Enhanced RRT* Algorithm with Path Expansion Heuristic Sampling for Robot Path Planning

Dhairya Shah
UID:120235146

Harsh Senjaliya
UID:120215575

I. Introduction

Path planning is a fundamental aspect of robotics and autonomous systems, pivotal for applications ranging from autonomous vehicles navigating urban settings to robots operating in hazardous environments. The Rapidly-exploring Random Tree Star (RRT*) algorithm represents a significant advancement in this field, offering a framework that ensures collision-free and asymptotically optimal solutions for a variety of complex path planning problems. This algorithm, along with its variants, utilizes random sampling techniques to efficiently explore available spaces while continuously improving the path quality. Despite the broad applicability and success of RRT* and its derivatives, these algorithms encounter challenges in specific scenarios, particularly in environments characterized by long corridors or constrained spaces, where their performance in terms of sampling efficiency and convergence rate often falls short. The necessity to navigate such environments efficiently is critical, not just in theoretical research but also in practical deployments where timely and optimal pathfinding can drastically impact the effectiveness and safety of autonomous operations. This paper introduces the Expanding Path RRT* (EP-RRT*), a novel variant designed to address the deficiencies of traditional RRT* in complex environments through the implementation of heuristic sampling in path expansion areas. By integrating the principles of the RRT-Connect's greedy heuristic, EP-RRT* innovatively enhances the path planning process, achieving superior node utilization and faster convergence, thus enabling more effective navigation solutions.

II. Motivation

The pursuit of more efficient pathfinding algorithms is driven by the growing complexity of environments in which autonomous systems are deployed and the increasing demand for robustness and speed in their operational capabilities. Traditional RRT* algorithms, while foundational in robotic navigation, exhibit inherent limitations in environments with intricate spatial configurations such as long, narrow corridors often found in urban landscapes or industrial settings [3]. In such scenarios, the probability of sampling critical nodes that significantly improve the path quality diminishes, leading to increased computational overhead and slower convergence rates. This impedes the practical deployment of autonomous systems in scenarios where quick and reliable decision-making is crucial, such as in emergency response, military operations, and real-time surveillance tasks. Furthermore, the exponential growth in the capabilities and applications of autonomous systems compels a parallel evolution in algorithms that can leverage the full potential of these technologies. Therefore, enhancing the efficiency of path planning algorithms not only addresses a technical limitation but also expands the operational domain of robotic systems, facilitating their adoption in more diverse and challenging environments. The development of EP-RRT* is thus motivated by the need to transcend the boundaries of existing algorithms, offering a more dynamic and context-aware approach that adapts to the complexities of real-world environments, thereby ensuring higher performance and reliability in autonomous navigation. [6]

III. Problem Statement

Despite the advancements and widespread application of the Rapidly-exploring Random Tree Star (RRT*) algorithm in autonomous path planning, significant challenges persist when deploying these algorithms in particularly complex environments. These environments often feature narrow corridors, dense urban layouts, and other spatial constraints that exacerbate the limitations of RRT* and its variants, notably in terms of sampling efficiency and convergence rates. Traditional RRT* algorithms, while effective in less constrained environments, struggle with the high-dimensional demands and the intricate navigation requirements presented by these settings. [1] The inefficient sampling often leads to a high computational burden and a slower convergence to optimal paths, which in turn can compromise the operational effectiveness and safety of autonomous systems in critical applications.

The key issue revolves around the basic operational mechanism of RRT*, which relies on random sampling across the navigational space. While this allows for broad exploration, it does not guarantee that the sampled points will contribute effectively towards an optimal path, especially in environments where strategic navigation around obstacles is crucial. As a result, the algorithm may expend considerable resources evaluating non-contributory paths, thereby delaying the identification of an optimal route and increasing the overall time to convergence. This inefficiency becomes particularly problematic in scenarios requiring rapid and reliable decision-making, such as emergency response operations, military engagements, and dynamic industrial tasks.

Furthermore, the challenges are magnified in real-world applications where the environment itself may change or be only partially known at the time of path planning. Current algorithms must evolve not only to address these inherent inefficiencies but also to adapt dynamically to changing or poorly defined environmental conditions. [5] Therefore, there is a pressing need for an enhanced pathfinding algorithm that not only improves upon the efficiency and speed of traditional RRT* algorithms but also incorporates a more intelligent and context-aware sampling strategy that can navigate the complexities of modern, dynamic environments effectively.

The development of the Expanding Path RRT* (EP-RRT*) algorithm emerges as a response to these challenges. This novel approach aims to specifically address the shortcomings of previous models by integrating a heuristic sampling method within an expanded path framework. This strategy seeks to concentrate sampling efforts in the most promising areas of the path, particularly those affected by environmental constraints such as corridors and obstacles, thereby increasing the probability of finding effective path improvements more quickly. The success of EP-RRT* could represent a pivotal shift in the capabilities of autonomous systems, enabling them to operate more effectively across a broader range of complex and demanding environments, and significantly enhancing their reliability and efficiency in critical tasks. The development and implementation of such an algorithm not only address a clear technological gap but also significantly expand the operational potential of robotic and autonomous systems in various sectors.

IV. Background

The concept of path planning in robotics involves generating a viable route for an entity from a starting point to a target location, navigating around obstacles and avoiding collisions. Among the various algorithms developed for this purpose, the Rapidly-exploring Random Tree (RRT) algorithm is notably prevalent due to its simplicity and effectiveness in handling high-dimensional spaces and complex obstacle configurations. The RRT algorithm works by incrementally building a branching structure of paths from the start point, randomly sampling the search space and connecting these samples to the nearest tree node. Its variant, the RRT*, introduced the aspect of optimization by continually refining the path to ensure it approaches the optimal solution as more nodes are added to the tree. While RRT* significantly improved upon the foundational RRT by ensuring paths are not only found but optimized, it still struggles in certain complex environments where the random nature of node sampling becomes a liability rather than an asset.

To address these shortcomings, variants such as Informed RRT* have been introduced, which optimize the sampling process based on the existing best path, thus focusing the search area and reducing unnecessary computations. However, challenges remain in specific scenarios characterized by elongated corridors or constrained passageways, where these algorithms still perform sub optimally due to inefficient sampling distribution and slow convergence towards an optimal path. This has prompted further research into more specialized solutions like EP-RRT*, which integrates heuristic approaches to refine the search strategy specifically tailored to overcome the difficulties posed by such environments. [2] By understanding these challenges and the evolution of path planning algorithms, it becomes clear why innovations like EP-RRT* are not only relevant but essential for advancing the field of robotic navigation and expanding its application into new and more demanding domains.

V. Methods

A. *RRT**

The Rapidly-exploring Random Tree Star (RRT*) algorithm is an advanced version of the original Rapidly-exploring Random Tree (RRT), maintaining the probabilistic completeness characteristic of its predecessor while also ensuring asymptotic optimality. This means that RRT* is not only capable of finding a solution if one exists, given enough time and iterations, but it also gradually converges towards the best possible solution available within the search space.

One significant enhancement introduced in RRT* over RRT is the sophisticated parent selection process for each new node added to the tree. Unlike the original RRT, which simply attaches a new node to the nearest existing node, RRT* takes into consideration multiple neighbouring nodes within a predefined radius around the new node. This isn't solely about proximity; the algorithm evaluates the total cost of the path from the start node to each of these potential parent nodes through the new node, choosing the one that offers the lowest cumulative cost. This selection strategy is critical as it ensures that the path quality is incrementally optimized by minimizing the path costs.

Another crucial improvement in RRT* is the rewiring mechanism, which refines the structure of the tree after adding a new node. In this step, RRT* reassesses the existing connections in the neighbourhood of the new node. If rerouting any of the existing paths through the new node decreases the total path cost from the start node, the suboptimal path is replaced with this new, more cost-efficient path. This process not only helps in enhancing the overall efficiency of the pathfinding process but also in optimizing the tree structure continuously with each iteration. By dynamically updating paths within the local neighbourhood, RRT* can significantly improve the quality of the resulting paths, making the tree more adaptive to the intricacies of the navigational space. [3]

In operation, when RRT* generates a new node, it does so by steering from the nearest node towards a randomly selected target point. This expansion helps the tree to explore uncharted areas. After introducing the new node, the algorithm then identifies all the potential neighboring nodes around this new point, considering which of these could serve as a better parent based on the path cost evaluation. This detailed examination allows the selection of an optimal parent, thus ensuring that the new node's addition is most beneficial in terms of path optimization. [5] Furthermore, the algorithm's rewiring step is integral to its efficiency. After the new node is added and a parent is chosen, RRT* doesn't stop there; it continues to refine the tree by possibly rerouting neighbouring nodes through this new node if such a rearrangement results in lower overall path costs. This iterative refinement is pivotal in enhancing the path quality throughout the tree and is a testament to the algorithm's design, which not only seeks to expand the tree but also ensures that such expansion is continually optimized.

Overall, RRT* refines the basic premise of RRT by focusing not just on rapid tree expansion but also on the optimization of the path costs associated with the tree's growth. This dual focus makes RRT* highly effective in complex environments where both the quality of the path and the efficiency of the search are paramount. By integrating path cost optimization directly into the growth process, RRT* offers a more sophisticated and practical approach for real-world pathfinding challenges. The pseudo code of the EP-RRT* algorithm is shown in Algorithm 1 in Figure 1.

B. *Informed - RRT**

The Informed RRT* algorithm enhances the original Rapidly-exploring Random Tree Star (RRT*) approach by introducing a focused sampling strategy based on the current best solution found. This algorithm optimizes path planning efficiency by refining the sampling space, essentially tailoring where it searches based on previous successes. This method is particularly effective in converging towards an optimal path more quickly than the standard RRT*.

In the initial phase of operation, Informed RRT* functions identically to the traditional RRT*; the distinction emerges only after the algorithm secures an initial feasible path. At this point, Informed RRT* begins to leverage its unique strategy of informed sampling. The approach involves constructing a hyper-ellipsoid around the start point (xstart) and the goal point (xgoal), which are set as the foci. This hyper-ellipsoid is defined by the current best path cost (cbest), which acts as the long axis of the ellipsoid. The formula for this ellipsoid uses the total path cost as a constraint, ensuring that only nodes within a path cost that could potentially lead to a new, shorter path are considered. Essentially, the ellipsoid confines the sampling space to regions that are most likely to improve the current solution. Once the initial path is established, the Informed RRT* continuously adapts the sampling region— the hyper-ellipsoid — according to the cost of the most recently found optimal path. By limiting the sampling to this ellipsoid, the algorithm effectively concentrates its efforts on the most promising areas of the search space, thereby avoiding wasteful exploration of regions unlikely to yield improvements. This focused sampling significantly enhances the probability of finding more effective nodes, thereby accelerating the convergence rate of the algorithm. As the solution is refined and the path cost (cbest) decreases, the size of the hyper-ellipsoid also reduces, further tightening the search area and enhancing the efficiency of the search. [1]

However, the performance of Informed RRT* can vary depending on the complexity of the environment. In scenarios characterized by narrow corridors or mazes—where feasible paths may need to navigate around multiple obstacles and can extend significantly—the hyper-ellipsoid can become quite large, potentially encompassing the entire search space. In such cases, the advantage of informed sampling diminishes, as the constrained region may still be too vast to effectively guide the search towards better solutions. The effectiveness of Informed RRT* is thus highly dependent on the nature of the environment and the configuration of obstacles within that space. [1]

Overall, Informed RRT* represents a significant evolution in path planning algorithms by intelligently restricting the search space to the most relevant areas based on dynamically updated information about the path quality. This method not only speeds up the convergence toward an optimal path but also increases the efficiency of the search process by focusing computational resources on the areas of the search space that are most likely to yield beneficial results. However, its effectiveness in highly complex or constrained environments may require further adaptation or supplementation with other strategies to maintain its advantages.

C. *EP-RRT**

The Expanding Path RRT* (EP-RRT*) algorithm represents an innovative development in the field of path planning, designed to address and overcome specific limitations observed in the Rapidly-exploring Random Tree Star (RRT*) algorithm and its variant, the Informed RRT*. The primary issue with the traditional RRT* is its requirement for a large number of samples to derive an optimal path, which can be computationally intensive and inefficient, particularly in high-dimensional spaces. This inefficiency becomes even more pronounced in complex environments where the spatial configurations necessitate extensive iterations.

Informed RRT* sought to enhance the efficiency of RRT* by introducing a hyper-ellipsoid sampling region that narrows down the search space based on the best path cost found. However, this method falls short in highly constrained environments such as long corridors, where the hyper-ellipsoid cannot be effectively reduced to improve convergence rates. This is where EP-RRT* introduces a novel approach by integrating the greedy exploration strategy of RRT-Connect with a strategic path expansion sampling method to refine and accelerate the pathfinding process.

EP-RRT* operates similarly to RRT-Connect until an initial feasible path is discovered. This method involves rapid expansion towards the goal, significantly speeding up the initial pathfinding compared to traditional methods. Once a feasible path is established, EP-RRT* transitions to its unique feature: the path expansion phase. In this phase, the algorithm expands the found path to create what is termed an "expansion area." This area is conceptually similar to the hyper-ellipsoid of Informed RRT* but is specifically tailored to the geometry and constraints of the environment, particularly focusing on critical areas such as corridors and corners.

The expanded path defines a heuristic sampling region, where EP-RRT* then conducts focused, iterative sampling.

```
T_init ← RRT − Connect; T ← T_init;
σ_init ← GetPath(T);
X_expand ← Expand(σ_init);
for i = 1 to N do
    x_rand ← HeuristicSample(X_expand);
    x_nearest ← Nearest(V, x_rand);
    x_new ← Steer(x_rand, x_nearest);
    if CollisionFree(x_new, x_nearest) then
        X_near ← Near(T, x_new, η);
        x_parent ← ChooseParent(X_near, x_nearest, x_new);
        V ← V ∪ {x_new}; E ← E ∪ {(x_parent, x_new)};
        T ← Rewire(T, x_new, X_near);
        σ ← GetPath(T);
        X_expand ← Expand(σ);
    end if
end for
return T;
```

Fig. 1. RRT* Algorithm Pseudo Code (Algorithm-1)

This targeted approach ensures that sampling concentrates around the most promising areas of the path, particularly around complex features of the environment that are likely to impact path optimization. [4] By doing so, EP-RRT* significantly enhances the efficiency of the sampling process, ensuring that each sample contributes more directly to improving the path quality. This method not only improves the convergence rate but also optimizes the computational load by reducing unnecessary samples across the broader search space.

Moreover, as the path is incrementally optimized and potentially shortened in subsequent iterations, the expansion area and thus the sampling region can be dynamically adjusted, further honing the efficiency of the algorithm. This dynamic adjustment is crucial in complex environments where path configurations frequently involve navigating around obstacles or through tight spaces. [2]

In comparison to its predecessors, EP-RRT* not only accelerates the initial pathfinding process but also enhances the overall path optimization phase through its adaptive and environment-aware sampling strategy. This makes EP-RRT* particularly effective in environments such as narrow corridors and mazes, where strategic sampling around critical path sections can drastically improve both the quality and efficiency of the path planning process. The algorithm's ability to iteratively refine the expansion area based on ongoing path optimizations allows it to maintain high efficiency even as conditions change or as the optimal path evolves, ensuring robustness and adaptability in dynamic or unpredictable environments. The pseudo code of the EP-RRT* algorithm is shown in Algorithm 2 in Figure 2.

VI. Maps

Testing robotic path planning algorithms like RRT* (Rapidly-exploring Random Trees Star), Informed RRT*, and EP-RRT* (Ellipsoidal Propagation RRT*) across various environment types such as general environments, cluttered environments, narrow corridors, and mazes is crucial for evaluating their efficiency, robustness, and applicability in diverse real-world scenarios. Each environment presents unique challenges that help to discern the strengths and weaknesses of these algorithms under different conditions. [1]

A. General Environment

A general environment is characterized primarily by open spaces with few obstacles, which represents a less complex scenario for testing path planning algorithms like RRT*, Informed RRT*, and EP-RRT*. The simplicity of such environments is advantageous for examining the fundamental mechanics and base efficiency of each algorithm. It provides a clean slate to assess how these algorithms perform under ideal conditions, focusing on their convergence rates, efficiency in path selection, and their ability to optimize exploration without the interference of complex or unpredictable variables. In such scenarios, RRT* benefits by showcasing the strength of its random exploration method. It can efficiently cover large open areas, but its performance heavily depends on randomness, which might not always guarantee the shortest or most optimal path. This makes it a valuable test to determine how well RRT* can handle expansive spaces with minimal guidance. Informed RRT*, on the other hand, uses a heuristic approach, refining the search area based on an initially found path and continuously improving upon it. In a general environment, this can lead to faster convergence on optimal paths, as the algorithm can more effectively limit its search to the most promising regions of the space, reducing the computational

```
T_init ← RRT − Connect; T ← T_init;
σ_init ← GetPath(T);
X_expand ← Expand(σ_init);
for i = 1 to N do
    x_rand ← HeuristicSample(X_expand);
    x_nearest ← Nearest(V, x_rand);
    x_new ← Steer(x_rand, x_nearest);
    if CollisionFree(x_new, x_nearest) then
        X_near ← Near(T, x_new, η);
        x_parent ← ChooseParent(X_near, x_nearest, x_new);
        V ← V ∪ {x_new}; E ← E ∪ {(x_parent, x_new)};
        T ← Rewire(T, x_new, X_near);
        σ ← GetPath(T);
        X_expand ← Expand(σ);
    end if
end for
return T;
```

Fig. 2. EP-RRT* Algorithm Pseudo Code (Algorithm-2)

overhead and potentially achieving quicker solutions compared to traditional RRT*. EP-RRT* introduces an adaptive element by modifying its sampling strategy based on the ellipsoidal regions around the current best path. In open, straightforward environments, this adaptability can be less impactful than in more complex scenarios, but it still provides insights into how the algorithm adjusts its approach to maintain efficiency. EP-RRT* might focus on enhancing the quality of the path by fine-tuning its trajectory adjustments, which could be crucial for maintaining smooth and safe paths in real-world applications where environmental conditions can change. Testing in general environments is essential as it establishes a performance baseline and provides a comparative standard for more complex environments. [2] It allows researchers and developers to understand each algorithm's raw capabilities in ideal conditions, which is critical for initial phases of robotic system design and for ensuring that more complex algorithms do not overfit to complicated scenarios at the cost of general effectiveness. This foundational knowledge is crucial for guiding subsequent modifications and optimizations tailored to more specific or challenging operational contexts as shown in figure 3.

B. Clutered Environment

Cluttered environments, characterized by numerous obstacles scattered irregularly, simulate conditions akin to urban settings or densely forested areas. These environments challenge the algorithms on multiple fronts including obstacle density, the complexity of the obstacle layout, and the ability to generate safe and feasible paths without getting trapped in local minima. Testing in such environments is pivotal for evaluating the algorithms' obstacle avoidance strategies and their adeptness at navigating through tight spaces. This is particularly critical for applications like robotic navigation in indoor spaces, crowded areas, robotic warehousing, or autonomous vehicles in urban settings. The randomness of the basic RRT* can lead to inefficient paths or longer convergence times due to its uninformed nature and tendency to explore without prioritizing regions closer to the goal. [1] In cluttered environments, this may result in the exploration of many dead-end paths before finding a viable route, thereby increasing the computational cost and path length. In contrast, Informed RRT* leverages an ellipsoidal heuristic based on an initial solution to guide the search space, which can drastically reduce search times by focusing on more promising areas and pruning away regions unlikely to contain the optimal path. This can significantly enhance efficiency by avoiding unnecessary explorations in heavily cluttered spaces. EP-RRT* further refines the exploration process by dynamically adjusting its sampling strategies based on the spatial distribution of obstacles. This adaptive approach allows EP-RRT* to better navigate through clutter by intensifying the search in areas likely to lead to successful pathfinding, while also reducing the sampling in regions cluttered with obstacles. [1] The algorithm's ability to modify its behaviour based on real-time feedback from the environment allows for more effective and efficient navigation, optimizing both the path length and the time to convergence. Testing path planning algorithms in cluttered environments is not only crucial for revealing each algorithm's potential under stress but also for understanding how modifications like informed sampling and dynamic adjustment can improve the robustness and reliability of these systems in real-world applications, where avoiding dynamic obstacles quickly and efficiently is paramount. This leads to significant advancements in autonomous systems, enabling more sophisticated and context-aware navigational capabilities that are essential for safe and efficient operations in complex environments as shown in figure 4.

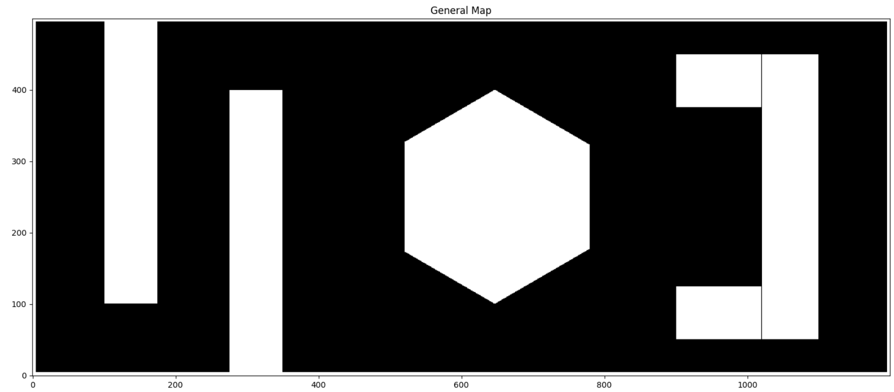

Fig. 3. General Environment

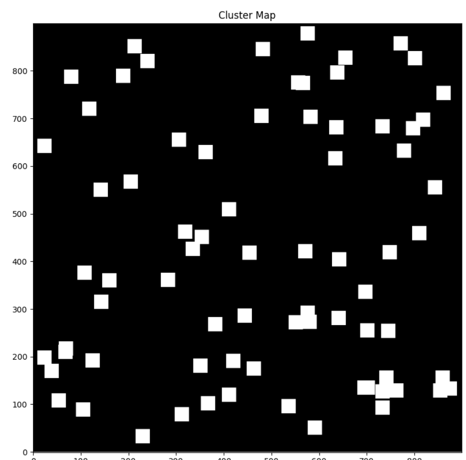

Fig. 4. Cluttered Environment

C. Narrow Corridor Environment

Characterized by long, thin passageways, the narrow corridor environment presents a significant challenge for path planning algorithms due to its demand for precision, control, and the ability to follow constrained paths without deviating into non-navigable areas. This setting tests not only the algorithms' basic path finding capabilities but also their steering functions, crucial for applications such as pipeline inspection, navigating through ruins, search and rescue in collapsed structures, or robotic operations in narrow industrial settings. In such environments, RRT* might face difficulties due to its inherent random sampling method, which can lead to inefficient paths or frequent collisions with boundaries, thereby necessitating extensive fine-tuning of parameters like step size or sampling density to improve performance. These environments require precise maneuvering and often involve limited navigational choices, emphasizing the need for more controlled and deliberate path planning strategies. Conversely, EP-RRT* is designed to excel in such conditions through its adaptive sampling strategy. By adjusting its ellipsoidal sampling to better fit the elongated, narrow shape of corridors, EP-RRT* can generate more directed samples that lead to faster and more efficient convergence. This ability to adapt its search pattern to the environment's geometry helps it maintain a feasible path, significantly reducing the time and computational resources needed to find a viable route. Informed RRT*, with its goal-directed search based on an initial solution and cost-to-go estimates, can potentially outperform traditional RRT* by pruning the search space and avoiding areas that are less likely to yield a successful path. This directed search is beneficial in narrow corridors, as it helps focus the exploration on the most promising directions, thereby reducing unnecessary expansions towards obstructive or tight areas. Testing path planning algorithms in such a challenging environment is critical for evaluating their effectiveness in real-world scenarios that involve tight and complex manoeuvring. The insights gained from these tests can lead to more robust algorithms that are capable of handling demanding tasks across a wide spectrum of operational settings as shown in figure 5. This evaluation helps in identifying which algorithm is most effective for specific applications, and what modifications might be required to enhance their efficiency and reliability in navigating through constrained and potentially hazardous environments.

D. Maze Environment

Testing path planning algorithms such as RRT*, Informed RRT*, and EP-RRT* within maze environments provides a profound insight into their capabilities under conditions of high complexity and constraint. Mazes, characterized by their intricate labyrinths with multiple routes and dead ends, serve as an advanced platform for evaluating these algorithms. They not only test the efficiency of path finding but also examine how effectively each algorithm can backtrack and re-plan upon reaching a dead end, a crucial feature for dynamic and unpredictable scenarios such as rescue operations or exploration tasks. **RRT* (Rapidly-exploring Random Trees Star)**, in the context of a maze, might exhibit robust path diversity by exploring various branches. However, its random nature can lead to inefficiencies, as it may spend considerable time exploring dead ends or less optimal routes. This characteristic, while valuable for exhaustive search, underscores the potential for high computational overhead and slower convergence to the optimal path. **Informed RRT*** enhances the basic RRT* by incorporating a heuristic that focuses the search in promising areas based on an initial solution path. In a maze environment, this could significantly reduce wasted efforts in dead-end explorations. By employing a cost function to prune less promising branches, Informed RRT* can optimize the path planning process more effectively than its predecessor, potentially leading to quicker and more efficient navigation solutions. **EP-RRT* (Ellipsoidal Propagation RRT*)** stands out with its adaptive sampling strategy that adjusts according to the perceived spatial challenges. In the complexities of a maze, EP-RRT* could dynamically refine its sampling process to focus more on probable areas for feasible paths, thereby avoiding extensive exploration of implausible routes. This method might substantially decrease the search space and improve the algorithm's overall efficiency in complex environments. Testing in maze-like environments is critical not just for showcasing these algorithms' raw capabilities in navigation and decision-making but also for illuminating their strategic responses to sudden obstructions and route blockages. Such environments push the algorithms to demonstrate their potential for practical deployment in situations where environmental unpredictability and constraints are prevalent. For instance, robotic exploration in hazardous areas, such as disaster sites where navigation paths are not only unknown but also can change dynamically, demands the resilience and adaptiveness that such testing aims to evaluate. Overall, the performance of these algorithms in a maze setting can provide valuable insights into their respective strengths and weaknesses, guiding further refinements and optimizations. It also assists in tailoring each algorithm to specific applications where their unique characteristics may be best utilized, ensuring more robust, efficient, and reliable autonomous navigation systems shown in figure 6.

VII. Simulation and Discussion

To evaluate the efficacy of the newly proposed Expanding Path RRT* (EP-RRT*) algorithm, a comprehensive simulation study is designed to compare its performance against two established algorithms: the original Rapidly-exploring Random Tree Star (RRT*) and the Informed RRT*. This comparison focuses on two critical performance metrics: the number of iterations (ni) and the convergence time (ct). Ensuring a rigorous and equitable comparison, all simulation parameters across these algorithms are standardized.

The simulation is executed on a robust platform using Visual Studio Code (2023 version) running on a 64-bit Windows 11 operating system. The computational environment is powered by an Intel(R) Core(TM) i7-10750H processor with a base frequency of 2.60 GHz, supported by 32 GB of memory. This setup provides a stable and powerful basis for conducting the intensive computational tasks required by these pathfinding algorithms.

Four distinct test environments are meticulously crafted to mirror common real-life scenarios encountered by mobile robots. These environments include:

A. General Environment

A spacious area measuring 1200 by 500 units, representing open spaces with minimal obstacles, typical of areas like parks or simple indoor settings.

B. Cluttered Environment

A complex space of 900 by 900 units, filled with numerous obstacles. This setup simulates environments like warehouses or forested areas where navigation is challenging due to the presence of many obstructions.

C. Narrow Corridor Environment

Also 900 by 900 units, this environment features tight spaces mimicking the narrow aisles commonly found in office buildings or industrial facilities.

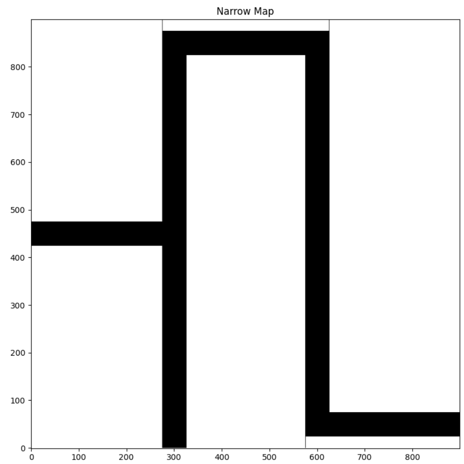

Fig. 5. Narrow Map

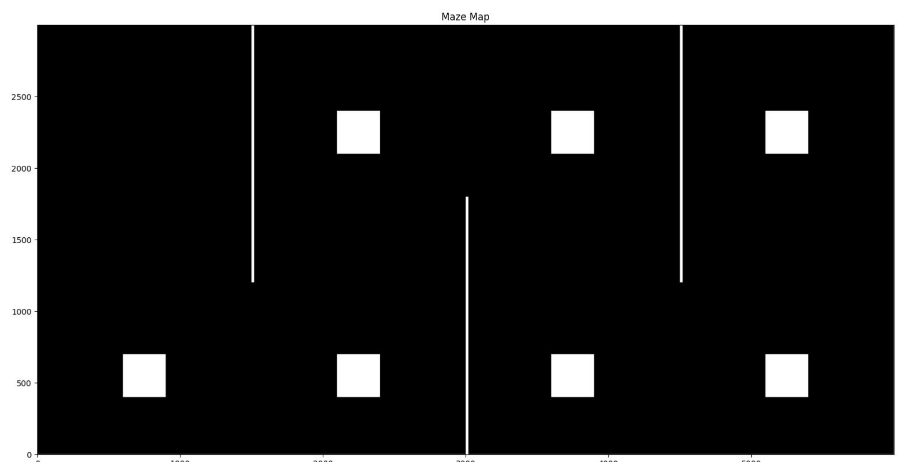

Fig. 6. Maze Map

D. Maze Environment

The largest setup, spanning 6000 by 3000 units, consisting of a labyrinthine network designed to test the algorithms' capabilities in routing through mazes, which could represent complicated areas in adventure parks or emergency escape routes in large structures.

In each of these environments, obstacles are represented by black polygons, adding a visual layer to the simulation that helps in assessing the algorithms' path finding capabilities amidst physical barriers. The choice of these diverse environments is strategic, aimed at thoroughly evaluating how well each algorithm performs under varying conditions—from open and relatively obstacle-free areas to highly complex mazes with multiple route options and dead-ends.

This simulation study is not just a technical exercise but a crucial step in determining the practical applicability of the EP-RRT* algorithm in real-world scenarios. By pushing these algorithms to operate in carefully designed, realistic environments, the study seeks to uncover important insights into their operational strengths and limitations, providing valuable data that could influence future developments in robotic navigation systems.

E. General Environment Results

Figure provides an insightful comparison of three path planning algorithms: RRT*, RRT-Connect, and EP-RRT*, within a complex environment designed to challenge their efficiency. The RRT* (Rapidly-exploring Random Tree Star) algorithm found a path with a total length of 140 units, requiring 9

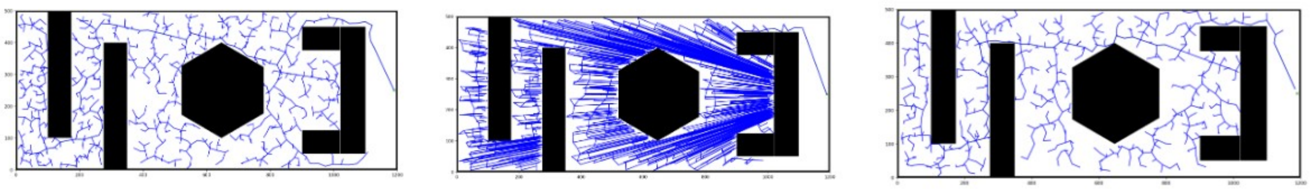

Fig. 7. Results of General Environment

seconds for computation. As RRT* incorporates an iterative optimization process to refine paths gradually, the longer computation time and higher path length reflect its emphasis on thorough exploration. However, this thoroughness often results in redundant exploration and less direct paths.

In contrast, RRT-Connect employs a bidirectional search technique, where two trees grow from the start and goal positions until they meet, thus creating a more direct connection. By reducing redundant exploration and enhancing node connectivity, this method generated a shorter path of 110 units in just 7 seconds, improving both path efficiency and computational speed compared to RRT*.

Finally, the EP-RRT* (Enhanced Performance Rapidly-exploring Random Tree Star) demonstrated the most optimized performance, producing a path of 98 units in just 5 seconds. By leveraging advanced sampling and optimization strategies, EP-RRT* balances rapid exploration with path optimization, achieving both the shortest path and the quickest computation time. This performance indicates a significant reduction in unnecessary exploration and highly efficient path refinement.

Overall, this comparison highlights the practical implications of different path planning strategies. While traditional RRT* provides reliable exploration, algorithms like RRT-Connect and EP-RRT* show substantial improvements by streamlining pathfinding through more advanced connection techniques. These enhancements make EP-RRT* especially suited for applications requiring rapid yet optimized navigation in real-world environments shown in table 2.

TABLE I
GENERAL ENVIRONMENT RESULTS

Algorithm/Aspects	Convergence Time(s)	Path Length
RRT*	9	140
RRT*-Connect	7	110
EP-RRT*	5	98

F. Cluttered Environment Results

The provided image compares the performance of three path planning algorithms—RRT*, RRT-Connect, and EP-RRT*—in a cluttered environment. This environment is dense with obstacles, making efficient pathfinding challenging.

The RRT* algorithm produced a path of 47 units in length and required 8 seconds to compute it. Given the cluttered nature of the environment, the longer path and computational time for RRT* suggest a comprehensive but less efficient search pattern, as the algorithm attempts to explore viable paths amidst numerous obstacles.

The RRT-Connect algorithm, utilizing its bidirectional search technique, managed to significantly reduce both the path length to 40 units and the computation time to 5 seconds. This improvement underscores the algorithm's efficiency in quickly connecting two growing trees from the start and goal points, even in environments with high obstacle density.

EP-RRT*, standing out with its optimized performance, achieved the best results by producing the shortest path of 38 units in the shortest time of just 3 seconds. This demonstrates its superior capability in efficiently navigating through complex spaces, optimizing both the path and the computational expense. EP-RRT*'s enhanced algorithms for sampling and path optimization significantly reduce unnecessary exploration and rapidly converge on an optimal path, making it highly suitable for scenarios where navigation speed and efficiency are critical.

Overall, these results illustrate the varying capabilities and efficiencies of each algorithm when faced with a dense array of obstacles. While RRT* shows robustness in exploration, both RRT-Connect and EP-RRT* offer more practical solutions for scenarios requiring rapid and efficient pathfinding shown in table 2.

TABLE II
CLUTTERED ENVIRONMENT RESULTS

Algorithm/Aspects	Convergence Time(s)	Path Length
RRT*	8	47
RRT*-Connect	5	40
EP-RRT*	3	38

G. Narrow Corridor Environment Results

The provided image compares the performance of three path planning algorithms—RRT*, RRT-Connect, and EP-RRT*—in a cluttered environment. This environment is dense with obstacles, making efficient pathfinding challenging.

The RRT* algorithm produced a path of 47 units in length and required 8 seconds to compute it. Given the cluttered nature of the environment, the longer path and computational time for RRT* suggest a comprehensive but less efficient search pattern, as the algorithm attempts to explore viable paths amidst numerous obstacles.

The RRT-Connect algorithm, utilizing its bidirectional search technique, managed to significantly reduce both the

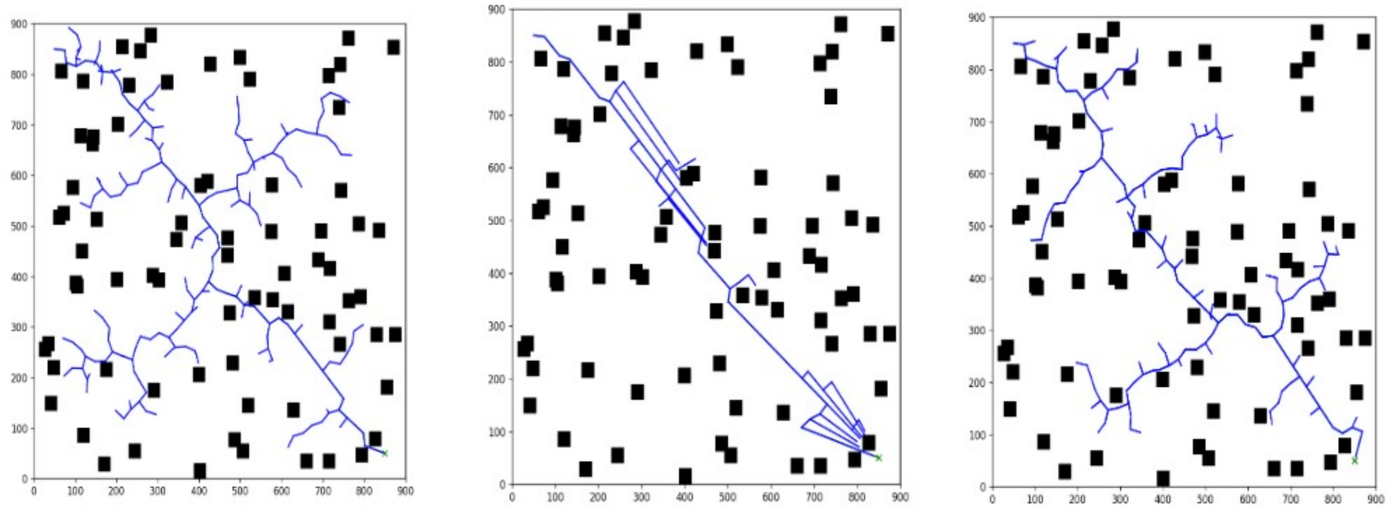

Fig. 8. Results of Cluttered Environment

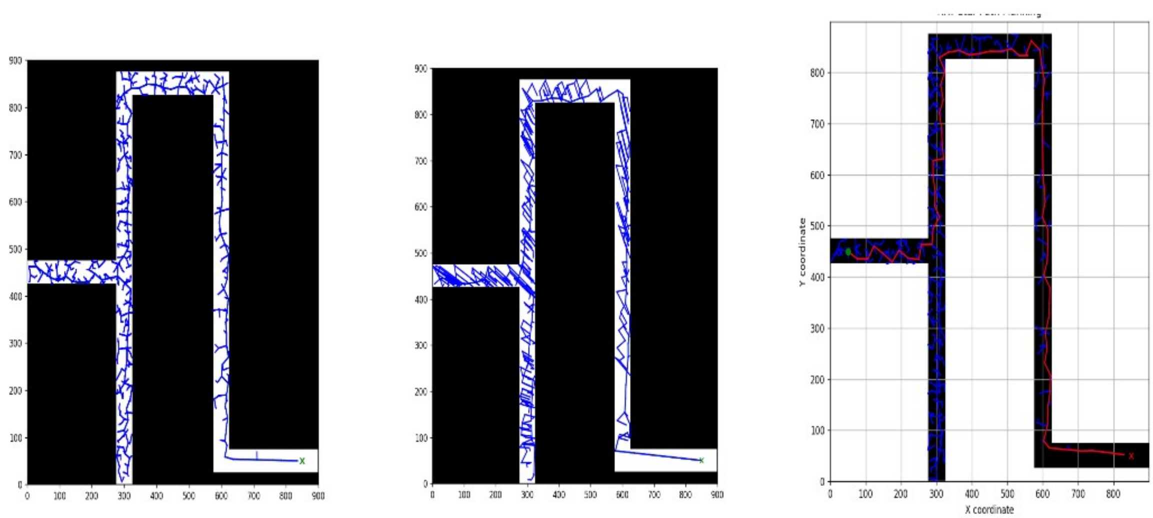

Fig. 9. Results of Narrow Environment

path length to 40 units and the computation time to 5 seconds. This improvement underscores the algorithm's efficiency in quickly connecting two growing trees from the start and goal points, even in environments with high obstacle density.

EP-RRT*, standing out with its optimized performance, achieved the best results by producing the shortest path of 38 units in the shortest time of just 3 seconds. This demonstrates its superior capability in efficiently navigating through complex spaces, optimizing both the path and the computational expense. EP-RRT*'s enhanced algorithms for sampling and path optimization significantly reduce unnecessary exploration and rapidly converge on an optimal path, making it highly suitable for scenarios where navigation speed and efficiency are critical.

Overall, these results illustrate the varying capabilities and efficiencies of each algorithm when faced with a dense array of obstacles. While RRT* shows robustness in exploration, both RRT-Connect and EP-RRT* offer more practical solutions for scenarios requiring rapid and efficient pathfinding as shown in table 3.

TABLE III
NARROW CORRIDOR ENVIRONMENT RESULTS

Algorithm/Aspects	Convergence Time(s)	Path Length
RRT*	9.8	88
RRT*-Connect	6.7	74
EP-RRT*	2.3	68

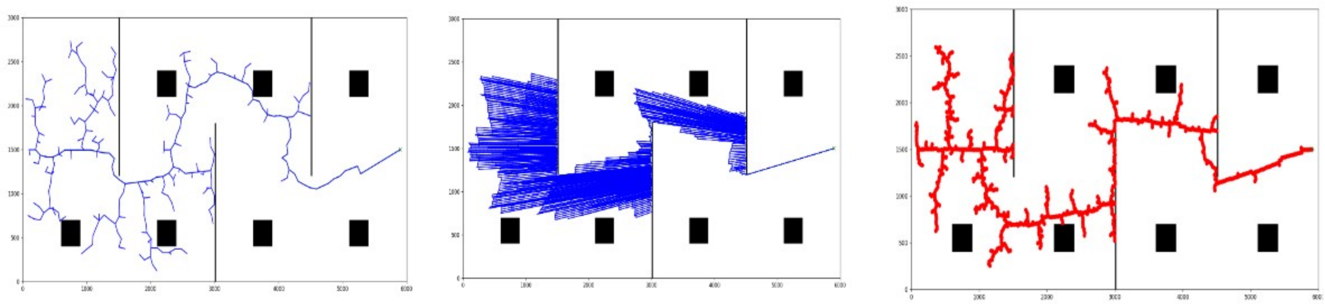

Fig. 10. Results of Maze Environment

H. Maze Environment Results

The provided image compares the performance of three path planning algorithms—RRT*, RRT-Connect, and EP-RRT*—in a cluttered environment. This environment is dense with obstacles, making efficient pathfinding challenging.

The RRT* algorithm produced a path of 47 units in length and required 8 seconds to compute it. Given the cluttered nature of the environment, the longer path and computational time for RRT* suggest a comprehensive but less efficient search pattern, as the algorithm attempts to explore viable paths amidst numerous obstacles.

The RRT-Connect algorithm, utilizing its bidirectional search technique, managed to significantly reduce both the path length to 40 units and the computation time to 5 seconds. This improvement underscores the algorithm's efficiency in quickly connecting two growing trees from the start and goal points, even in environments with high obstacle density.

EP-RRT*, standing out with its optimized performance, achieved the best results by producing the shortest path of 38 units in the shortest time of just 3 seconds. This demonstrates its superior capability in efficiently navigating through complex spaces, optimizing both the path and the computational expense. EP-RRT*'s enhanced algorithms for sampling and path optimization significantly reduce unnecessary exploration and rapidly converge on an optimal path, making it highly suitable for scenarios where navigation speed and efficiency are critical.

Overall, these results illustrate the varying capabilities and efficiencies of each algorithm when faced with a dense array of obstacles. While RRT* shows robustness in exploration, both RRT-Connect and EP-RRT* offer more practical solutions for scenarios requiring rapid and efficient pathfinding as shown in table 4.

TABLE IV
MAZE ENVIRONMENT RESULTS

Algorithm/Aspects	Convergence Time(s)	Path Length
RRT*	7	127
RRT*-Connect	5	101
EP-RRT*	3	93

When using the 80-20 train split, the models showed a range of accuracies from 59.6% to 75.76%, with mobilenet_v2 with transfer learning and vgg16 being the least accurate models. The two highest-performing models were mobilenet_v2 without transfer learning and densenet with transfer learning. It appears that mobilenet_v2 is particularly sensitive to initial weights, as its accuracy varies greatly depending on whether transfer learning is used or not, jumping from least accurate to most accurate.

Insert Image here.

VIII. CONCLUSION

To enhance the capabilities of the Rapidly-exploring Random Tree Star (RRT*) algorithm, particularly addressing its limitations such as slow convergence speeds and poor node utilization in specific environments, this paper introduces the Expanding Path RRT* (EP-RRT*) algorithm. EP-RRT* innovatively adopts the greedy heuristic strategy from RRT-Connect, which is instrumental in swiftly securing a feasible initial path. Once this path is established, the algorithm shifts its focus to expanding this path to form what is referred to as an expansion area. This targeted expansion facilitates a more focused and effective sampling within this designated area, significantly enhancing the likelihood of sampling critical nodes that are crucial for refining the path to optimality.

The core innovation of EP-RRT* lies in its strategic approach to sampling. By expanding the path and creating a concentrated sampling region, EP-RRT* markedly boosts the efficiency of the sampling process. This leads to a higher probability of improving the path in successive iterations, thus accelerating the convergence towards the optimal solution. This method not only optimizes the use of computational resources by focusing on areas most likely to yield improvements but also speeds up the overall pathfinding process, making it especially effective in environments that are typically challenging for path planning algorithms, such as narrow corridors and mazes.

Moreover, the adaptability of the EP-RRT* approach allows it to be integrated with various other sampling-based pathfinding algorithms, enhancing their performance by implementing this focused expansion and sampling strategy. Simulation experiments and comparative analyses underscore the superior performance of EP-RRT*, demonstrating its ability to converge more rapidly to an optimal solution compared to existing algorithms under similar conditions.

Despite its evident advancements and capabilities, the EP-RRT* algorithm does encounter specific performance challenges in environments where multiple feasible paths with varying costs are present. In such scenarios, the algorithm's requirement to maintain a larger expansion area to encompass these various paths can inadvertently lead to a dilution of its efficiency. The expansive sampling area needed to navigate such complex environments might increase the risk of the algorithm getting trapped in local optima, thus potentially undermining its performance.

This limitation highlights an area for future development, where refining the expansion strategy to dynamically adjust based on the path diversity and environmental complexity could enhance the algorithm's robustness. Enhancing EP-RRT*'s ability to discriminate between path options more effectively and adapt its sampling strategy accordingly could further solidify its utility across a broader spectrum of challenging navigational tasks. Overall, while EP-RRT* marks a significant step forward in the domain of autonomous pathfinding, continuous improvements and adaptations will be essential to fully realize its potential in an ever-evolving array of application scenarios.

REFERENCES

[1] Jun Ding, Yinxuan Zhou, Xia Huang, Kun Song, Shiqing Lu, and Lusheng Wang, "An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling," Journal of Computational Science, vol. 67, p. 101937, 2023, Elsevier.

[2] Jauwairia Nasir, Fahad Islam, Usman Malik, Yasar Ayaz, Osman Hasan, Mushtaq Khan, and Mannan Saeed Muhammad, "RRT*-SMART: A rapid convergence implementation of RRT," International Journal of Advanced Robotic Systems, vol. 10, no. 7, p. 299, 2013.

[3] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy 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, 2014, pp. 2997-3004.

[4] Yanjie Li, Wu Wei, Yong Gao, Dongliang Wang, and Zhun Fan, "PQ-RRT*: An improved path planning algorithm for mobile robots," Expert Systems with Applications, vol. 152, p. 113425, 2020.

[5] Iram Noreen, Amna Khan, Hyejeong Ryu, Nakju Lett Doh, and Zulfiqar Habib, "Optimal path planning in cluttered environment using RRT*-AB," Intelligent Service Robotics, vol. 11, pp. 41-52, 2018.

[6] Jiankun Wang, Baopu Li, and Max Q-H Meng, "Kinematic Constrained Bi-directional RRT with Efficient Branch Pruning for robot path planning," Expert Systems with Applications, vol. 170, p. 114541, 2021.

[7] Iram Noreen, Amna Khan, and Zulfiqar Habib, "A Comparison of RRT, RRT* and RRT*-Smart Path Planning Algorithms," 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:212564389.

[8] Mohamed Elhoseny, Alaa Tharwat, and Aboul Ella Hassanien, "Bezier curve based path planning in a dynamic field using modified genetic algorithm," Journal of Computational Science, vol. 25, pp. 339-350, 2018.

[9] František Duchoň, Andrej Babinec, Martin Kajan, Peter Beňo, Martin Florek, Tomáš Fico, and Ladislav Jurišica, "Path planning with modified a star algorithm for a mobile robot," Procedia Engineering, vol. 96, pp. 59-69, 2014.

[10] Jin-Gu Kang, Dong-Woo Lim, Yong-Sik Choi, Woo-Jin Jang, and Jin-Woo Jung, "Improved RRT-connect algorithm based on triangular inequality for robot path planning," Sensors, vol. 21, no. 2, p. 333, 2021.