Received: June 26, 2023. Revised: July 20, 2023.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Enhanced Global Path Planning Efficiency by Bidirectional A*, Gradient Descent, and Orientation Interpolation Algorithms

Zuhair A. Ahmed[1]* **Safanah M. Raafat[1]**

[1]*University of Technology, Iraq*
* Corresponding author's Email: cse.20.27@grad.uotechnology.edu.iq

Abstract: This paper presents an innovative approach to global path planning (GPP) that uniquely integrates an enhanced Bidirectional A* search algorithm, gradient descent smoothing, and orientation filter interpolation. These methods are applied within a quadratic approximation-based potential field and global cost map. The key idea is successfully integrating these powerful techniques, ensuring minimal impact on computational time while enhancing path size and smoothness. Rigorous testing in environments like the Amazon web services (AWS) RoboMaker Hospital Gazebo environment and the sections at the control and systems engineering building at Iraq's University of Technology (CSE-UOT) in Iraq shows significant improvement over existing planners like wavefront GPP and robot operating system (ROS) default GPP. The proposed algorithm reduced pathfinding time by approximately 60.3% and path size by 9.0%. These findings highlight the efficiency of the proposed global planner, suggesting it as a robust solution for autonomous robotic navigation in complex architectural scenarios.

Keywords: Global path planning, Bidirectional A*, Gradient descent smoothing, Orientation filter interpolation, Mobile robotics.

1. Introduction

In mobile robotics, achieving autonomous navigation represents a formidable challenge that encompasses the determination and execution of an optimal trajectory from the robot's current position to a predestined location [1]. The execution of autonomous navigation in mobile robots involves a set of critical tasks that include creating accurate maps of the environment, precise localization of the robot's pose, sophisticated motion planning, and efficient trajectory tracking [2].

Motion planning, a core aspect of this process, aims to create a comprehensive path preventing collisions while maneuvering from the robot's current location to its desired destination. The algorithms underpinning this process are principally classified into global path planners (GPPs) and real-time obstacle avoidance algorithms, commonly called local planners [3]. In path planning, GPPs utilize an offline model or a static environmental map to compute an optimal path from the robot's current location to its goal. These global paths provide an overarching strategy for navigation, facilitating long-range decisions. Meanwhile, local planners operate dynamically, determining the robot's motion in real time by harnessing sensory data. This mechanism enables the robot to adapt to any unforeseen obstacles or changes in the environment, ensuring effective short-range decisions and immediate collision avoidance.

Path planning's effectiveness is gauged using metrics like path length, turning angle, computation time, and risk factor. However, most existing algorithms exhibit limitations, such as paths that are too close to obstacles or unnecessarily long paths due to right-angle turns, compromising the planned path's robustness and reducing the mobile robot's speed [4].

In mobile robotics, the A* algorithm stands as a reputable tool for path planning, especially in static environments where its performance excels [5]. However, as map complexity and size grow, this algorithm's efficiency is curtailed, leading to increased computational demands and prolonged

processing time [6]. Multiple strategies have been proposed to refine the performance of the A* algorithm, encompassing a myriad of heuristic algorithms, including genetic algorithms (GA), simulated annealing algorithms (SA), ant colony algorithms (AOC), and particle swarm algorithms (PSO) [7, 8]. Each of these algorithms exhibits its unique strengths and challenges. For instance, GA is prone to premature convergence [9]. At the same time, SA and AOC struggle with slow convergence speed, with the latter also prone to becoming ensnared in local optima [10]. Furthermore, attempts to enhance the efficacy of path planning have resulted in various techniques and strategies, like introducing a "safety distance" constraint in unmanned surface vehicles (USVs) path planning. However, such methods frequently neglect to account for map obstacles [11]. Another approach employs smoothing techniques for collision avoidance as a distinct phase between global and local planning or during local planning. These techniques, albeit functional, have the adverse effect of increasing the computational burden on the robotic system [12]. Despite the strides made in augmenting the A* algorithm's performance, like the utilization of varied heuristic functions [13], expanding the search neighborhood [14], and refining the cost function [15], most of these advancements tend to optimize either efficiency or robustness but seldom both [16]. This introduces a challenge as an effective path-planning algorithm ought to concurrently maintain a balance between efficiency and robustness [17]. Some notable attempts at enhancement include Zhang et al.'s implementation of a regular hexagonal grid [18], the breaking of the single-step grid shape restriction by the Theta* algorithm [19], and Ju et al.'s Weighted JPS algorithm. While these advancements have significantly improved the A* algorithm's efficiency and path length, they have not satisfactorily addressed robustness, leading to paths that are too close to obstacles, contain numerous inflection points, and involve excessive right-angle turns [19].

Several works have attempted to address these limitations. Although the conventional relaxed version of the A* algorithm is efficient [20], it often results in paths that are not smooth or not feasible for real-world implementation [21]. Another work enhanced the A* algorithm by integrating it with a potential field and cost map, which often improved the found path's quality, but it often increased the computational time significantly [22]. Furthermore, some works have focused on smoothing the path, such as the default ROS GPP and the Wavefront GPP [23]. While effective in generating smoother paths, these methods often result in longer paths and increased computational time.

Given these challenges, this paper proposes an approach that aims to balance efficiency and robustness in the context of mobile robotics path planning. A comprehensive GPP that integrates several strategies in response to these issues has been proposed. These include a quadratic approximation method for calculating potential values of the potential field (PF), Bidirectional A* for efficient pathfinding, discrete gradient descent for interpolated path, and linear interpolation for filtering waypoints' orientations.

The proposed GPP addresses these limitations by amalgamating several strategies, offering a reliable, feasible, quick, and computationally efficient solution. Its effectiveness has been demonstrated using the robot operating system (ROS)-based TurtleBot3 burger (TB3B) robot in a navigation test, underscoring the practicability of the paths generated by the proposed planner.

The remainder of this paper is structured as follows: The proposed GPP algorithm is described in section 2. Section 3 delves into the experimental design and evaluation metrics. Section 4 assesses the proposed algorithm's performance, compares it to other algorithms in realistic environments using the developed benchmark tool, and discusses the results. Lastly, the conclusions are presented in the final section.

2. Proposed work

The GPP structure in Fig. 1 proposes a unique amalgamation of five sequential techniques. These techniques work harmoniously to generate an optimal, pre-smoothed global path for a differential drive mobile robot to navigate in a complex, real-world environment, represented as an occupancy grid map (OGM). The subsequent sections elucidate a distinctive strategy that merges algorithmic procedures with mathematical principles to compute the most efficient path for the robot.

2.1 Global cost map configuration

In robotic path planning, cost map configuration is pivotal, influencing the robot's recognition of and interaction with obstacles. The robot discerns obstacles through their associated high costs, thus enabling it to circumnavigate these barriers efficiently. Cost values will be allocated to the individual cells in the OGM, contingent on their distance from obstacles. For illustrative purposes, Fig. 2. depicts an example of an OGM containing a cantered obstacle. In contrast, Fig. 2.b displays its corresponding global cost map layer and grayscale

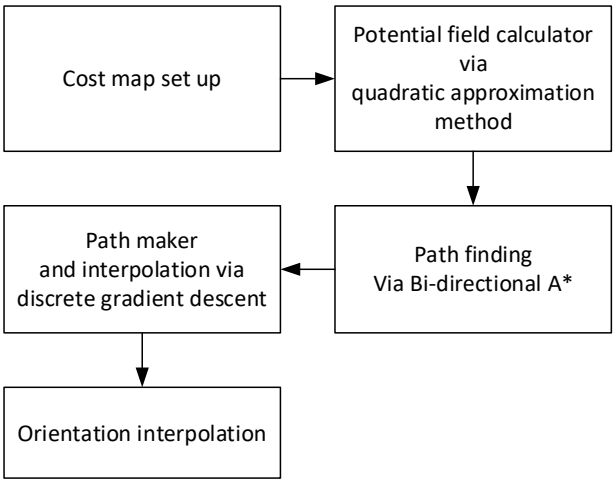


Figure. 1 Proposed GPP structure

gradient surrounding the obstacle in the map's central region, illustrating the varying cost values in proximity to these obstacles.

Initially, the cost map layer is derived from an OGM layer, which portrays the environment in terms of free spaces, occupied areas, and unknown regions. The OGM undergoes initialization, whereby cost values are allotted to cells based on their distances from obstacles. The categorization of cells is illustrated in Fig. 3 as follows:

- Free cost cells: These cells are allotted a value of 0, denoting areas without obstacles, allowing the robot to move without hindrance.
- Inflation cost cells: These cells are attributed values ranging from 1 to 127, indicating their proximity to obstacles. The closer a cell is to an obstacle, the higher its inflation cost.
- Circumscribed cost cells: These cells have an assigned value of 128, delineating the obstacles' boundaries.
- Inscribed cost cells: With a value of 253, these cells indicate spaces where the robot's radius can fit within the cell.
- Lethal cost cells: These cells bear a value of 254 and signify the presence of obstacles that are impassable for the robot.

An exponential decay function ensures a gradual decrease in cost values as the distance from obstacles increases. The cost function is expressed as:

$$C = e^{-\alpha\,(d_{obs}-r_{ins})}\,V_{lethal} \tag{1}$$

where:
C : The cost assigned to a specific grid cell.
α : A scale factor regulating the rate of decay.
d_{obs} : The distance from the nearest obstacle.
r_{ins} : The inscribed radius of the robot.
V_{lethal} : The default lethal cost value, typically set at 254.

This methodology integrates a buffer zone around obstacles, designated as the "fatal zone." This zone is assigned the highest cost value of 253-128, ensuring the robot maintains a safe berth from obstacles, thereby averting collisions. The cost values surrounding this fatal zone do not appear consistent, and it is updated based on the exponential decay function described in Eq. (1). Global cost map facilitates efficient path planning while minimizing the risk of collisions.

2.2 PF calculation

In robotic navigation and path planning, computing PF is essential. It involves setting up a virtual environment represented by a grid where each cell has a potential value. These potentials help guide the robot to its destination while avoiding obstacles. PFs are composed of two main components: attractive and repulsive potentials, U_{att} and U_{rep}, respectively. The attractive potential is linked to the goal, drawing the robot towards it. The repulsive potential is associated with obstacles, pushing the robot away from them to avoid collisions. The total potential U is a combination of these two potentials [24]:

$$\mathrm{U} = U_{att} + U_{rep} \tag{2}$$

The attractive potential is generally proportional to the square of the distance to the goal:

$$U_{att} = \frac{1}{2}\,\zeta\,DistanceToGoal^2 \tag{3}$$

Repulsive potential U_{rep} is inversely proportional to the square of the distance to the obstacles:

$$U_{rep} = \frac{1}{2}\eta\left[\frac{1}{DistanceToObstacle}\right]^2 \tag{4}$$

where ζ and η are positive constants representing the strengths of the attractive and repulsive potentials, respectively.

However, the PF method may encounter issues such as trapping the robot in local minima [25]. To address these issues and improve the responsiveness of the PF to the environment's spatial layout, researchers can utilize advanced methods like the Quadratic approximation for grid potentials. In the

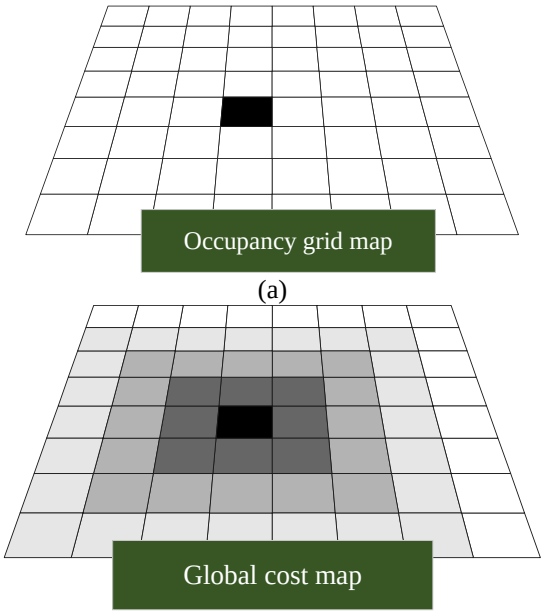


Figure. 2 Illustration of: (a) OGM layer and (b) Global cost map layer.

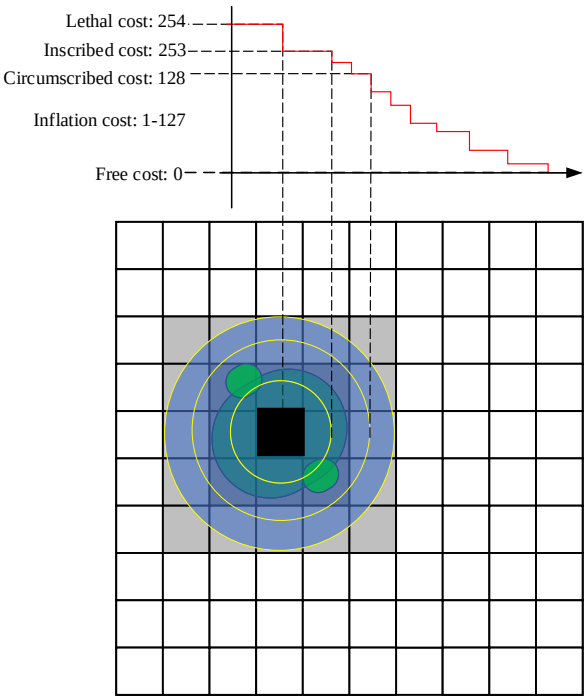


Figure. 3 Cells value categorization of the global cost map

Quadratic approximation for grid potentials, a second-degree polynomial is used to approximate the potential value of a cell $P(n)$ based on its neighboring cells and a cost value using variables, t_a, h_f, and d_c [26]. It is more adaptive, considering each cell's spatial configuration and costs. Here is what each variable represents:

- t_a: This means the minimum potential among the neighboring cells of the cell n. It signifies the lowest potential value around the cell, used as a baseline for calculating its potential.
- h_f : This represents the cost of the current cell n. It could be considered a neutral cost or proximity to obstacles. It is an integral part of calculating the new potential, considering the inherent difficulty or ease in moving through the cell.
- d_c : This is the relative cost between the lowest neighboring potentials. It provides a measure of the variation in potential among the adjacent cells. This value helps determine how much the current cell's potential should be influenced by its surroundings.

If the relative cost d_c between the lowest neighboring potentials is greater than or equal to the cost h_f, the potential $P(n)$ is calculated as:

$$P(n) = t_a + h_f \tag{5}$$

Otherwise, the potential is calculated as follows:

$$P(n) = t_a + h_f \left[-0.2301\left(\frac{d_c}{h_f}\right)^2 + 0.5307\left(\frac{d_c}{h_f}\right) + 0.7040\right] \tag{6}$$

moreover, it can be further generalized as:

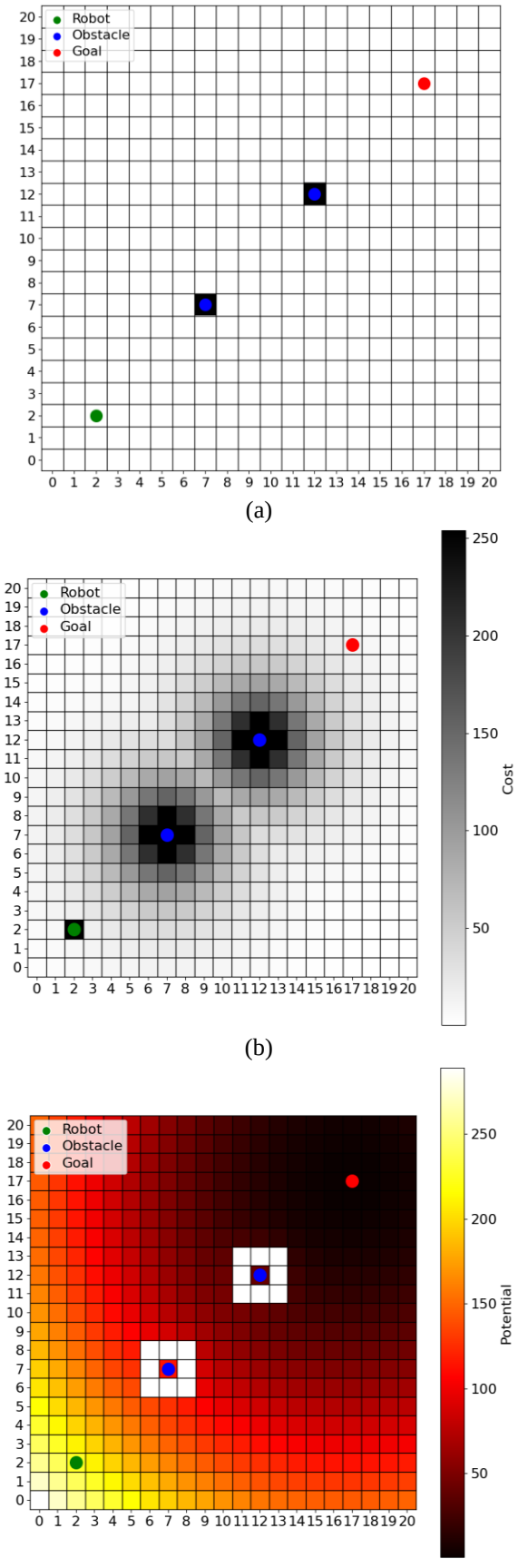
$$P(X) = \mathrm{a}X^2 + \mathrm{bX} + \mathrm{c} \tag{7}$$

where X represents the cell's position, and a, b, and c are coefficients determined based on the potentials of the neighboring cells.

These equations are derived for efficient and adaptive estimation of the potential of a cell in a grid. Eq. (5) is a more straightforward estimation when the variation in potential among the neighbors, d_c, is high. Eq. (6) is a more adaptive estimation, considering the low potential of d_c. Using t_a, h_f, and d_c, in this manner, helps to make the PF more responsive to the spatial configuration of the environment. It enables smoother and more efficient paths for robot navigation.

Fig. 4 provides a visual representation of the (a) OGM, (b) the associated global cost map, and (c) the PF map. The strength of the attractive potential increases as we approach the goal, while the repulsive potential is notably pronounced around the two central obstacles. This graphical representation offers crucial insights into how PFs navigate the robot through the environment.

In addition, Fig. 5 depicts a flowchart that elucidates the sequential procedure employed in



(a)

(b)

(c)

Figure. 4 Visual representation of OGM, global cost map, and the PF map: (a) OGM visualization, (b) Global cost map visualization, and (c) OGM visualization

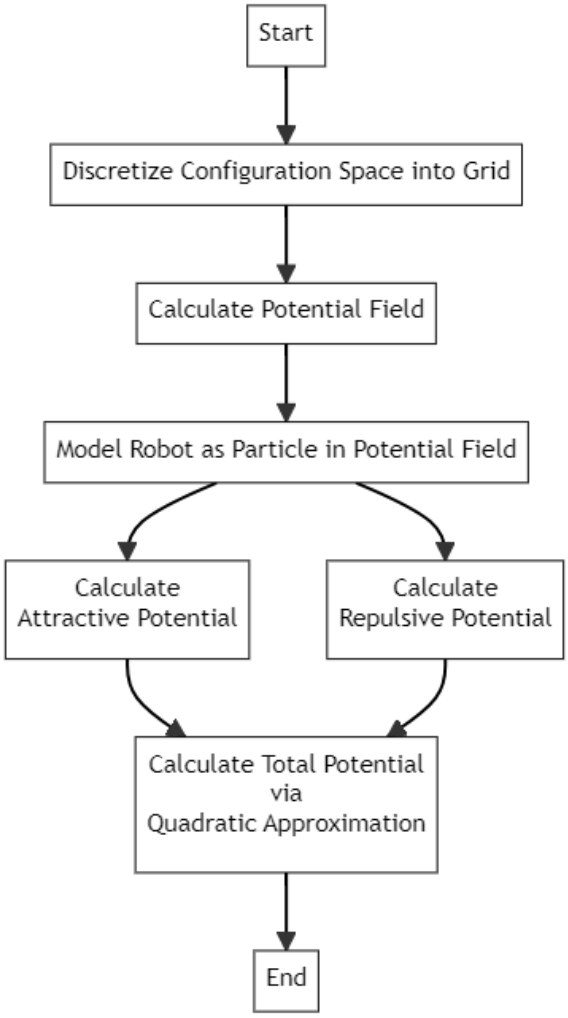


Figure. 5 Workflow of PF calculation via quadratic approximation

calculating the PF. The discretization of the configuration space initiates the process into a grid structure. Following this, the attractive and repulsive potentials are derived. The culmination of this sequence involves calculating the total potential through Quadratic approximation.

Notably, adopting Quadratic approximation for grid potentials results in a more refined and precise estimation of the PF. However, this refinement is accompanied by an escalation in computational resource requirements. Therefore, a judicious equilibrium must be maintained between computational accuracy and efficiency.

2.3 Bidirectional A* algorithm

This paper adopts a variation of the widely regarded A* search algorithm, namely the Bidirectional A* search algorithm. Distinct from the traditional A*, the Bidirectional A* initiates search concurrently from both the start and goal nodes, resulting in two simultaneous, opposing searches.

The first search progresses forward from the start node, while the second progresses backward from the goal. The point at which these searches meet or overlap signifies that a path has been established [27]. This Bidirectional strategy yields significant efficiency enhancements compared to the traditional A* algorithm, particularly in expansive search spaces. The algorithm can reduce the search area and lessen the computational burden. Nonetheless, it adheres to the fundamental principle of the A* algorithm, which is to minimize the sum of the actual cost from the start node to a given node and the heuristic estimated cost from that node to the goal.

Mathematically, the total cost function, $f(n)$, for a given node n in the A* algorithm is defined as the sum of the actual cost, $g(n)$, from the start node to n and the heuristic estimated cost, $h(n)$, from n to the goal. This relationship can be represented as:

$$f(n) = g(n) + h(n) \tag{8}$$

In the Bidirectional A* algorithm, two distinct cost values are maintained for each node. The actual cost from the start node to n , $f_1(n)$. The actual cost from n to the goal node $f_2(n)$. Subsequently, the forward and backward costs for each node n are defined as $g_1(n)$ and$f_2(n)$, respectively, and can be represented as:

$$f_1(n) = g_1(n) + h(n) \tag{9}$$

$$f_2(n) = g_2(n) + h(n) \tag{10}$$

This work employs the Manhattan distance as the heuristic, a common choice for grid-based environments where movement is restricted to horizontal and vertical directions. The search process concludes when the forward and backward paths intersect, at which point the total costs ($f_1(n)$ and $f_2(n)$), are minimal, ensuring the minimization of the total cost for the complete path from the start to the goal. Fig. 6 provides a graphical illustration related to this subsection and continues the example in Fig. 4. It demonstrates the forward and backward searching spaces in the Bidirectional A* algorithm and visually highlights where the best cells from both searches meet. Incorporation of the Bidirectional A* algorithm is anticipated to bolster the robustness and efficiency of pathfinding in the proposed system. This optimization is especially vital in complex environments where expeditious and effective pathfinding is paramount.

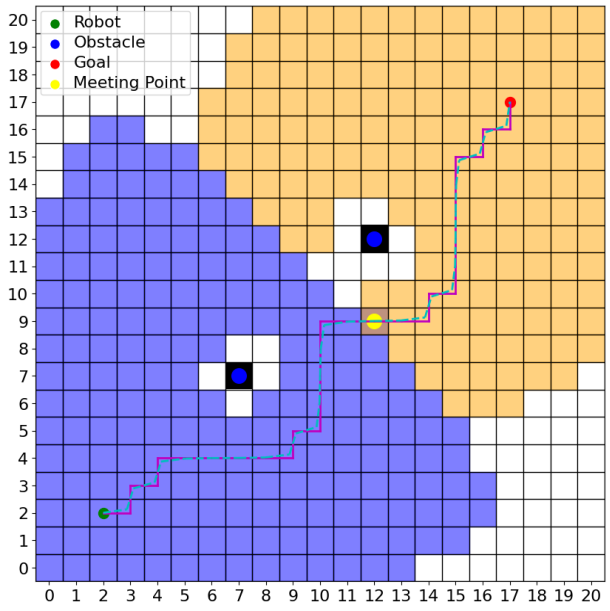


Figure. 6 Bidirectional A* search with gradient descent

2.4 Discrete gradient descent for path generation

The path generation phase employs a discrete variant of the gradient descent algorithm. This technique assimilates the principles of gradient descent to yield a path conducive to fluid robot movements and efficient evasion of obstacles. The algorithm iteratively progresses toward the steepest descent by utilizing the PF.

The mathematical description of the gradient descent algorithm is as follows [28]: Given a PF P (x, y), where (x, y) denote the coordinates of a cell, the gradient of the PF at point (x, y) is expressed by the vector:

$$\nabla \mathrm{P}(x, y) = \left[\frac{\partial \mathrm{P}}{\partial \mathrm{x}}, \frac{\partial \mathrm{P}}{\partial \mathrm{y}}\right] \tag{11}$$

This vector indicates the direction of the steepest ascent in the PF. The negative gradient, which corresponds to the steepest descent, is used to move toward the goal. The negative of the gradient is taken:

$$\nabla \mathrm{P}(x, y) = \left[-\frac{\partial \mathrm{P}}{\partial \mathrm{x}}, -\frac{\partial \mathrm{P}}{\partial \mathrm{y}}\right] \tag{12}$$

The subsequent position on the path (x_{next} t, y_{next}) is determined from the current position ($x_{current}$, $y_{current}$) by:

$$x_{next} = x_{current} - \delta \frac{\partial \mathrm{P}}{\partial \mathrm{x}} \tag{13}$$

$$y_{next} = y_{current} - \delta \frac{\partial \mathrm{P}}{\partial \mathrm{y}} \tag{14}$$

here, δ is a positive constant dictating the step

size.

This iterative process commences at the goal and retrogresses to the starting position, with each step advancing toward the steepest descent. The result is a smooth, continuous path aligned with the PFs gradient. In practical terms, this path represents an interpolation of the steepest descent in the PF, ensuring a feasible and efficient trajectory for the robot.

Crucially, the derivatives $\partial P/\partial x$ and $\partial P/\partial y$ are approximated using finite differences in the implementation, as the PF is discretized. This adaptation renders the gradient descent algorithm compatible with path generation in discrete spaces typical in robotic path planning. Furthermore, the discrete gradient descent method operates within the framework of the ROS GPP. Fig. 6 also offers a graphical illustration of the discrete gradient descent process in path generation. The Figure elucidates the smooth path obtained through this algorithm, emphasizing the PF interpolation of the steepest descent in the PF. The discrete gradient descent algorithm generates smooth paths in discrete environments. It is a viable solution for robotic path planning that necessitates natural robot movements and strategic obstacle circumvention.

2.5 Linear interpolation of orientation

In the terminal phase of the methodology, orientation filtering is implemented through linear interpolation, centering primarily on the yaw angle amid waypoints. From a mathematical standpoint, the linear interpolation of an angle θ situated between θ_1 and θ_2 can be articulated as:

$$\theta = (1 - t)\,\theta_1 + t\,\theta_2 \qquad (15)$$

here, t is a scalar that spans from 0 to 1. When t assumes the value of 0, θ equals θ_1, and conversely, when t is 1, θ equates to θ_2.

Incorporating linear interpolation for orientation filtering is a critical component within a broader framework that amalgamates a constellation of cutting-edge robotic path-planning paradigms. This holistic strategy cultivates a pragmatic and efficacious resolution for global path planning. Orientation filtering, particularly the yaw angle, ensures seamless transitions between waypoints, rendering the robotic movements more natural and streamlined. It is particularly imperative in complex environments where the robot is tasked with maneuvering around obstacles to reach its destination most efficiently. The orientation filtering by linear interpolation is visually exemplified in Fig. 7. This

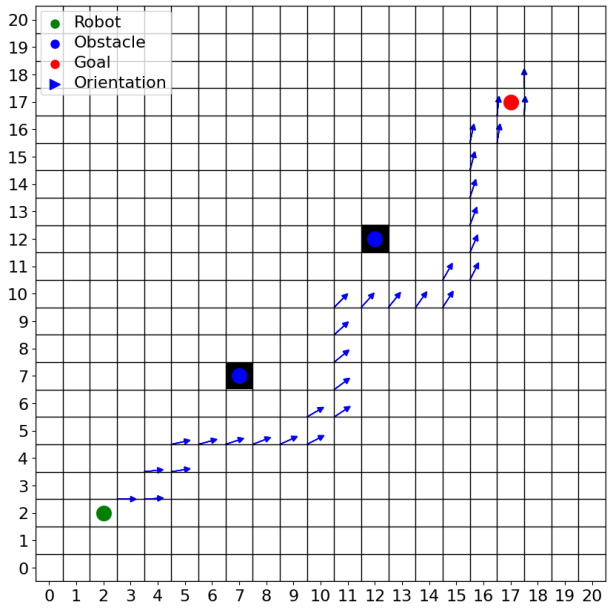


Figure. 7 Orientation interpolation

illustration highlights the transitioning of the orientation through linear interpolation from an initial 0-degree orientation to a 90-degree orientation. This depiction brings the continuity and smoothness introduced by employing linear interpolation for orientation filtering to the fore.

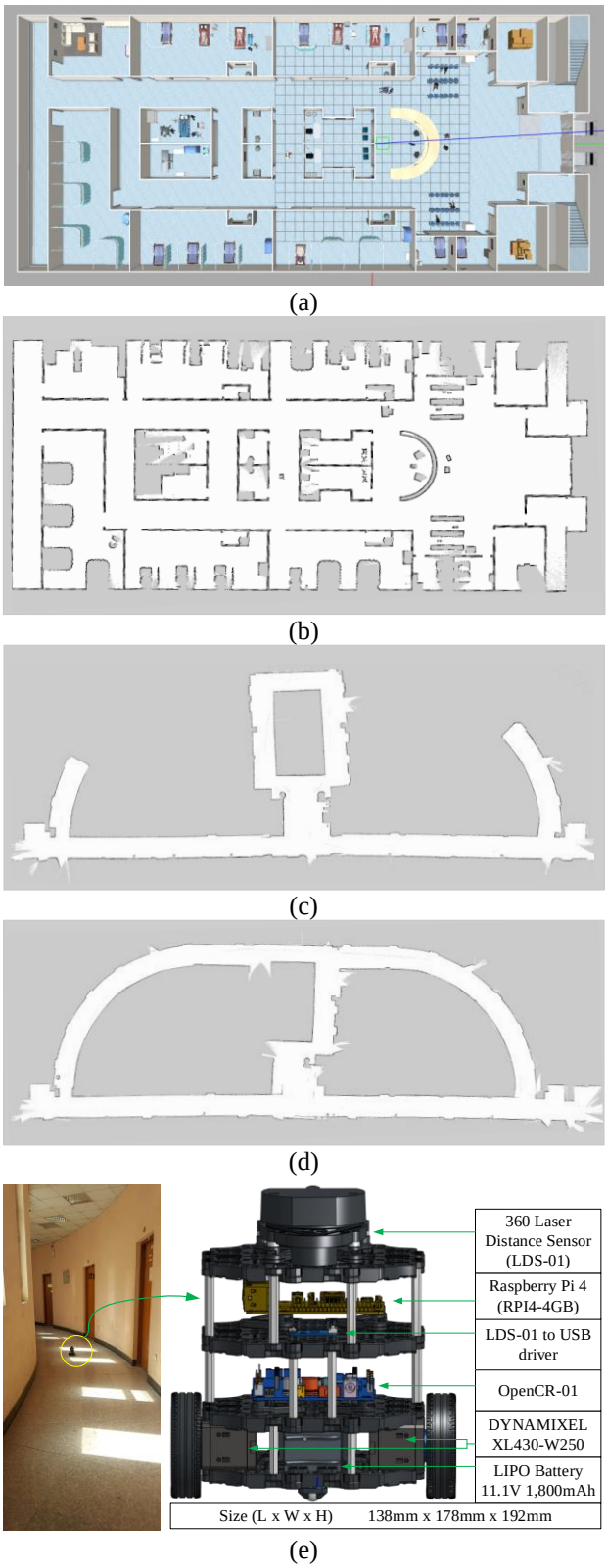
As the culmination of a sophisticated methodology that combines quadratic approximation for PFs, Bidirectional A* for pathfinding, discrete gradient descent for path generation, and linear interpolation for orientation filtering, this final stage enhances the overall reliability, feasibility, expediency, and computational efficiency of the proposed global path-planning algorithm. The following sections of this article will furnish experimental evidence underpinning the effectiveness of the proposed methodology in robotic path planning.

3. Experimental design and evaluation metrics

3.1 Experimental environments

The performance of the proposed GPP was evaluated using the TB3B with an optimal configuration through Gmapping, as detailed in previous work [29]. This evaluation involved the generation of three benchmark occupancy grid maps OGMs, as depicted in Fig. 8.

Fig. 8 illustrates the experimental environments and the properties of TB3B. The primary benchmark map was the Amazon Web Services (AWS) RoboMaker Hospital Gazebo environment [30]. This environment skillfully replicates a real-world



(a) (b) (c) (d) (e)

Figure. 8 Experimental environments and TB3B properties (a) AWS hospital environment (b) OGM of AWS hospital environment (c) OGM of CSE-UOT environment 1st floor (d) OGM of CSE-UOT environment 2nd floor and (e) TB3B in a corridor of CSE-UOT environment 2nd floor [29]

Table 2. Dimensions of the experimental environments

Environment	Longest length (m)	Longest width (m)
AWS Hospital environment	55	24
CSE-UOT environment 1st floor	60	17
CSE-UOT environment 2nd floor	60	17

hospital setting, with rooms of varying sizes, curved and straight walls, and multiple obstructions (see Fig. 8.a and Fig. 8.b).

Additionally, two OGMs were used, representing distinct sections of the Control and Systems Engineering building at Iraq's University of Technology (CSE-UOT). These environments, characterized by a complex network of long, winding, straight, and curved corridors (see Fig. 8.c and Fig. 8.d), added complexity to the testing framework.

The start and goal points, which were significantly separated and embedded within these intricate layouts, were carefully chosen to test the proposed GPP. The dimensions of these environmental maps, expressed in meters (m), are approximately tabulated in Table 1.

3.2 Evaluation metrics

A bespoke benchmarking tool was employed to compare several GPPs in-depth. This tool is tailored to gauge the efficiency of GPPs in terms of path size (PS), planning time (PT), and visual inspection.

PS: the first metric to indicate the complexity of the paths generated by the algorithms is PS. It is gauged by measuring the number of waypoints that constitute the paths. This metric serves as a surrogate for the physical path length. It alleviates the complexities associated with precise path length measurement in various environments. Using PS as a metric provides a standard for direct comparison among different planners without considering the methods by which the waypoints are generated.

PT: PT is the second metric employed as instrumental in quantifying the duration expended by each GPP in generating viable paths. This metric is invaluable as it can evaluate the planners' responsiveness in real-time applications where expedient decision-making is paramount.

Visual inspection: The third metric entails a qualitative analysis as a visual inspection of the paths generated by each algorithm. This evaluation is accomplished through visualizations that depict the quality of the paths. Visualization inspection is integral as it reveals potential issues such as collisions

with obstacles or unnecessarily convoluted paths. Moreover, it provides an intuitive understanding of path navigability, smoothness, and adherence to environmental constraints.

A comprehensive and rigorous benchmark tool is established for comparing different global path-planning algorithms by employing PS, PT, and visualization inspection as evaluation metrics. This evaluation paradigm is particularly pertinent for realistic environments and robot configurations, shedding light on the practical applicability and performance of the algorithms under a diverse array of conditions.

4. Results and discussions

The proposed GPP algorithm was intensively evaluated. The investigation was initiated by analyzing the enhanced A* algorithm before the application of smoothing. This was followed by the incorporation of gradient descent smoothing and orientation filter interpolation, which culminated in the final form of the proposed GPP algorithm. This refined algorithm was then subjected to a thorough evaluation process.

4.1 Quantitative analysis

The evaluation commenced with critically analyzing the enhanced A* algorithm's performance in its unsmoothed state. This preliminary phase (1st evaluation phase) rigorously assessed three variations of the algorithm: The proposed Bidirectional A* integrated with a global cost and PF maps versus A* integrated with a global cost and PF maps [22], and the conventional Relaxed A* [20]. PT and PS were employed as performance measures, detailed in Table 2.

The empirical data in Table 2 highlights the significant optimization of the A* algorithm when a global cost map and PF map are included, with the Bidirectional variant showcasing superior enhancements in PT and PS. This phase provides a benchmark for comparing the enhanced A* algorithm against its contemporaries.

The 2nd evaluation phase, presented in Table 3, builds upon the first by incorporating gradient descent and linear interpolation into the enhanced A* algorithm to constitute the final proposed GPP. This phase concentrates on the improvements attained through path smoothing, which expedites communication with the local planner, leading to more fluid robot trajectories in complex environments.

Tables 4 and 5, meanwhile, present the performance metrics from the 1st and 2nd validation stages, respectively. The proposed GPP algorithm is benchmarked against the Wavefront GPP [23] and the ROS Default GPP [31] in varying environments, such as AWS Hospital and CSE-UOT.

Table 2. Performance metrics of enhanced GPP without smoothing for the 1st evaluation phase

GPP	PT (Sec)	PS (Waypoints)
Bidirectional A* with global cost map and PF map	0.101448	1323
A* with a global cost map and PF map	0.376148	1391
A* without a global cost map and PF map	2.789501	1731

Table 3. Performance metrics of enhanced GPP with smoothing 2nd evaluation phase

GPP	PT (Sec)	PS (Waypoints)
Proposed GPP	0.065486	2165
Wavefront GPP	0.103634	2859
ROS Default GPP	0.236587	2928

Table 4. Performance metrics in 1st validation

GPP	PT (Sec)	PS (Waypoints)
Proposed GPP	0.107653	4351
Wavefront GPP	0.192431	4373
ROS Default GPP	0.806832	4480

Table 5. Performance metrics in 2nd validation

GPP	T (Sec)	PS (Waypoints)
Proposed GPP	0.124785	5088
Wavefront GPP	0.192135	5097
ROS Default GPP	0.937977	5100

The empirical data spanning these tables substantiate the proposed GPP algorithm's superior performance concerning PT and PS. Across the tables, an average improvement of roughly 60.3% in PT and 9.0% in PS were registered, compared to Wavefront and ROS Default GPPs.

4.2 Qualitative analysis

A qualitative evaluation of the proposed GPP algorithm was conducted alongside the quantitative assessment. This involved visual inspections of the paths generated by the algorithm and those produced

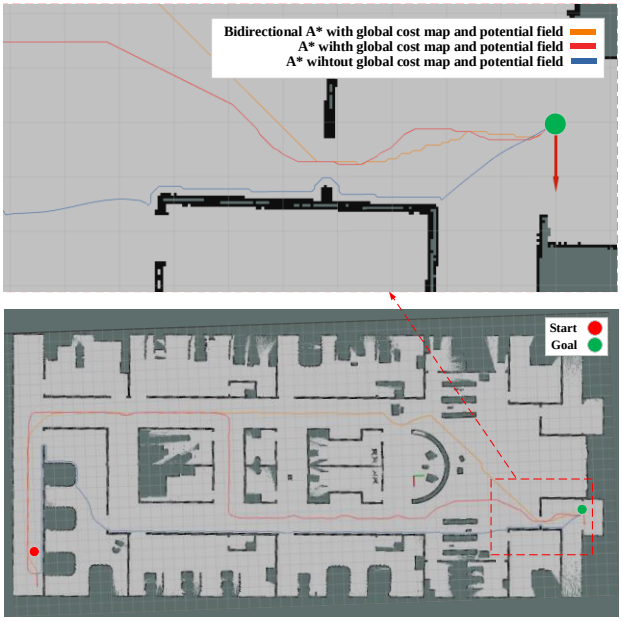


Figure. 10 Visual inspection and comparison of 1st phase

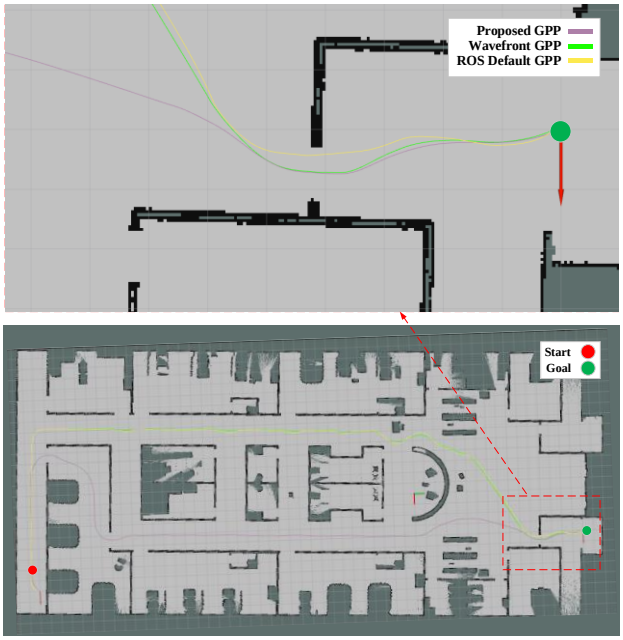


Figure. 10 Visual inspection and comparison of the 2nd phase

by wavefront GPP and ROS default GPP. Fig. 9 to 12 visually compare the first and second evaluation phases and the first and second validations, respectively.

The visual analysis affirms the superior performance of the proposed GPP algorithm, which creates more streamlined and well-aligned paths, especially noticeable within corridors and doorways. In contrast, the wavefront GPP and ROS default GPP paths exhibited excessive curvature and suboptimal alignments, particularly in complex environments and over long distances.

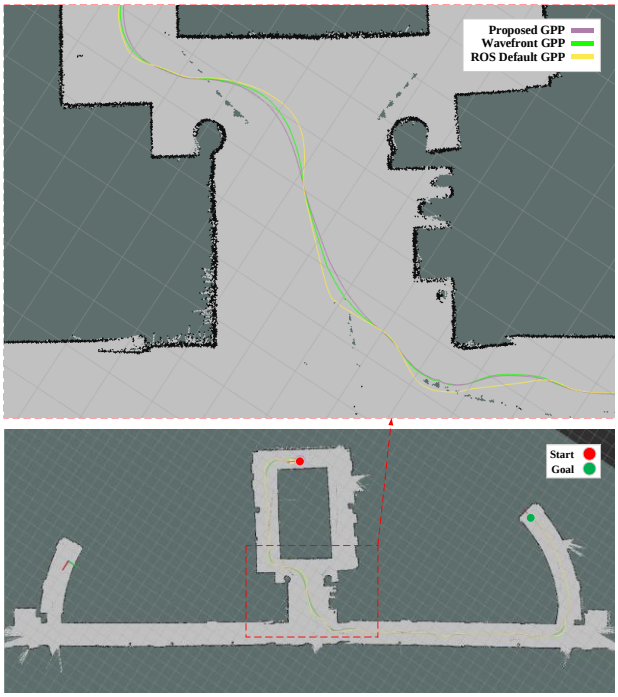


Figure. 11 Visual inspections and comparison of 1st validation

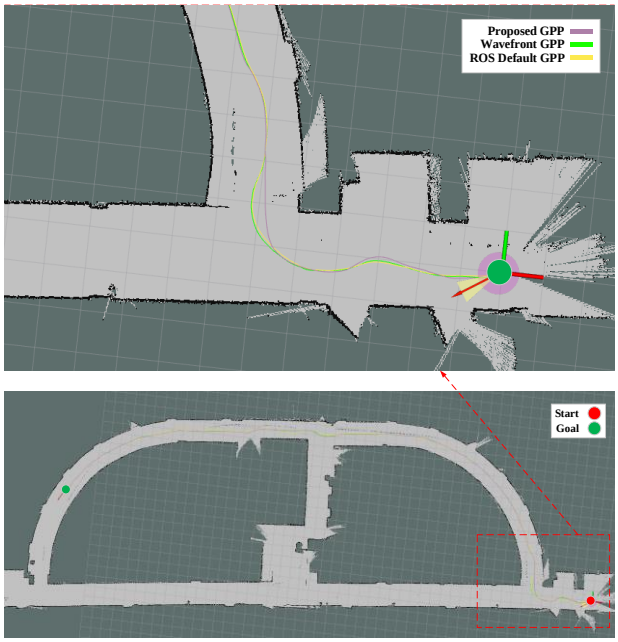


Figure. 12 Visual inspections and comparison of 1st validation

The proposed GPP algorithm's performance was compared to existing methods to highlight its scientific contribution. The algorithm's superior performance can be attributed to the unique integration of an enhanced Bidirectional A* search algorithm, gradient descent smoothing, and orientation filter interpolation. These methods, applied within a quadratic approximation-based potential field and global cost map, address the limitations and drawbacks of existing methods

discussed earlier in the paper.

The results obtained from the evaluation and validation stages demonstrate significant improvements in the performance of the proposed GPP algorithm. The algorithm's improved performance can be attributed to the unique integration of several powerful techniques, ensuring minimal impact on computational time while enhancing path size and smoothness. The algorithm's ability to generate smoother, environment-adapted trajectories underlines its potential for deployment in various complex real-world applications. The results also highlight the algorithm's balance between efficiency and robustness.

5. Conclusions

This paper focused on developing and comprehensively evaluating an innovative GPP algorithm, which synergistically combined an enhanced Bidirectional A* search algorithm with gradient descent smoothing and orientation filter interpolation. The algorithm was rigorously tested in various simulated environments that faithfully replicated real-world scenarios, providing insights into its practical applications.

The algorithm's performance was analyzed using a bespoke benchmarking tool, which assessed PT, PS, and visual inspection of paths. Gradient descent smoothing was integrated, resulting in paths that were smoother and better aligned with environmental constraints, thereby enhancing the algorithm's communication with the local planner.

Visual examination corroborated the efficiency of the proposed GPP algorithm in generating streamlined, environment-adapted paths. This superiority was further substantiated quantitatively, with significant reductions in path generation time and complexity. Notably, compared to the wavefront and ROS default GPPs, the proposed algorithm demonstrated an average improvement of approximately 60.3% in PT and 9.0% in PS.

The proposed algorithm's adaptability, computational efficiency, and capacity to generate high-quality paths underscore its practicability across various real-world scenarios. Future research avenues include examining the algorithm's performance on different robotic platforms, its adaptability to dynamic environments, and potential integration with machine learning strategies for enhanced adaptability. In summary, this research has successfully developed a robust and efficient GPP algorithm, setting the stage for further advancements and refinements in robotic path planning and navigation.

Conflicts of interest

The authors declare that they have no conflicts of interest. This research was conducted independently and without funding from any external source.

Author contributions

Conceptualization: Zuhair A. Ahmed and Safanah M. Raafat; methodology: Zuhair A. Ahmed, Safanah M. Raafat; software: Zuhair A. Ahmed; validation: Zuhair A. Ahmed, Safanah M. Raafat; formal analysis: Zuhair A. Ahmed; investigation: Zuhair A. Ahmed and Safanah M. Raafat; resources: Zuhair A. Ahmed and Safanah M. Raafat; data curation: Zuhair A. Ahmed; writing-original draft preparation: Zuhair A. Ahmed, Safanah M. Raafat; writing-review and editing: Zuhair A. Ahmed and Safanah M. Raafat; visualization: Zuhair A. Ahmed; supervision: Safanah M. Raafat; project administration: Safanah M. Raafat and Zuhair A. Ahmed; funding acquisition, Not funded.

Acknowledgments

The authors acknowledge the invaluable support of Dr. Firas A. Raheem and the staff from the Automation and Robotics Research Unit at the Control and Systems Engineering Department, University of Technology - Iraq (CSE-UOT).

List of notations

Variable	Description
The cost function	
C	The cost assigned to a specific grid cell.
α	A scale factor regulating the rate of decay
d_{obs}	The distance from the nearest obstacle
r_{ins}	The inscribed radius of the robot
V_{lethal}	The default lethal cost value, typically set at 254
Attractive and repulsive potentials	
U	Total potential
U_{att}	Attractive potential
U_{rep}	Repulsive potential
ζ	Positive constant representing the strength of the attractive potential
η	
Quadratic Approximation for Grid Potentials	
$P(n)$	The potential value of a cell

t_a	Minimum potential among the neighboring cells of the cell n
h_f	Cost of the current cell n
a, b, c	Coefficients are determined based on the potentials of the neighboring cells.
X	Cell's position
Bidirectional A* algorithm	
$f(n)$	The total cost function for a given node n
$g(n)$	Actual cost from the start node to n
$h(n)$	Heuristic estimated cost from n to the goal
$f_1(n)$	Actual cost from the start node to n in the Bidirectional A* algorithm
$f_2(n)$	Actual cost from n to the goal node in the Bidirectional A* algorithm
$g_1(n)$	Forward cost for each node n
$f_2(n)$	Backward cost for each node n
Gradient descent algorithm	
$\mathrm{P}(x, y)$	PF where (x, y) denote the coordinates of a cell
$\nabla \mathrm{P}(x, y)$	The gradient of the PF at point (x, y)
x_{next}	Subsequent position on the path
y_{next}	Subsequent position on the path
δ	Positive constant dictating the step size.
Linear interpolation of an angle θ	
θ_1	Initial angle
θ_2	Final angle
θ	Angle situated between θ_1 and θ_2
t	Scalar that spans from 0 to 1

References

[1] F. A. Raheem, S. M. Raafat, and S. M. Mahdi, "Robot Path-Planning Research Applications in Static and Dynamic Environments", in *Earth Systems Protection and Sustainability*, *Cham: Springer International Publishing*, pp. 291–325, 2022.

[2] A. A. Kareem, B. K. Oleiwi, and M. J. Mohamed, "Planning the Optimal 3D Quadcopter Trajectory Using a Delivery System-Based Hybrid Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 2, pp. 427–439, 2023.

[3] J. R. S. Ibáñez, C. J. P. D. Pulgar, and A. G. Cerezo, "Path Planning for Autonomous Mobile Robots: A Review", *Sensors (Basel, Switzerland)*, Vol. 21, No. 23, p. 7898, 2021.

[4] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent robotics", *Journal of Intelligent Manufacturing*, Vol. 33, No. 2, pp. 387–424, 2021.

[5] M. L. Muhammed, A. J. Humaidi, and E. H. Flaieh, "A Comparative Study of Graph Search Algorithms for Planar Manipulator to Avoid Obstacle Collision", *Iraqi Journal of Computer, Communication, Control and System Engineering*, Vol. 22, No. 4, pp. 99–114, 2022.

[6] D. Foead, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. Gunawan, "A Systematic Literature Review of A* Pathfinding", *Procedia Computer Science*, Vol. 179, pp. 507–514, 2021.

[7] O. A. R. A. Wahhab and A. S. A. Araji, "An Optimal Path Planning Algorithms for a Mobile Robot", *Iraqi Journal of Computer, Communication, Control and System Engineering*, Vol. 21, No. 2, pp. 44–58, 2021.

[8] M. Abed, O. Lutfy, and Q. A. Doori, "A Review on Path Planning Algorithms for Mobile Robots", *Engineering and Technology Journal*, Vol. 39, No. 5A, pp. 804–820, 2021.

[9] P. G. Luan and N. T. Thinh, "Hybrid genetic algorithm based smooth global-path planning for a mobile robot", *Mechanics Based Design of Structures and Machines*, Vol. 51, No. 3, pp. 1758–1774, 2023.

[10] X. Wu, G. Wei, Y. Song, and X. Huang, "Improved ACO-based path planning with rollback and death strategies", *Systems Science & Control Engineering*, Vol. 6, No. 1, pp. 102–107, 2018.

[11] S. Hu, S. Tian, J. Zhao, and R. Shen, "Path Planning of an Unmanned Surface Vessel Based on the Improved A-Star and Dynamic Window Method", *Journal of Marine Science and Engineering*, Vol. 11, No. 5, p. 1060, 2023.

[12] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, and C. C. Peng, "Path Smoothing Techniques in Robot Navigation: State-of-the-Art, Current and Future Challenges", *Sensors (Basel, Switzerland)*, Vol. 18, No. 9, p. 3170, 2018.

[13] D. Xiang, H. Lin, J. Ouyang, and D. Huang, "Combined improved A* and greedy algorithm for path planning of multi-objective mobile robot", *Scientific Reports*, Vol. 12, No. 1, p. 13273, 2022.

[14] H. Wang, S. Lou, J. Jing, Y. Wang, W. Liu, and T. Liu, "The EBS-A* algorithm: An improved A* algorithm for path planning", *PLOS ONE*,

Vol. 17, No. 2, p. e0263841, 2022.
[15] H. Huang, Y. Li, and Q. Bai, "An improved A star algorithm for wheeled robots path planning with jump points search and pruning method", *Complex Engineering Systems*, Vol. 2, No. 3, p. 11, 2022.
[16] A. Vagale, R. T. Bye, R. Oucheikh, O. L. Osen, and T. I. Fossen, "Path planning and collision avoidance for autonomous surface vehicles II: a comparative study of algorithms", *Journal of Marine Science and Technology*, Vol. 26, No. 4, pp. 1307–1323, 2021.
[17] A. F. Gil, E. L. Ruiz, M. G. Sánchez, and C. Castro, "A Review of Heuristics and Hybrid Methods for Green Vehicle Routing Problems considering Emissions", *Journal of Advanced Transportation*, Vol. 2022, pp. 1–38, 2022.
[18] J. Zhang, X. Wang, L. Xu, and X. Zhang, "An Occupancy Information Grid Model for Path Planning of Intelligent Robots", *ISPRS International Journal of Geo-Information*, Vol. 11, No. 4, p. 231, 2022.
[19] S. Han, L. Wang, Y. Wang, and H. He, "A dynamically hybrid path planning for unmanned surface vehicles based on non-uniform Theta* and improved dynamic windows approach", *Ocean Engineering*, Vol. 257, p. 111655, 2022.
[20] A. Ammar, H. Bennaceur, I. Châari, A. Koubâa, and M. Alajlan, "Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments", *Soft Computing*, Vol. 20, No. 10, pp. 4149–4171, 2016.
[21] K. Karur, N. Sharma, C. Dharmatti, and J. E. Siegel, "A Survey of Path Planning Algorithms for Mobile Robots", *Vehicles 2021, Vol. 3, Pages 448-468*, Vol. 3, No. 3, pp. 448–468, 2021.
[22] G. Klančar, A. Zdešar, and M. Krishnan, "Robot Navigation Based on Potential Field and Gradient Obtained by Bilinear Interpolation and a Grid-Based Search", *Sensors 2022, Vol. 22, Page 3295*, Vol. 22, No. 9, p. 3295, 2022.
[23] M. Psotka, F. Duchoň, M. Roman, T. Michal, and D. Michal, "Global Path Planning Method Based on a Modification of the Wavefront Algorithm for Ground Mobile Robots", *Robotics*, Vol. 12, No. 1, p. 25, 2023.
[24] A. K. Reddy, V. Malviya, and R. Kala, "Social Cues in the Autonomous Navigation of Indoor Mobile Robots", *International Journal of Social Robotics*, Vol. 13, No. 6, pp. 1335–1358, 2021.
[25] F. A. Raheem and M. I. Abdulkareem, "Development of A* algorithm for robot path planning based on modified probabilistic roadmap and artificial potential field", *Journal of Engineering Science and Technology*, Vol. 15, No. 5, pp. 3034–3054, 2020.
[26] X. Tang *et al.*, "Prediction-Uncertainty-Aware Decision-Making for Autonomous Vehicles", *IEEE Transactions on Intelligent Vehicles*, Vol. 7, No. 4, pp. 849–862, 2022.
[27] S. Ahmadi, G. Tack, D. Harabor, and P. Kilby, "Weight Constrained Path Finding with Bidirectional A*", In: *Proc of the International Symposium on Combinatorial Search*, Vol. 15, No. 1, pp. 2–10, 2022.
[28] S. Luo, X. Li, and Z. Sun, "An optimization-based motion planning method for autonomous driving vehicle", In: *Proc of 2020 3rd International Conference on Unmanned Systems, ICUS 2020*, pp. 739–744, 2020.
[29] Z. A. Ahmed and S. M. Raafat, "An Extensive Analysis and Fine-Tuning of Gmapping's Initialization Parameters", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 126–138, 2023.
[30] A. Robotics, "aws-robomaker-hospital-world", *AWS Robotics*, 2021. https://github.com/aws-robotics/aws-robomaker-hospital-world (accessed Jun. 19, 2023).
[31] A. Filotheou, E. Tsardoulias, A. Dimitriou, A. Symeonidis, and L. Petrou, "Quantitative and Qualitative Evaluation of ROS-Enabled Local and Global Planners in 2D Static Environments", *Journal of Intelligent & Robotic Systems*, Vol. 98, No. 3–4, pp. 567–601, 2020.