A Comparative Analysis of A* and Theta* Search for Non-Holonomic Movement Path Planning

Calhoun, Tyler
College of Computing and SWEG
Kennesaw State University
Kennesaw, USA
tcalho@students.kennesaw.edu

Shears, Quann
College of Computing and SWEG
Kennesaw State University
Kennesaw, USA
qshears@students.kennessaw.edu

Lois, Diana
College of Computing and SWEG
Kennesaw State University
Kennesaw, USA
dlois@students.kennesaw.edu

Uppalapti, Naveena
College of Computing and SWEG
Kennesaw State University
Kennesaw, USA
nuppalap@students.kennesaw.edu

***Abstract*—In this paper a comparative analysis of A* Search Theta* search, and Theta* Search with a smoothing function is completed. Each algorithms suitability for non-holonomic movement path planning is discussed. Test results showed while Theta* produced shorter paths than A*, its associated computational costs make it impractical for large maps.**

I. Introduction

A* Search and Theta* Search are popular path finding algorithms which return the shortest path between a source and a destination. These algorithms are implemented to determine their suitability for application in non-holonomic movement path planning. Further, Dubin's minimal length path is applied to Theta* paths to ensure non-holonomic movement is preserved.

A* Search is first described in [2]. It relies on an informed search in which a cost heuristic determines which nodes are explored first. In this way, A* Search minimizes the number of nodes which are explored and therefore reduces the computation required to produce the shortest path. We apply this algorithm to a two-dimensional grid space to determine a feasible path between a source and a destination.

Theta* Search is a variant of A* Search that does not constrain the produced path to the grid space. That is, while every node in a path produced by A* Search must be adjacent by some measure, a path produced by Theta* may consist of nodes which are not directly adjacent. By relaxing this criteria, Theta* Search can produce shorter paths. This method does however introduce computational costs as the algorithm must do work to determine that the produced path does not intersect obstacles in the grid space.

The path produced by Theta* Search is somewhat realistic to the movement capabilities of a robot which makes the algorithm a promising candidate for application in non-holonomic path planning. To ensure the Theta* path is traversable for a robot incapable of holonomic movement further processing must be done. Dubin's Shortest Path [3] is applied to the Theta* path to constrain movements based on steering angle, turn radius, and vehicle heading.

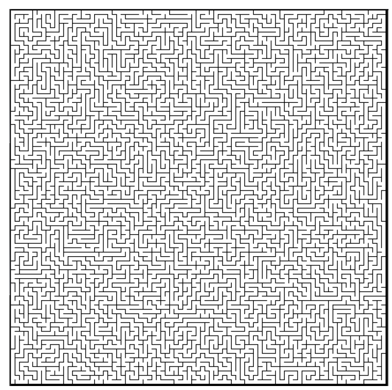

Fig. 1. A Map Generated with Recursive Backtracking

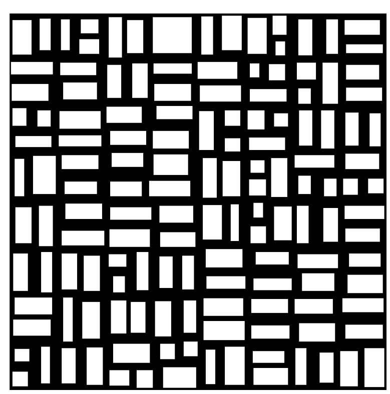

Fig. 2. A Map Generated with Spatial Partitioning

A map generation tool is implemented to allow for the creation of random and complex maps. The creation of the maps takes place in two parts. First a fully connected maze

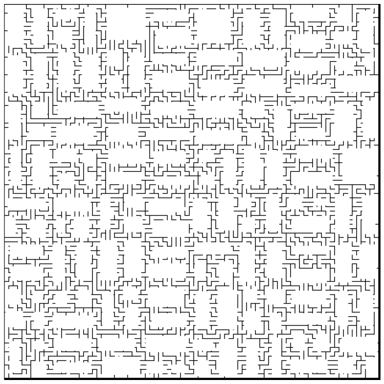

Fig. 3. A Map Generated With Both Techniques

is generated using a recursive backtracking technique. In this algorithm, a grid is initialized with no traversable spaces. A starting grid unit is given and is labeled as explored and made traversable. A random adjacent grid unit is selected if it has not already been explored. Note that two grid units are considered adjacent if they are separated by one grid unit. The adjacent coordinate is made traversable and labeled explored. The coordinate between the current grid unit and the previous grid unit is made traversable. This traversal is carried on recursively until all valid grid units have been explored. The resulting grid is fully connected, meaning any traversable grid unit can be reached by any other traversable grid unit. Next a spatial partition scheme is used to remove areas of the grid. The grid space is recursively randomly divided along either the x-axis or the y-axis. The stopping criteria is determined by the area of the rectangular subsection of the grid. Once each subsection meets the stop criteria, all grid units within each subsection are made traversable. The resulting grid consists of several open spaces connected by narrow corridors.

The motivation for this research is based on the shortcomings found in previous studies on path planning for robots, particularly with the Theta* algorithm. The earlier studies used minimal simple maps, which were not enough to fully understand the algorithm's effectiveness in different and complex situations. The first problem arises from the fact that the Theta* algorithm requires a lot of computing power to find the shortest path, which in turn makes it less suitable for large and complicated maps. The second problem was the high-order smoothing methods that were implemented to help refine the paths essentially added more complexity and are not practical for large, dynamic environments where conditions change quickly. In addition to the high-order smoothing complications, creating Voronoi diagrams which are used to help avoid obstacles, can be very resource intensive. These challenges point to the need for more efficient and practical approaches to path planning for robots. This research aims to find better methods for non-holonomic planning that use less computing power, require fewer resources, and are effective in a wide range of real-world scenarios.

II. Methodology

To determine the search algorithms suitability to real world robotics completion time and path length are determined. These metrics are used as a basis for comparison as in many applications the shortest path length is most desired and completion time should not be prohibitively high.

In order to test the search algorithms, 50 two-dimensional grids are generated using the map generation tool. Each has a width and height of 103 units. A valid source and destination are determined for each grid. Next A* Search and Theta* search are performed on each grid to obtain a path from the source to the destination. Completion time is determined by the amount of time in seconds the algorithm runs until it produces a path. The path length for each path is determined by the sum of the Euclidean distance between each node in the path.

Dubin's shortest path is then applied to the Theta* path. Each node in the path is first converted to a waypoint, which has a position, heading, and minimum turn radius. The position of the waypoint is the coordinate of the node in the grid. A waypoints heading is determined using the vector difference between the previous waypoint's position and the next waypoint's position. Each waypoint is given a minimum turn radius of two units. The length of the path is calculated as the sum of the shortest Dubin's path between each connected waypoint. Similarly, completion time is determined for the algorithm.

III. Results

The performance of both algorithms was evaluated using two metrics: completion time and path length. Completion time approximates the computational efficiency of the algorithms, while the path length assesses the effectiveness of the path finding in terms of the distance covered. The implementation also includes a function to smooth the path generated by the Theta* algorithm, using Dubin's car model to ensure the path is feasible for non-holonomic robots. This smoothing process is crucial for practical applications where smooth and continuous paths are required.

A. Completion Time

Fig. 5. shows the average completion time for A*, Theta*, and Theta* with the smoothing function on 50 103 by 103 unit grids. On average, A* took 0.018 seconds to produce a suitable path. This is markedly faster than the completion time for Theta* which took on average 4.18 seconds. Further disparities in completion time were seen on larger maps. Both algorithms were further tested on a map with a width and height of 613 units. In this single case A* returned a path in 0.64 seconds while Theta* returned a path in 422.10 seconds. Theta*'s scalability is severely limiting and should disqualify its use in large scale environments. Using the smoothing function to constrain the path to non-holonomic movements did not noteably increase the completetion time.

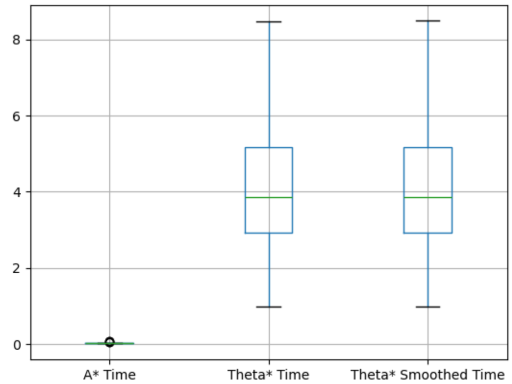

Fig. 4. Completion Time

A*	Theta*	Theta* Smoothed
0.02	4.18	4.18

Fig. 5. Average Completion Time in Seconds

B. Path Length

Fig. 7. shows the average path length for A*, Theta* and Theta* with the smoothing function on 50 103 by 103 unit grids. These results show A* produced slightly longer paths than Theta*. When the smoothing function is applied to the Theta*, the path length increases which is expected. While Theta* does successfully minimize path length its prohibitive completion time makes it ill-suited to real world applications where larger maps are required to represent the environment.

IV. Future Works

While it is shown that A* can handily beat Theta* in terms of completion time, the characteristics of its resulting path may not be viable for non-holonomic movement. In the future, we

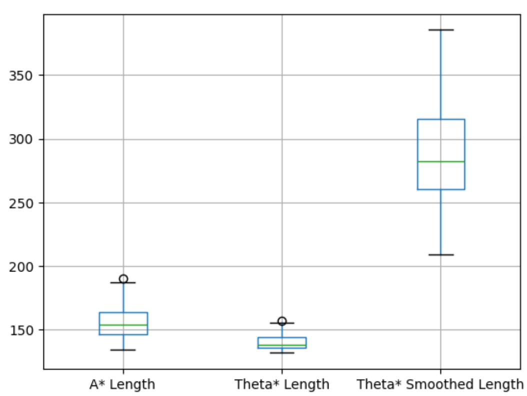

Fig. 6. Path Length

A*	Theta*	Theta* Smoothed
155.3	140.3	287.5

Fig. 7. Average Path Length

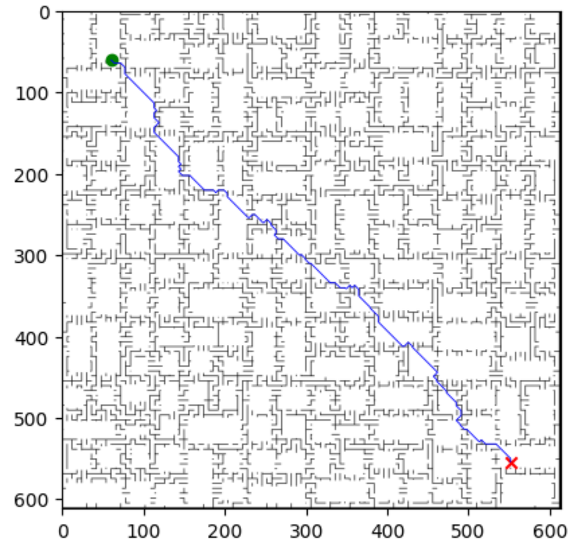

Fig. 8. A* Resulting Path

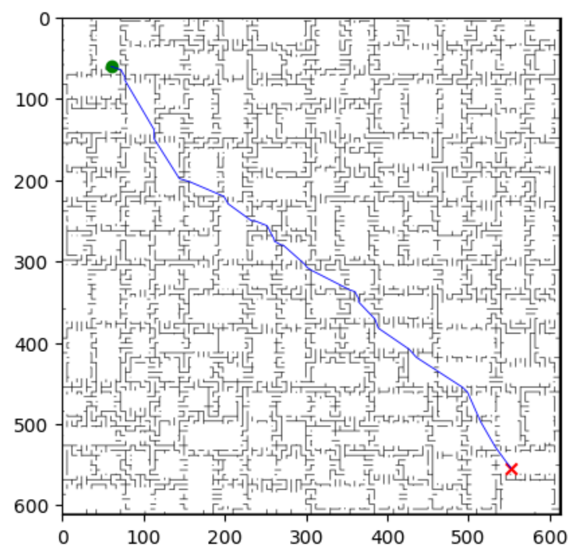

Fig. 9. Theta* Resulting Path

will explore post processing techniques which will transform the A* resulting path into a path that is feasible for non-holonomic movement.

V. Conclusion

This study conducted a thorough comparison of the A* and Theta* algorithms for robotic path planning, focusing on their performance in complex map scenarios and the impact of path smoothing for non-holonomic robots. A* is computationally efficient but limited by its angle constraints, resulting in longer paths. Theta*, on the other hand, allows for any- angle path planning, producing shorter and more flexible paths but at a significantly higher computational cost. By generating and testing on complex maps, we observed that although Theta* showed minor improvements in path length, completion times were extremely high. Incorporating Dubin's path smoothing to Theta* paths made them feasible for non-holonomic robots and did not noticeably increase completion time. Overall, Theta* may be favorable for scenarios where path length is crucial, but its computational demands limit its practicality in large-scale, dynamic environments. While A* enjoys low completetion times due to its computational efficiency, it does not produce paths suitable for non-holonomic movement. As such, smoothing functions compatable with A* should be explored. This study highlights the need for further optimization of path planning algorithms to achieve a better balance between path quality and computational efficiency, ensuring their applicability in diverse and real-world robotic applications.

References

[1] C. Van Dang, H. Ahn, D. S. Lee and S. C. Lee, "A Path Planning Method Based on Theta-star Search for Non-Holonomic Robots," 2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS), Ise, Japan, 2022, pp. 1-6, doi: 10.1109/SCISISIS55246.2022.10002141.

[2] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," in IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100-107, July 1968, doi: 10.1109/TSSC.1968.300136. keywords: Costs;Mathematical programming;Minimization methods;Functional programming;Automatic control;Minimax techniques;Gradient methods;Chemical technology;Automatic programming;Instruction sets,

[3] Dubins, L. E. "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents." American Journal of Mathematics, vol. 79, no. 3, 1957, pp. 497–516. JSTOR, https://doi.org/10.2307/2372560. Accessed 27 July 2024.