangle: X:[0-30] — Y:[120-121]

Obstacle Rectangle: X:[50-150] — Y:[120-121]

Obstacle Rectangle: X:[170-198] — Y:[120-121]

Obstacle Rectangle: X:[22-44] — Y:[150-151]

Obstacle Rectangle: X:[66-88] — Y:[150-151]

Obstacle Rectangle: X:[110-132] — Y:[150-151]

Obstacle Rectangle: X:[154-176] — Y:[150-151]

Obstacle Rectangle: X:[0-30] — Y:[180-181]

Obstacle Rectangle: X:[50-150] — Y:[180-181]

Obstacle Rectangle: X:[170-198] — Y:[180-181]

===

CorridorOne: W:50 H:50

Robot Rectangle: X:[3-8] — Y:[5-45]

Goal Rectangle: X:[43-48] — Y:[5-45]

————————————————-

Obstacle Rectangle: X:[13-37] — Y:[0-21]

Obstacle Rectangle: X:[13-37] — Y:[30-49]

===

CorridorTwo: W:50 H:50

Robot Rectangle: X:[3-8] — Y:[5-45]

Goal Rectangle: X:[43-48] — Y:[5-45]

————————————————-

Obstacle Rectangle: X:[13-23] — Y:[0-21]

Obstacle Rectangle: X:[23-37] — Y:[0-11]

Obstacle Rectangle: X:[13-30] — Y:[30-49]

Obstacle Rectangle: X:[30-37] — Y:[18-49]

===

CorridorThree: W:50 H:50
Robot Rectangle: X:[3-8] — Y:[5-45]
Goal Rectangle: X:[43-48] — Y:[5-45]
——————————————-
Obstacle Rectangle: X:[13-37] — Y:[0-7]
Obstacle Rectangle: X:[13-18] — Y:[0-35]
Obstacle Rectangle: X:[13-28] — Y:[29-37]
Obstacle Rectangle: X:[23-37] — Y:[14-22]
Obstacle Rectangle: X:[32-37] — Y:[15-49]
Obstacle Rectangle: X:[13-37] — Y:[43-49]
===
BugTrapOne: W:50 H:50
Robot Rectangle: X:[10-41] — Y:[3-7]
Goal Rectangle: X:[13-38] — Y:[18-25]
——————————————-
Obstacle Rectangle: X:[10-41] — Y:[15-16]
Obstacle Rectangle: X:[10-11] — Y:[15-36]
Obstacle Rectangle: X:[40-41] — Y:[15-36]
===
BugTrapTwo: W:50 H:50
Robot Rectangle: X:[10-41] — Y:[3-7]
Goal Rectangle: X:[13-38] — Y:[18-25]
——————————————-
Obstacle Rectangle: X:[10-41] — Y:[15-16]
Obstacle Rectangle: X:[10-11] — Y:[15-36]
Obstacle Rectangle: X:[40-41] — Y:[15-36]
Obstacle Rectangle: X:[10-16] — Y:[35-36]
Obstacle Rectangle: X:[35-41] — Y:[35-36]
===
BugTrapThree: W:50 H:50
Robot Rectangle: X:[10-41] — Y:[3-7]
Goal Rectangle: X:[13-38] — Y:[18-25]
——————————————-
Obstacle Rectangle: X:[10-41] — Y:[15-16]
Obstacle Rectangle: X:[10-11] — Y:[15-36]
Obstacle Rectangle: X:[40-41] — Y:[15-36]
Obstacle Rectangle: X:[10-23] — Y:[35-36]
Obstacle Rectangle: X:[28-41] — Y:[35-36]

All Experimental Results

This appendix contains all the results from the experiment conducted in Chapter 5. The results are grouped by different map types where:

- Approach: the name of the navigational approach
- Overall: the overall achieved score for this approach on the specific map type
- Success: the rate at which the robot successfully reaches the goal
- Path Sc: the path length score where the shorter the distance results in a higher score
- Smooth Sc: the path smoothness score where straighter paths with fewer turns obtain a higher score
- Time Sc: the duration it took the robot to achieve the goal location
- Vis Sc: the clearance rate from obstacles where robots that are safely navigating further from the obstacle

All values are normalized to be between 0 and 1 where the optimal score is 1.

WallOne	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.861	1.00	0.90	0.77	0.90	0.72
AStar	0.861	1.00	0.90	0.78	0.90	0.72
RRT	0.857	1.00	0.89	0.77	0.90	0.71
RRTExtended	0.858	1.00	0.90	0.77	0.90	0.71
PotentialField	0.802	0.93	0.90	0.77	0.90	0.73
PheromoneField	0.862	1.00	0.90	0.78	0.90	0.72

Table 1: This table shows the scores for the map type: 'WallOne' in the range of [0.0-1.0]

WallTwo	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.855	1.00	0.90	0.83	0.90	0.61
AStar	0.851	1.00	0.89	0.83	0.90	0.60
RRT	0.845	1.00	0.87	0.83	0.90	0.62
RRTExtended	0.849	1.00	0.88	0.83	0.90	0.62
PotentialField	0.503	0.59	0.90	0.77	0.90	0.65
PheromoneField	0.838	0.99	0.87	0.84	0.90	0.62

Table 2: This table shows the scores for the map type: 'WallTwo' in the range of [0.0-1.0]

WallThree	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.851	1.00	0.90	0.88	0.90	0.53
AStar	0.845	1.00	0.89	0.88	0.90	0.52
RRT	0.830	1.00	0.85	0.84	0.89	0.56
RRTExtended	0.838	1.00	0.87	0.84	0.90	0.55
PotentialField	0.234	0.28	0.90	0.72	0.90	0.57
PheromoneField	0.700	0.85	0.84	0.87	0.89	0.54

Table 3: This table shows the scores for the map type: 'WallThree' in the range of [0.0-1.0]

SlitOne	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.866	1.00	0.90	0.83	0.90	0.70
AStar	0.865	1.00	0.90	0.83	0.90	0.69
RRT	0.860	1.00	0.89	0.81	0.90	0.69
RRTExtended	0.863	1.00	0.90	0.83	0.90	0.69
PotentialField	0.801	0.92	0.90	0.85	0.90	0.71
PheromoneField	0.867	1.00	0.90	0.84	0.90	0.70

Table 4: This table shows the scores for the map type: 'SlitOne' in the range of [0.0-1.0]

SlitTwo	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.857	1.00	0.90	0.82	0.90	0.64
AStar	0.853	1.00	0.90	0.81	0.90	0.63
RRT	0.846	1.00	0.88	0.80	0.90	0.64
RRTExtended	0.851	1.00	0.89	0.81	0.90	0.64
PotentialField	0.637	0.74	0.90	0.82	0.90	0.67
PheromoneField	0.846	0.99	0.89	0.84	0.90	0.65

Table 5: This table shows the scores for the map type: 'SlitTwo' in the range of [0.0-1.0]

SlitThree	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.850	1.00	0.90	0.86	0.90	0.54
AStar	0.843	1.00	0.89	0.85	0.90	0.53
RRT	0.828	1.00	0.85	0.84	0.89	0.55
RRTExtended	0.834	1.00	0.88	0.84	0.88	0.54
PotentialField	0.356	0.42	0.90	0.82	0.90	0.56
PheromoneField	0.807	0.98	0.84	0.87	0.89	0.54

Table 6: This table shows the scores for the map type: 'SlitThree' in the range of [0.0-1.0]

RoomOne	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.838	1.00	0.90	0.95	0.90	0.36
AStar	0.829	1.00	0.89	0.94	0.90	0.34
RRT	0.821	1.00	0.86	0.94	0.89	0.37
RRTExtended	0.825	1.00	0.88	0.94	0.89	0.37
PotentialField	0.326	0.39	0.90	0.93	0.90	0.35
PheromoneField	0.785	0.96	0.86	0.92	0.89	0.36

Table 7: This table shows the scores for the map type: 'RoomOne' in the range of [0.0-1.0]

RoomTwo	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.859	1.00	0.90	0.93	0.90	0.54
AStar	0.847	1.00	0.88	0.93	0.90	0.50
RRT	0.825	1.00	0.83	0.91	0.88	0.56
RRTExtended	0.831	1.00	0.86	0.91	0.87	0.55
PotentialField	0.222	0.26	0.90	0.88	0.90	0.55
PheromoneField	0.665	0.81	0.83	0.88	0.89	0.53

Table 8: This table shows the scores for the map type: 'RoomTwo' in the range of [0.0-1.0]

RoomThree	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.884	1.00	0.90	0.96	0.90	0.71
AStar	0.866	1.00	0.88	0.96	0.90	0.64
RRT	0.822	1.00	0.82	0.91	0.82	0.74
RRTExtended	0.807	1.00	0.86	0.91	0.75	0.70
PotentialField	0.151	0.17	0.90	0.94	0.90	0.75
PheromoneField	0.416	0.49	0.82	0.94	0.89	0.73

Table 9: This table shows the scores for the map type: 'RoomThree' in the range of [0.0-1.0]

PlankPileOne	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.847	1.00	0.90	0.93	0.90	0.45
AStar	0.840	1.00	0.88	0.94	0.90	0.44
RRT	0.821	1.00	0.85	0.88	0.89	0.46
RRTExtended	0.829	1.00	0.87	0.88	0.89	0.46
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.814	0.99	0.87	0.83	0.90	0.46

Table 10: This table shows the scores for the map type: 'PlankPileOne' in the range of [0.0-1.0]

PlankPileTwo	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.873	1.00	0.90	0.95	0.90	0.63
AStar	0.861	1.00	0.88	0.97	0.90	0.59
RRT	0.837	1.00	0.83	0.90	0.89	0.65
RRTExtended	0.848	1.00	0.86	0.91	0.89	0.64
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.787	0.95	0.82	0.85	0.89	0.65

Table 11: This table shows the scores for the map type: 'PlankPileTwo' in the range of [0.0-1.0]

PlankPileThree	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.895	1.00	0.90	0.97	0.90	0.78
AStar	0.876	1.00	0.88	0.97	0.90	0.70
RRT	0.817	0.98	0.83	0.90	0.84	0.78
RRTExtended	0.837	1.00	0.87	0.92	0.81	0.74
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.759	0.88	0.85	0.90	0.90	0.76

Table 12: This table shows the scores for the map type: 'PlankPileThree' in the range of [0.0-1.0]

CorridorOne	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.823	1.00	0.90	0.85	0.90	0.33
AStar	0.819	1.00	0.90	0.84	0.90	0.33
RRT	0.809	1.00	0.86	0.84	0.89	0.36
RRTExtended	0.812	1.00	0.89	0.80	0.89	0.35
PotentialField	0.276	0.34	0.90	0.74	0.90	0.36
PheromoneField	0.702	0.87	0.88	0.77	0.90	0.35

Table 13: This table shows the scores for the map type: 'CorridorOne' in the range of [0.0-1.0]

CorridorTwo	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.829	1.00	0.90	0.94	0.90	0.30
AStar	0.819	1.00	0.89	0.91	0.90	0.28
RRT	0.801	1.00	0.85	0.86	0.89	0.32
RRTExtended	0.810	1.00	0.88	0.87	0.89	0.31
PotentialField	0.081	0.10	0.89	0.76	0.90	0.31
PheromoneField	0.708	0.89	0.86	0.78	0.89	0.31

Table 14: This table shows the scores for the map type: 'CorridorTwo' in the range of [0.0-1.0]

CorridorThree	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.822	1.00	0.90	0.95	0.90	0.22
AStar	0.812	1.00	0.88	0.94	0.90	0.22
RRT	0.794	1.00	0.85	0.88	0.89	0.25
RRTExtended	0.798	1.00	0.88	0.89	0.88	0.24
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.556	0.72	0.82	0.84	0.89	0.23

Table 15: This table shows the scores for the map type: 'CorridorThree' in the range of [0.0-1.0]

BugTrapOne	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.841	1.00	0.90	0.96	0.90	0.37
AStar	0.832	1.00	0.88	0.96	0.90	0.35
RRT	0.816	1.00	0.84	0.93	0.89	0.40
RRTExtended	0.825	1.00	0.87	0.93	0.89	0.39
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.822	1.00	0.87	0.88	0.90	0.40

Table 16: This table shows the scores for the map type: 'BugTrapOne' in the range of [0.0-1.0]

BugTrapTwo	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.841	1.00	0.90	0.97	0.90	0.36
AStar	0.831	1.00	0.88	0.96	0.90	0.35
RRT	0.817	1.00	0.85	0.92	0.89	0.40
RRTExtended	0.824	1.00	0.87	0.93	0.89	0.39
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.822	1.00	0.87	0.88	0.90	0.40

Table 17: This table shows the scores for the map type: 'BugTrapTwo' in the range of [0.0-1.0]

BugTrapThree	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.838	1.00	0.90	0.97	0.90	0.33
AStar	0.829	1.00	0.88	0.96	0.90	0.33
RRT	0.812	1.00	0.84	0.92	0.89	0.38
RRTExtended	0.821	1.00	0.87	0.91	0.89	0.37
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.820	1.00	0.87	0.87	0.90	0.38

Table 18: This table shows the scores for the map type: 'BugTrapThree' in the range of [0.0-1.0]

ObstacleOne	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.876	1.00	0.90	0.89	0.90	0.72
AStar	0.869	1.00	0.90	0.87	0.90	0.70
RRT	0.808	0.96	0.81	0.87	0.89	0.75
RRTExtended	0.824	0.97	0.87	0.80	0.88	0.74
PotentialField	0.036	0.04	0.90	0.95	0.90	0.84
PheromoneField	0.807	0.93	0.87	0.92	0.90	0.72

Table 19: This table shows the scores for the map type: 'ObstacleOne' in the range of [0.0-1.0]

ObstacleTwo	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.842	1.00	0.90	0.89	0.90	0.44
AStar	0.829	1.00	0.89	0.90	0.89	0.40
RRT	0.285	0.36	0.77	0.87	0.87	0.51
RRTExtended	0.446	0.60	0.84	0.85	0.69	0.49
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.347	0.43	0.82	0.87	0.89	0.45

Table 20: This table shows the scores for the map type: 'ObstacleTwo' in the range of [0.0-1.0]

ObstacleThree	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.823	1.00	0.90	0.92	0.90	0.27
AStar	0.792	1.00	0.86	0.93	0.87	0.22
RRT	0.007	0.01	0.71	0.88	0.84	0.38
RRTExtended	0.000	0.00	0.00	0.00	0.00	0.00
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.016	0.02	0.86	0.85	0.90	0.33

Table 21: This table shows the scores for the map type: 'ObstacleThree' in the range of [0.0-1.0]

ObstacleFour	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.815	1.00	0.90	0.93	0.90	0.19
AStar	0.622	0.81	0.84	0.94	0.85	0.16
RRT	0.000	0.00	0.00	0.00	0.00	0.00
RRTExtended	0.000	0.00	0.00	0.00	0.00	0.00
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.000	0.00	0.00	0.00	0.00	0.00

Table 22: This table shows the scores for the map type: 'ObstacleFour' in the range of [0.0-1.0]

ObstacleFive	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.811	1.00	0.90	0.93	0.90	0.16
AStar	0.712	0.93	0.84	0.94	0.84	0.14
RRT	0.000	0.00	0.00	0.00	0.00	0.00
RRTExtended	0.000	0.00	0.00	0.00	0.00	0.00
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.000	0.00	0.00	0.00	0.00	0.00

Table 23: This table shows the scores for the map type: 'ObstacleFive' in the range of [0.0-1.0]

TunnelOne	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.843	1.00	0.90	0.91	0.90	0.44
AStar	0.830	1.00	0.89	0.90	0.89	0.40
RRT	0.684	0.85	0.81	0.89	0.89	0.46
RRTExtended	0.734	0.97	0.86	0.86	0.72	0.46
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.694	0.85	0.87	0.79	0.90	0.45

Table 24: This table shows the scores for the map type: 'TunnelOne' in the range of [0.0-1.0]

TunnelTwo	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.833	1.00	0.90	0.92	0.90	0.35
AStar	0.819	1.00	0.88	0.93	0.89	0.31
RRT	0.524	0.67	0.79	0.89	0.88	0.36
RRTExtended	0.645	0.91	0.85	0.89	0.62	0.37
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.632	0.79	0.86	0.77	0.90	0.36

Table 25: This table shows the scores for the map type: 'TunnelTwo' in the range of [0.0-1.0]

TunnelThree	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.828	1.00	0.90	0.92	0.90	0.30
AStar	0.812	1.00	0.88	0.93	0.88	0.28
RRT	0.595	0.76	0.79	0.90	0.88	0.34
RRTExtended	0.498	0.72	0.85	0.89	0.59	0.34
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.532	0.67	0.86	0.76	0.90	0.32

Table 26: This table shows the scores for the map type: 'TunnelThree' in the range of [0.0-1.0]

TunnelFour	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.827	1.00	0.90	0.95	0.90	0.27
AStar	0.808	1.00	0.87	0.95	0.88	0.25
RRT	0.525	0.68	0.80	0.90	0.86	0.29
RRTExtended	0.382	0.57	0.86	0.89	0.53	0.29
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.424	0.54	0.86	0.75	0.89	0.28

Table 27: This table shows the scores for the map type: 'TunnelFour' in the range of [0.0-1.0]

TunnelFive	Overall	Success	Path Sc	Smooth Sc	Time Sc	Vis Sc
BaseLine	0.824	1.00	0.90	0.96	0.90	0.23
AStar	0.803	1.00	0.87	0.96	0.88	0.22
RRT	0.607	0.79	0.82	0.90	0.84	0.26
RRTExtended	0.295	0.46	0.87	0.89	0.46	0.25
PotentialField	0.000	0.00	0.00	0.00	0.00	0.00
PheromoneField	0.423	0.54	0.86	0.74	0.89	0.25

Table 28: This table shows the scores for the map type: 'TunnelFive' in the range of [0.0-1.0]