

Research Article

Mobile Robot Path Planning Based on a Generalized Wavefront Algorithm

Sifan Wu ¹, Yu Du ², and Yonghua Zhang ³

¹Smart City College, Beijing Union University, Beijing 100101, China

²College of Robotics, Beijing Union University, Beijing 100101, China

³School of Computer Science and Engineering, Beihang University, Beijing 100083, China

Correspondence should be addressed to Yu Du; duyu@buu.edu.cn

Received 16 October 2019; Accepted 11 February 2020; Published 10 March 2020

Academic Editor: Oscar Reinoso

Copyright © 2020 Sifan Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study develops a generalized wavefront algorithm for conducting mobile robot path planning. The algorithm combines multiple target point sets, multilevel grid costs, logarithmic expansion around obstacles, and subsequent path optimization. The planning performances obtained with the proposed algorithm, the A^* algorithm, and the rapidly exploring random tree (RRT) algorithm optimized using a Bézier curve are compared using simulations with different grid map environments comprising different numbers of obstacles with varying shapes. The results demonstrate that the generalized wavefront algorithm generates smooth and safe paths around obstacles that meet the required kinematic conditions associated with the actual maneuverability of mobile robots and significantly reduces the planned path length compared with the results obtained with the A^* algorithm and the optimized RRT algorithm with a computation time acceptable for real-time applications. Therefore, the generated path is not only smooth and effective but also conforms to actual robot maneuverability in practical applications.

1. Introduction

Path planning is one of the key technologies for facilitating the autonomous maneuverability of mobile robots within environments that include obstacles. The path planning process seeks to obtain an optimum collision-free path from a starting point to a target point around obstacles according to the particular criteria such as distance traveled, travel time, and incurred cost [1–3]. As such, path planning is a subject of intense interest in the mobile robot research community [4–12]. Path planning is generally divided into local path planning conducted in an unknown environment and global path planning conducted in a known environment. This study mainly focuses on the global path planning process.

Research focused on global path planning has widely adopted search-based path planning algorithms and sampling-based path planning algorithms. In terms of search-based path planning algorithms, the A^* algorithm is a classic heuristic optimal search algorithm [13] that has had a

significant impact on motion planning research. The A^* algorithm usually searches for the best path by creating a discrete state space in the form of a state lattice [14, 15]. The state lattice consists of a node, which represents a state, and a motion primitive that arrives at a neighboring node from the original node. As such, a state node is transformed by its motion primitive to another state node. Thus, the state lattice transforms the original continuous state space into a search map, and the motion planning problem becomes a search employing various search algorithms [16] for a series of motion primitives that transform the initial state to the target state. However, the planning paths obtained by the A^* algorithm are generally insufficiently smooth to conform to the maneuverability limitations of mobile robots, and the computational cost is too great for real-time operation with limited computational resources. Numerous studies have sought to address these deficiencies in the A^* algorithm. For example, Guruji et al. [17] modified the A^* algorithm to reduce the number of neighbor searches using the slope detection calculation method, thereby increasing its

computational efficiency. However, the modified algorithm generally increased the length of the planned path, and the obtained paths remain insufficiently smooth. Li et al. [18] proposed an Index Based A-Star (IBAS) algorithm, which sets three indexes for the vertex and removes useless vertices by calculating the upper limit and the lower line of the shortest distance from the source vertex to the target through the index. The effectiveness of the algorithm was verified experimentally. In addition, the A^* algorithm has been further modified by adding a machine learning model and a hierarchical structure planning method with a two-level structure. In the first level, the algorithm employs grids to find a geometric path quickly, and several path points are selected as subgoals for the next level. In the second level, an approximate policy iteration algorithm denoted as least-squares policy iteration (LSPI) is used to learn a near-optimal local planning policy that can generate smooth trajectories under the kinematic constraints of the robots [19]. Further improvements in the A^* algorithm have been obtained by dividing the nodes generated by the algorithm into smaller steps, eliminating redundant nodes, and reducing the cost of step and handover [20]. In terms of sampling-based path planning algorithms, the rapidly exploring random tree (RRT) [21] is a widely used path planning algorithm that has been applied in various path planning tasks for mobile robots [22–24]. A motion planning algorithm based on random sampling [25] essentially seeks to construct a connected graph by uniformly and randomly sampling the state space. The problem is solved when the initial and target states are in the map or can be connected to the map. Although sampling-based search algorithms can generate paths more quickly than search-based algorithms, these algorithms tend to obtain nonoptimal paths, and the random operation of the algorithms produces widely differing path results. These issues have been addressed in recent years by numerous researchers. For example, Jaillet et al. [26] proposed environment-guided RRT (EG-RRT), which combines standard RRT with a cost model based on the linear-quadratic Gaussian motion planning (LQG-MP) framework to estimate the probability of collision under uncertainty. This algorithm has been demonstrated to facilitate collision-free path planning in environments with narrow passages between obstacles. Jaillet et al. [11] proposed transition-based RRT (T-RRT), which employs transition tests and relies on the Metropolis criterion commonly used in stochastic optimization to detect new nodes and determine whether they can meet expansion requirements to ensure continuous tree growth. In addition, the algorithm applies the concept of mechanical work to produce a feasible (i.e., no collision) route with the optimum path quality. Ghosh et al. [27] proposed the Kinematic Constraints Based Bidirectional RRT (KB-RRT) algorithm, which improved the performance of Bidirectional RRT (Bi-RRT) by introducing robot kinematic constraints. Here, the bidirectional search process adopted Kinematic Constraints based Random State Search (KCRSS) to reduce the number of nodes and generate smooth paths. The algorithm reduced the running time in complex environments. Li et al. [28] proposed a multipoint region attraction RRT (Mra-RRT*) algorithm,

which adopted a multipoint region attraction strategy to improve the convergence speed, while simultaneously calling the path storage function to store the path and generate a new path. Ubic spline interpolation was used to make the path smoother. Liu et al. [29] proposed the Goal-biased Bidirectional RRT algorithm based on curve smoothing, which combined the goal-biased strategy with bidirectional search, and applied a sixth-order Bézier curve for smoothing between two nodes. The algorithm was demonstrated to effectively reduce the search time. Wu et al. [30] proposed a BPIB-RRT* algorithm that guided two random trees toward each other by incorporating a proposed bidirectional biased sampling strategy and bidirectional potential field heuristics, and thereby reduced the number of invalid nodes in the random sampling process and improved the sampling efficiency of the algorithm. Palmieri and Arras [31] proposed a path planning algorithm that combines machine learning and RRT. The proposed approach replaces the original time metric with a nonlinear parameter model having a constant time inference. As a result, the algorithm can improve path smoothness while reducing planning time costs, particularly in less cluttered environments.

This work extends past accomplishments in global path planning algorithms by proposing a wavefront-based path planning algorithm for obstacle avoidance scenarios involving low obstacle densities. The algorithm applies the principle of wavefronts to update the state lattice. In the first stage of the algorithm, a multilevel cost grid is used to initialize each state lattice and calculate its total cost. The process of calculating the total cost of the state lattice where the obstacle is located is simplified by dividing the total cost into intrinsic cost and steering cost values, where the intrinsic cost is determined according to whether that position can accommodate robot occupancy and the steering cost denotes the cost of a grid point corresponding to the steering angle of the mobile robot. At the same time, a logarithmic obstacle expansion function is employed to avoid the possibility of collision between the robot and environmental obstacles and to ensure that a path is generated that meets the required kinematic conditions associated with the actual maneuverability of mobile robots. The algorithm sets multiple target points that provide alternative target points if the original target point is occupied by an obstacle. The characteristics of the algorithm ensure that the additional computational complexity and space complexity associated with multiple target points is very low compared to generating an obstacle avoidance trajectory based on a single target point. The second stage of the algorithm seeks to avoid generating redundant paths and optimize the generated path points by smoothing the generated path based on two different Bézier curve approximations. Finally, the performance of the proposed algorithm is compared with those of three other state-of-the-art path planning algorithms based on simulations involving four different obstacle environments of varying complexity. The results demonstrate that the proposed generalized wavefront algorithm reduces the path length and the computation time to an extent suitable for practical mobile robot applications. At the same time, the generated path is smooth and reliable, making it suitable for facilitating the safe autonomous maneuverability of robots.

2. Problem Description

2.1. Environmental Map Representation. Environmental maps are usually described in one of the three ways: grid maps, geometric maps, and topological maps [32]. However, grid maps are most commonly adopted for describing environmental information in robot path planning applications because they are simple, effective, and easily implemented [33]. This study adopts a grid map for describing the environmental state around the robot, where the side length of a grid is $0.2a$. The intrinsic cost of a grid position (x, y) , denoted as $\text{map}(x, y).I$, is determined according to whether that position can accommodate robot occupancy and is divided into three states based on the environmental information as follows:

$$\text{map}(x, y).I = \begin{cases} = 0, \\ = 2000, \\ 0 < \text{map}(x, y).I < 2000. \end{cases} \quad (1)$$

Here, a value $\text{map}(x, y).I = 0$ indicates that grid position (x, y) is idle (i.e., open or obstacle free) and can accommodate robot occupancy. A value $\text{map}(x, y).I = 2000$ indicates that grid position (x, y) is occupied by an obstacle or represents a grid point reserved for avoiding collision with obstacles and therefore cannot accommodate robot occupancy. A value $0 < \text{map}(x, y).I < 2000$ indicates that the grid point resides at or nearby an obstacle and therefore must be subjected to special treatment to avoid collision between the obstacle and the robot. This is discussed in detail later.

2.2. Grid Distance Representation. In this study, the A^* algorithm and the generalized wavefront algorithm adopt the grid search method based on 8 nearest neighbor grids. The starting point of each search is the current grid center point, and the end point is an adjacent grid center point. The distance between grid points is divided into the two cases illustrated in Figures 1(a) and 1(b), where the black dots represent the current position of each search and the red lines denote the search directions. The search in Figure 1 is applied to adjacent grid points to the left and right and top and bottom of the current point, and the grid distance is set to $0.2a$. Meanwhile, the search in Figure 2 is applied to adjacent grid points along diagonal lines, and the grid distance is $0.2 * 1.414 * a$ ($0.283a$).

3. Generalized Wavefront Algorithm

The generalized wavefront algorithm uses a multilevel cost grid map to assign values to each grid point and sets up multiple target points to ensure that the original target point grid is not occupied by one or more obstacles. Then, a logarithmic function is employed to expand the number of reserved grid points around an obstacle to avoid robot collision, and the obtained path is smoothed based on a Bézier curve. These components of the algorithm are introduced as follows.

3.1. Generalized Wavefront Algorithm Search. The search principle of the generalized wavefront algorithm is illustrated in Figure 2. Here, the red position is the current grid point, and the total generated values of the eight neighboring grid points in the surrounding eight directions are compared. The grid with the lowest total generated value is selected as the next grid point to be expanded. This process is repeated until the expansion has progressed to one of the grid points of the target state set denoted by the blue circle in the figure.

3.2. Multilevel Cost Grid Map. The multilevel cost grid map consists of three levels: an intrinsic value grid layer, a steering value grid layer, and a total value grid layer. The intrinsic value grid layer stores the values of $\text{map}(x, y).I$. The steering value grid layer stores the cost value of a grid point corresponding to the steering angle of the mobile robot and thereby seeks to maintain uniform robot trajectories by penalizing pathways requiring large steering angles. This is illustrated in Figure 3, where the black circle is the starting point and the blue circle is the end point. The steering angle of the mobile robot in grid points lying along the direct line from the starting point to the target point is 0° , and the steering angle of the robot gradually increases as the grid position moves away from the direct path line on both sides. Therefore, the steering angle ranges from -180° to 180° . The corresponding steering value ranges from 0 to 180, where the absolute values of steering angles are employed. The total value grid layer is used to store the total cost of the multilevel cost grid map, where the total cost of the starting point is set as 1 while that of the end point is set as 2. The total generation value consists of the sum of the original value, the steering value, and the distance value.

3.3. Multiple Target Point Set. Conventional obstacle avoidance algorithms select only a single target point. As such, the path cannot be replanned if an obstacle lies in the target grid. The generalized wavefront algorithm avoids this disadvantage by taking the original target point as the center of a circle and selects the four points of symmetry as extended target points on the circle with the specific detection distance as the radius. These five target points together constitute the target point set. When the original target grid point is occupied by an obstacle or is a reserved grid point, the other four target points in the target point set are employed as end points to obtain four different obstacle avoidance paths, and the best obstacle avoidance path is selected from this set according to a comparison of the total generation value. This avoids the unnecessary computational costs associated with reselecting a target point and planning a new path. This process is illustrated in Figures 4(a) and 4(b) for a target point free of obstacles and a target point containing an obstacle, respectively. In Figure 4(a), the target grid point is in an idle state, and the original target point is preferentially selected to generate the optimal obstacle avoidance path indicated by the red line. However, in Figure 4(b), one of the other targets in the target point set is selected to generate the optimal obstacle avoidance path.

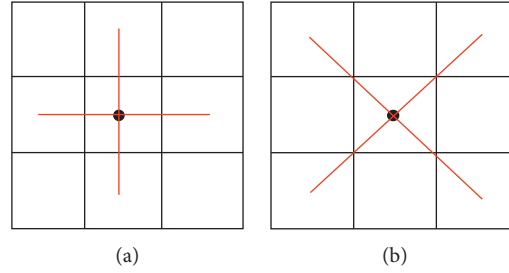


FIGURE 1: Grid search methods applied for searching 8 nearest neighbor grids: (a) first method and (b) second method.

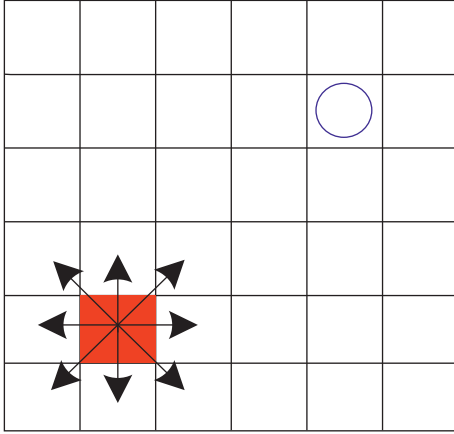


FIGURE 2: Grid expansion update process conducted according to the generation values of the eight points neighboring the current grid point (in red).

3.4. Obstacle Expansion. The generalized wavefront algorithm conducting the obstacle expansion stage is divided into two substages: first, a reserve expansion is conducted, like that conducted by a conventional obstacle expansion algorithm. Here, a reserved grid is generated around the obstacle in accordance with the shape and size of the robot. In the present work, the reserved grid occupies three grid points around the outside of the obstacle. After conducting the reserve expansion, the mobile robot can be regarded as a particle. Then, logarithmic expansion is applied to all grid points other than the reserved and obstacle grid points according to the following obstacle expansion function:

$$v = \frac{300 \times \log_{10} 3 - \log_{10}(m - 3)}{\log_{10}(2)}, \quad 4 \leq m \leq 6. \quad (2)$$

This expansion process is illustrated in Figure 5, where the horizontal axis represents the integer grid distance from the grid point to the edge of the obstacle grid, which includes the three units of the reserved grid, and the vertical axis represents the value of map $(x, y).I$ for the grid point. The default value is 2000.

3.5. Path Optimization. The path obtained by the generalized wavefront algorithm can be unnecessarily long and is not sufficiently smooth to conform to the actual maneuverability of the robot. Therefore, path length optimization and path smoothing are conducted as follows.

3.5.1. Path Point Sequence Optimization. The path length is optimized based on path sequence point optimization. First, all of the points in the original path that induce a change in slope, which are denoted herein as path points, are detected, a sequence of path points is generated in the order of the generated path, and the first three points of the sequence are omitted. The path points are selected in order, and an extension point is selected for each point in the sequence as a point that is not adjacent to the selected path point. The expansion process is conducted to determine whether a new path exists between the two path points. The criteria for selecting an alternative path are as follows: (1) no obstacle grid, obstacle expansion grid, or its logarithmic expansion grid exist between the two points; (2) the path between the two points is shorter than the original path length. If these criteria are met, the intermediate path point is removed to generate a new path. Otherwise, the original path is retained.

3.5.2. Key Point Selection. The generalized wavefront algorithm outputs a set of connected points representing the planned path $\{P_0, P_1, P_2, \dots, P_5\}$, where P_0 is the starting point and P_5 is the end point.

(1) Key Point Selection without Path Point Sequence Optimization. The key point selection process is illustrated in Figure 6, which involves selecting some key points within this set that are suitable as end points for smoothing the planned path. Here, the yellow and red circles represent the respective starting and end grid points, and the black circles represent the selected key points along the planned path. The key point selection process is conducted according to the following steps:

- Step 1: detect grid points on the planned path at which the slope of the path changes
- Step 2: two path segments with the same slope are detected at the point where the slopes of two adjacent paths change
- Step 3: select 1/3 of the first path segment nearest to its end point, 1/3 of the third path segment nearest to its start point, and all corner points of the transitional segment connecting the two points
- Step 4: smooth the selected points, as discussed in the following subsection.

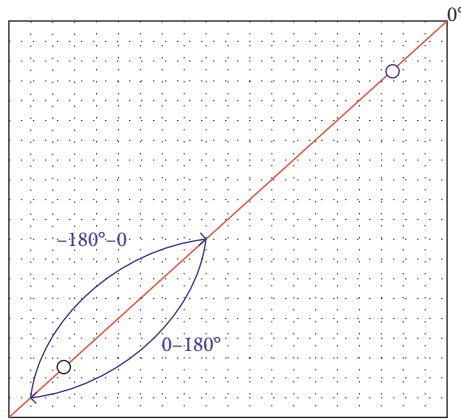


FIGURE 3: Description for the steering value.

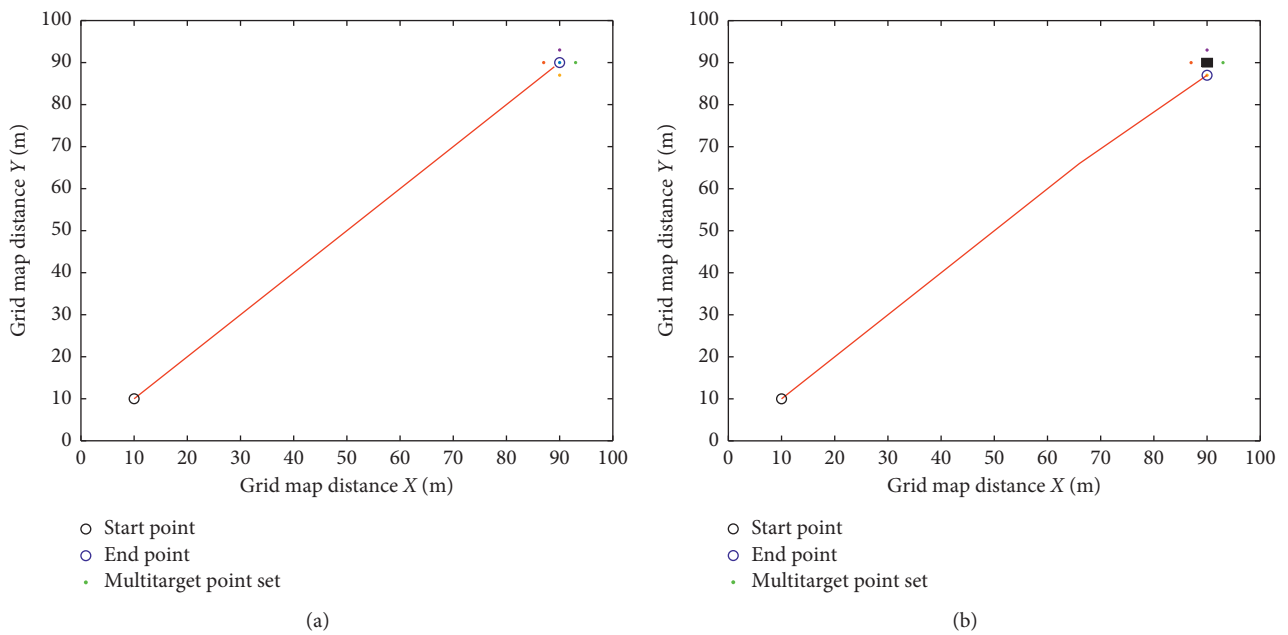


FIGURE 4: Optimal obstacle avoidance path generation for a target grid point free of obstacles (a) and a target grid point occupied by an obstacle (b).

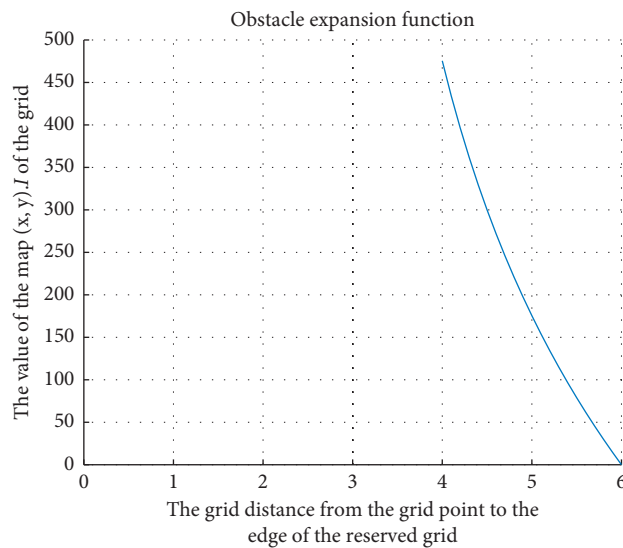


FIGURE 5: Logarithmic obstacle expansion function.

(2) *Key Point Selection Using Path Point Sequence Optimization*. If path point optimization results in the generation of a new path, key point selection is conducted as illustrated in Figure 7. Here, the yellow and red circles represent the respective start and end grid points, and the black circles represent the selected key points along the planned path. In this case, the key point selection process is conducted according to the following steps:

Step 1: detect new generated path points

Step 2: select 1/6 of the first path segment nearest to its end point, 1/6 of the second path segment nearest to its start point, and a corner point of the transitional segment connecting the two points

Step 3: smooth the selected points, as discussed in the following subsection.

3.5.3. *Bézier Curve Smoothing*. A Bézier curve is an approximation-based parametric curve [34]. A continuous contour for the given set $\{P_0, P_1, P_2, \dots, P_n\}$ can be approximately expressed as a smooth curve using the following n -th order polynomial Bézier curve approximation:

$$B(t) = \sum_{i=0}^n \binom{n}{i} P_i (1-t)^{n-i} t^i = \binom{n}{0} P_0 (1-t)^n t^0 + \binom{n}{1} P_1 (1-t)^{n-1} t + \dots + \binom{n}{n-1} P_{n-1} (1-t)^1 t^{n-1} + \binom{n}{n} P_n (1-t)^0 t^n, \quad t \in [0, 1]. \quad (3)$$

Here, t is a positional parameter with $t \in [0, 1]$.

(3) *Bézier Curve Path Smoothing*. Figure 8 presents a comparison of the original path in Figure 6 with the smoothed path. As can be seen from Figure 8, the smoothed path has no corner points compared to the original path and thereby improves the smoothness. Moreover, the smoothed path is very close to the original path, while remaining far from the obstacle because the path is processed by the expansion function and the selected key points are close to the corner points of the original path. Accordingly, the Bézier curve conforms to the actual maneuverability of a mobile robot in accordance with the original planned travel trajectory of the robot.

The Bézier curve smoothing process conducted for the path obtained by path point sequence optimization in Figure 7 is shown in Figure 9. Here, the three path points adjacent to the start point and the end point are smoothed.

The flowchart of the proposed generalized wavefront algorithm is shown in Figure 10.

4. Simulation Results

The feasibility and effectiveness of the proposed generalized wavefront algorithm were validated by comparing its path

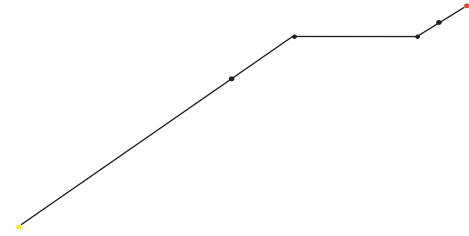


FIGURE 6: Selection of key grid points (black circles) along a planned path from a starting point (yellow circle) to an end point (red circle).

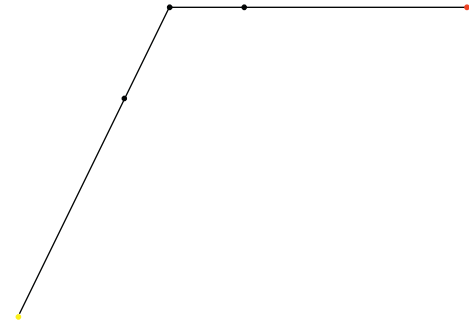


FIGURE 7: Selection of key grid points (black circles) along a planned path from a starting point (yellow circle) to an end point (red circle).

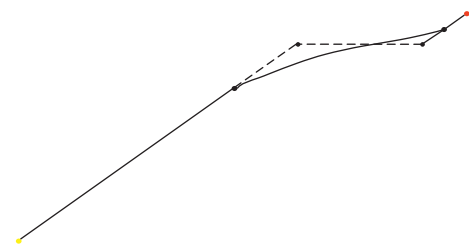


FIGURE 8: The original planned path presented in Figure 6 (dashed line) overlaid with the final planned path trajectory obtained as a Bézier curve (solid line).

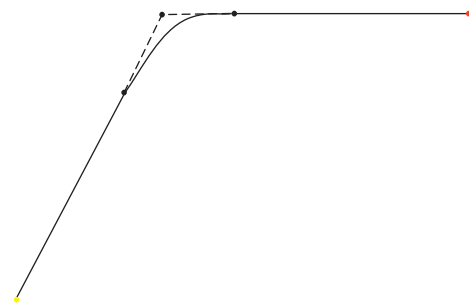


FIGURE 9: The original planned path presented in Figure 7 (dashed line) based on path point sequence optimization overlaid with the final planned path trajectory obtained as a Bézier curve (solid line).

planning performance with those obtained using the A^* algorithm and the RRT algorithm optimized using a Bézier curve via simulations conducted using a grid map with four different obstacle environments involving 1, 2, or many

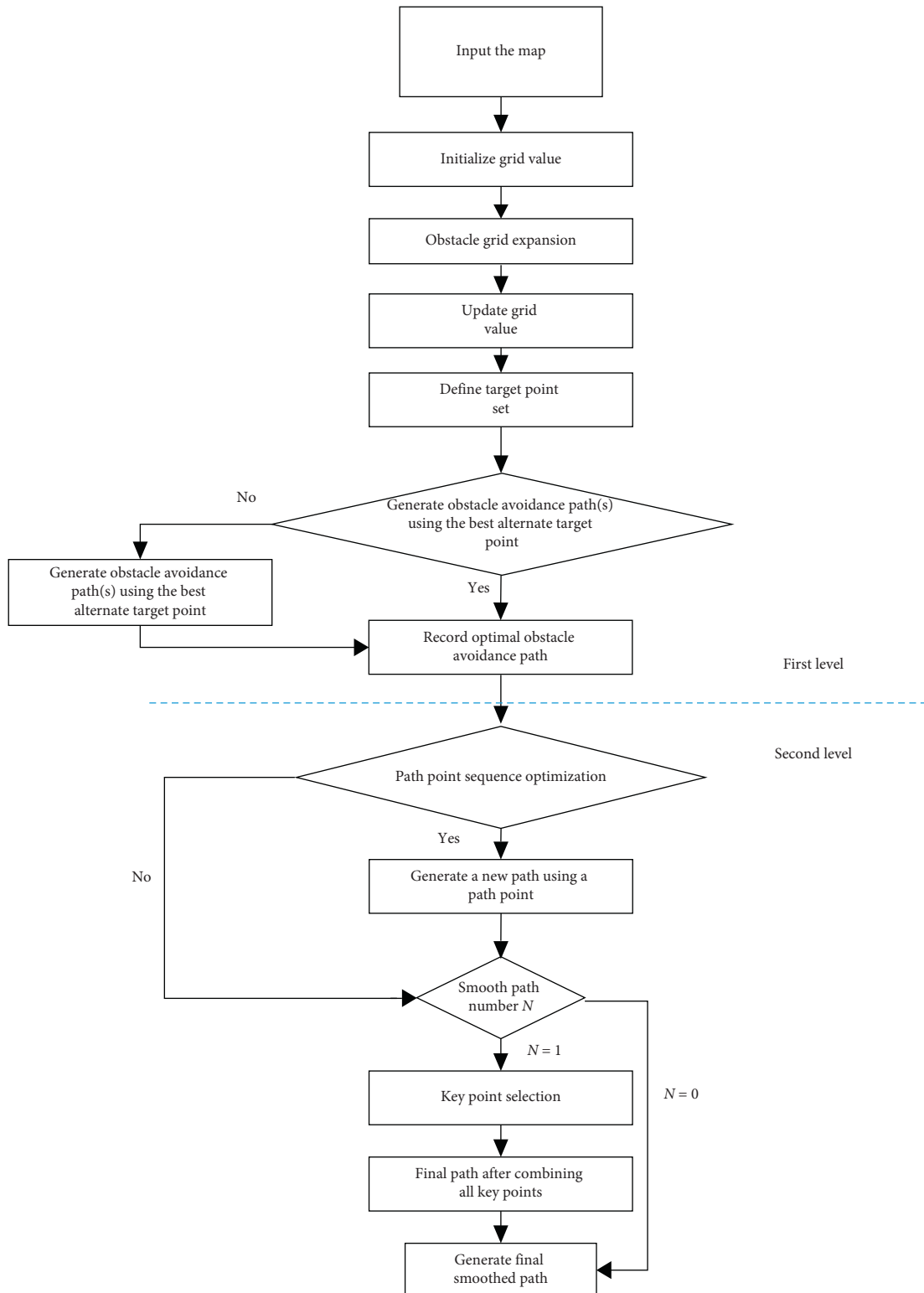


FIGURE 10: Flowchart of the generalized wavefront algorithm.

obstacles of varying shapes and sizes. Among them, Figures 11–13 compare path quality in terms of smoothness and distance from obstacles, while Figure 14 compares path quality in terms of smoothness and distance from obstacles, as well as path length, and planning algorithm runtime as

evaluation indicators. The experimental map is 100×100 grid points, in which the robot begins at (10, 10) and the end position is (90, 90). Because the RRT algorithm provides different results each time the algorithm is applied to a given environment, only one of the generated path results

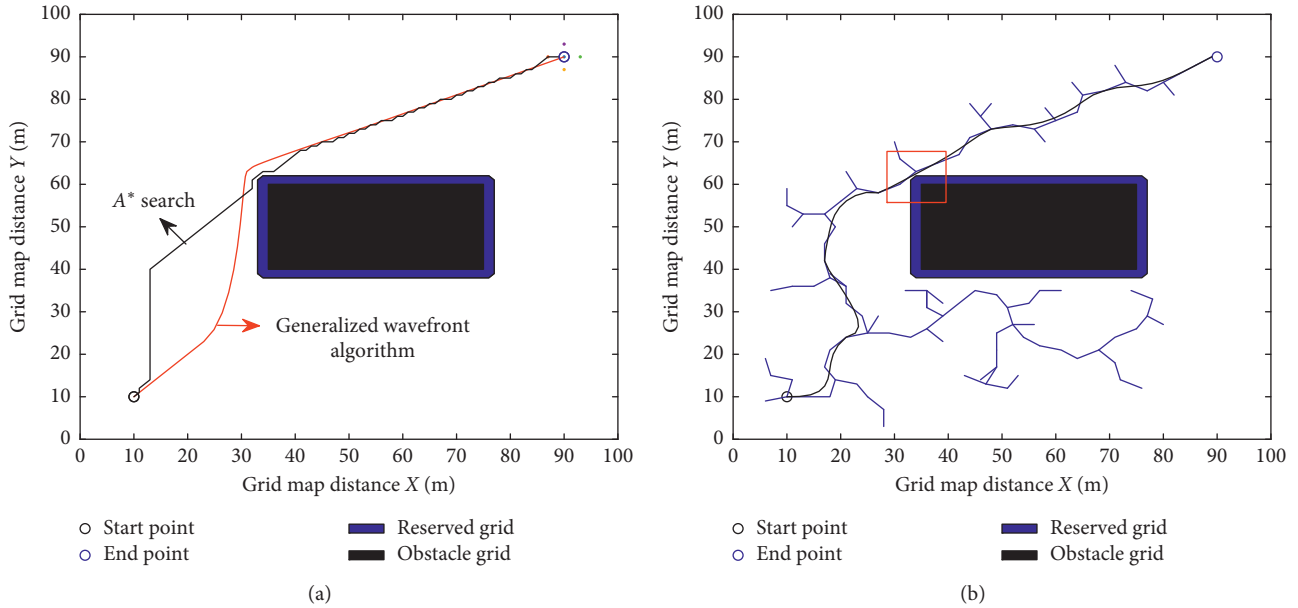


FIGURE 11: Planning results for a single-obstacle grid map: (a) planning results of the A^* algorithm and the proposed algorithm; (b) planning result of the optimized RRT algorithm.

are presented, and average path length and planning algorithm runtime values were obtained over 10 trials. The experimental platform employed MATLAB R2016a operated on a personal computer with an i5-6300HQ CPU, 2.3 GHz clock frequency, and 16.0 GB of RAM operating Windows 10.

4.1. Comparison of Path Planning Results. Figure 11(a) presents a comparison of the path planning results obtained for a single obstacle using the proposed algorithm and the A^* algorithm, while Figure 11(b) presents the path planning results obtained using the optimized RRT algorithm. The figures indicate that the proposed algorithm performs significantly better than the A^* and RRT algorithms. The path generated by the proposed algorithm is smoother, while the other paths pass too closely to the obstacle, which is particularly the case near the corners of the obstacle and for the path generated by the optimized RRT algorithm. As such, these paths are not safe for mobile robots. However, the path generated by the proposed algorithm resides far away from the obstacle, making the generated path safer and more suitable for mobile robots. Figures 12(a) and 13(a) present a comparison of the path planning results obtained for two obstacles using the proposed algorithm and the A^* algorithm, while Figures 12(b) and 13(b) present the path planning results obtained using the optimized RRT algorithm. As was observed for the single-obstacle environment, the path generated by the proposed algorithm is smoother, while the other paths pass too closely to one or both obstacles, especially near the corners. Again, this is particularly the case for the path generated by the optimized RRT algorithm, which approaches dangerously close to the obstacles, as indicated by the areas outlined by red boxes in Figures 12(b) and 13(b).

While the algorithm proposed in this paper reduces the path length, it also clearly provides a path that avoids obstacles on either side of a narrow corridor. These results demonstrate that the proposed algorithm successfully avoids the corners of rectangular obstacles and reduces unnecessary turning operations compared to the paths generated by the other two algorithms. Therefore, the paths generated by the proposed algorithm are better suited to the actual maneuverability of mobile robots.

4.2. Comparison of Path Planning Time and Path Length in a Complex Environment. Figure 14(a) presents a comparison of the path planning results obtained for a complex environment using the proposed algorithm and the A^* algorithm, while Figure 14(b) presents the path planning results obtained using the optimized RRT algorithm. As discussed above, the path generated by the proposed algorithm is smoother and passes much farther from obstacles. In this respect, the disadvantages of the A^* path planning algorithm are particularly evident, in that the generated path tends to adhere very closely to the boundaries of one obstacle or another while traversing the scene from the start point to the target point. The path length and planning time results of the three algorithms are listed in Table 1 and illustrated in Figure 15. While the optimized RRT algorithm requires considerably less planning time than the other planning algorithms, it obtains a much longer path that passes dangerously close to obstacles, as indicated by the areas in the red boxes (Figure 14(b)), and fails to suitably accommodate the actual maneuverability of robots. Therefore, the optimized RRT algorithm is not feasible and safe for the path planning process. Compared with the A^* algorithm, the proposed algorithm provides both shorter path length and planning time.

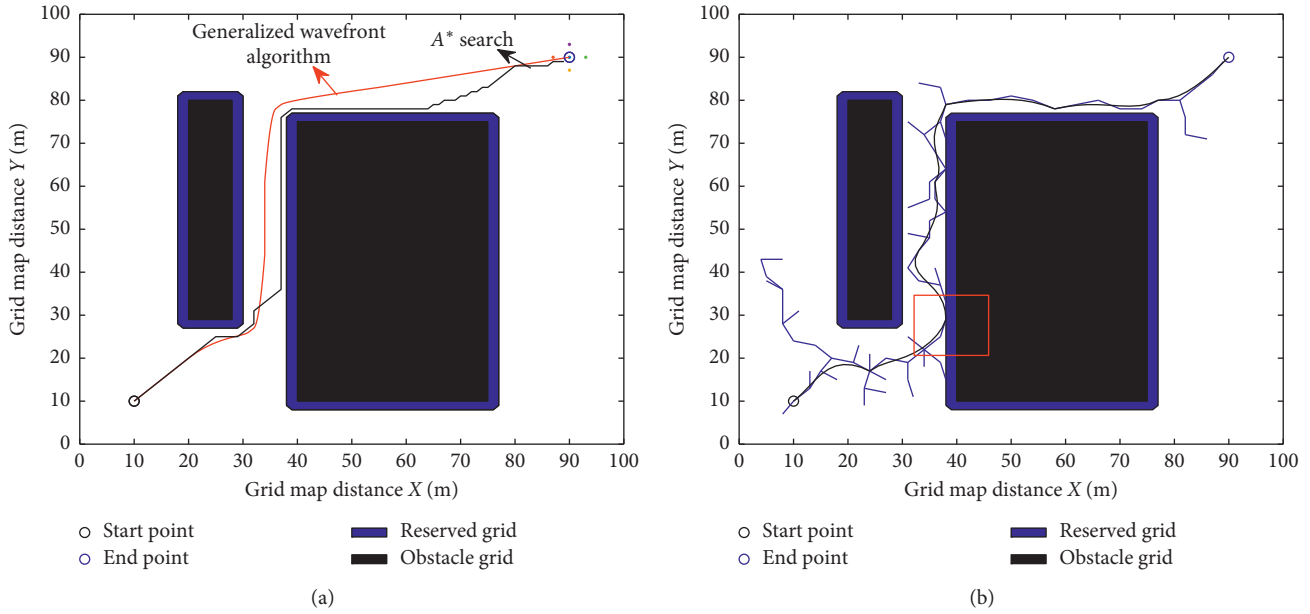


FIGURE 12: Planning results for a two-obstacle grid map: (a) planning results of the A* algorithm and the proposed algorithm; (b) planning result of the optimized RRT algorithm.

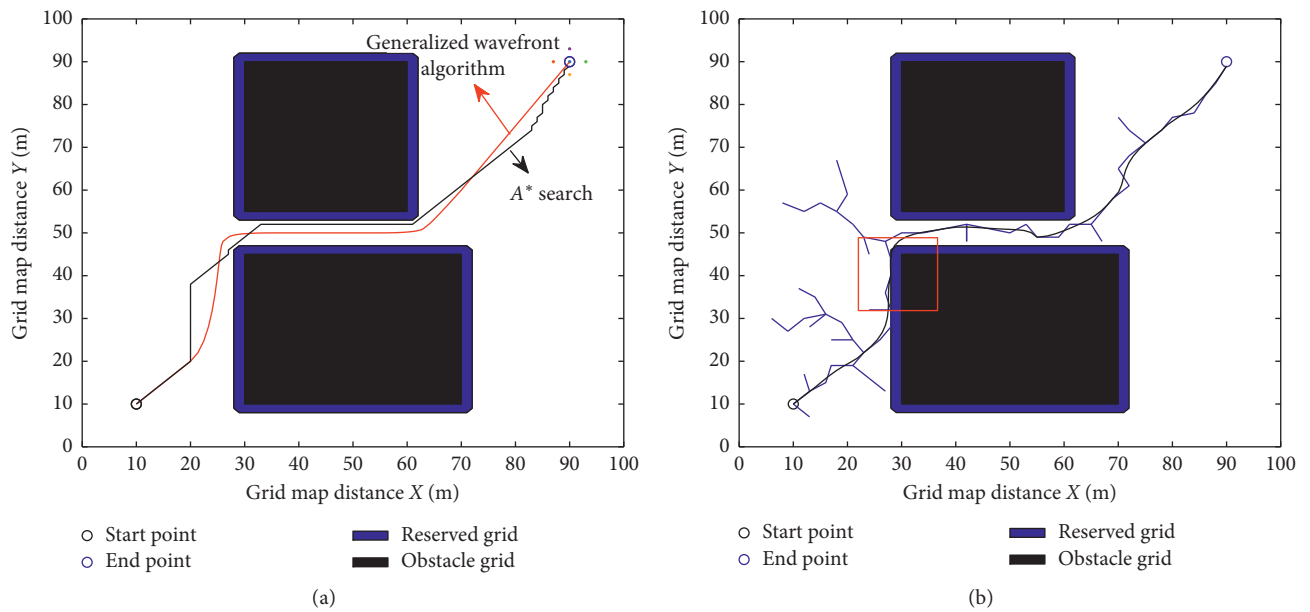


FIGURE 13: Planning results for a two-obstacle grid map: (a) planning results of the A* algorithm and the proposed algorithm; (b) planning result of the optimized RRT algorithm.

5. Discussion

The results in Table 1 indicate that the application of the generalized wavefront algorithm provides a path length that is 2.953 m shorter and a planning time that is 0.677 s less than those obtained by the A* algorithm. The A* algorithm greatly increases the planning time and path length due to its heuristic search method. In addition, while the planning time of the generalized wavefront algorithm is similar to that of the optimized RRT algorithm, the path

length obtained by the proposed algorithm is 3.964 m shorter because the optimized RRT algorithm employs a completely random search method. This results in an increase in the number of redundant nodes and an insensitivity to the environment, which greatly increases the path length obtained by the optimized RRT algorithm and decreases path reliability.

The logarithmic expansion algorithm employed for accommodating obstacles by the generalized wavefront algorithm generates paths through a narrow space around

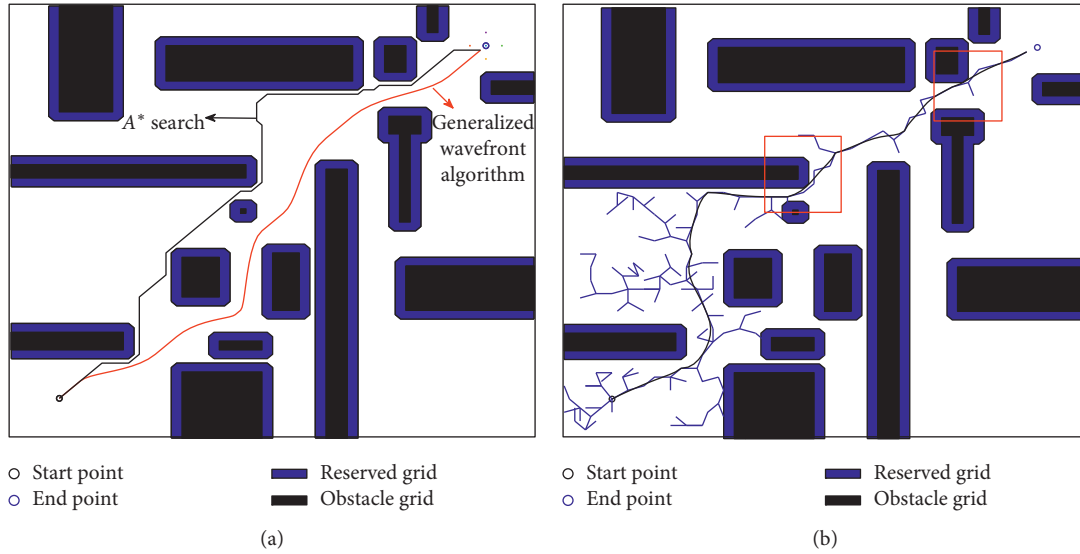


FIGURE 14: Planning results in a complex grid map: (a) planning results of the A* algorithm and the proposed algorithm; (b) planning result of the optimized RRT algorithm.

TABLE 1: Path lengths and algorithm planning times of the three planning algorithms for a complex environment.

Algorithm	Path length (m)	Planning time (s)
Generalized wavefront	23.892	1.008
A*	26.845	1.685
Optimized RRT	27.85	0.857

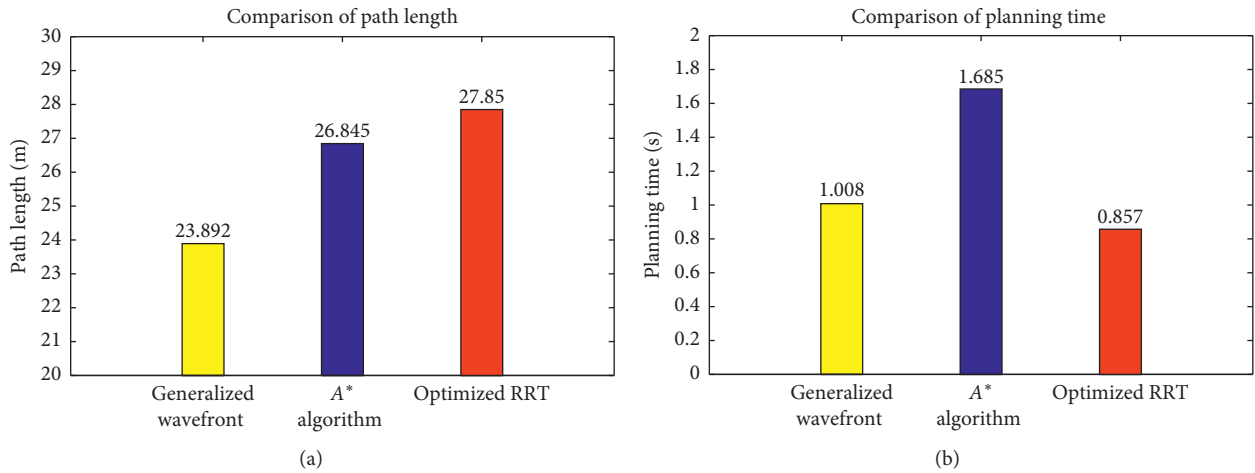


FIGURE 15: Comparison results in a complex grid map: (a) path lengths of the three planning algorithms for a complex environment; (b) planning times of the three planning algorithms for a complex environment.

obstacles while maintaining a specified distance from obstacles. As a result, the algorithm can be safely employed under conditions of narrow pathways. Sequence point optimization facilitates the merging of nonadjacent sequence points between two obstacles into two adjacent sequence points, so that the generation of redundant paths is greatly reduced. Accordingly, the resulting path lengths are greatly reduced. At the same time, the path is smoothed using Bézier curves, so that the generated curve conforms to the kinematic characteristics of robots at inflection points. These

features also provide the proposed algorithm with a measure of adaptability in multiobstacle environments. Finally, when the end point is occupied, the algorithm can choose other target points for planning, which effectively improves the planning efficiency.

One limitation of the generalized wavefront algorithm is that a large number of sequence points (knee points) may be generated when encountering multiple complex environments, which can increase the processing time. The processing time under these conditions can be greatly reduced

by adding a selector to filter the sequence points. This will be addressed in future work.

6. Conclusion

This study presented a generalized wavefront algorithm. In the first stage, a multilevel cost grid map, logarithmic expansion, and multiple target point sets were used to generate an optimum obstacle avoidance trajectory. In the second stage, path point sequences were optimized and key points in the obtained trajectory were selected for conducting smoothing based on a Bézier curve, thereby generating a smooth path that conforms to the kinematic requirements of mobile robots. Finally, simulation experiments were conducted using MATLAB, and comparisons were made with the performances of other planning algorithms. The simulation results demonstrated the algorithm proposed in this study meets the required kinematic conditions associated with the actual maneuverability of mobile robots, and the generated path is both shorter and safer and requires less computation time in the planning process. As such, the approach is not only reasonable and effective but also more reliable. The required computation time during path planning is acceptable for real-time operation, making the proposed algorithm suitable for practical applications. In future work, the performance of the proposed approach will be verified by experimental testing with an actual mobile robot.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors have no conflicts of interest to declare regarding the publication of this paper.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (Grant no. 91420202).

References

- [1] D. K. Qu, Z. J. Du, D. G. Xu, and F. Xu, "Research on path planning method for mobile robots," *Robot*, vol. 30, no. 2, pp. 97–101, 2008.
- [2] D. J. Wang, "Path planning for indoor mobile robot based on improved A* algorithm," *Journal of Tsinghua University: Natural Science Edition*, vol. 52, no. 8, pp. 1085–1089, 2012.
- [3] K. Jeddisaravi, R. J. Alitappeh, L. C. A. Pimenta, and F. G. Guimarães, "Multi-objective approach for robot motion planning in search tasks," *Applied Intelligence*, vol. 45, no. 2, pp. 305–321, 2016.
- [4] R. A. Brooks, "Solving the find-path problem by good representation of free space," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 2, pp. 190–197, 1983.
- [5] R. A. Brooks and T. Lozano-Perez, "A subdivision algorithm in configuration space for findpath with rotation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 2, pp. 224–233, 1985.
- [6] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [7] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 224–241, 1992.
- [8] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.
- [9] H. Choset, K. Lynch, and S. Hutchinson, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, The MIT Press, Boston, MA, USA, 2005.
- [10] S. M. LaValle, *Planning Algorithms*, Cambridge University Press, New York, NY, USA, 2006.
- [11] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 635–646, 2010.
- [12] A. K. Pamosoaji and K.-S. Hong, "A path-planning algorithm using vector potential functions in triangular regions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 832–842, 2013.
- [13] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [14] M. Pivtoraiko and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2010.
- [15] A. Kushleyev and M. Likhachev, "Time-bounded lattice for efficient planning in dynamic environments," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pp. 747–754, Kobe, Japan, June 2009.
- [16] M. Pivtoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," *International Symposium on Artificial Intelligence Robotics & Automation in Space*, vol. 603, no. 3, pp. 214–221, 2005.
- [17] A. K. Guruj, H. Agarwal, and D. K. Parsediya, "Time-efficient A* algorithm for robot path planning," *Procedia Technology*, vol. 23, no. 2, pp. 144–149, 2016.
- [18] Y. Li, H. Zhang, H. Zhu, J. Li, W. Yan, and Y. Wu, "IBAS: Index based A-star," *IEEE Access*, vol. 6, pp. 11707–11715, 2018.
- [19] L. Zuo, Q. Guo, X. Xu, and H. Fu, "A hierarchical path planning approach based on A* and least-squares policy iteration for mobile robots," *Neurocomputing*, vol. 170, pp. 257–266, 2015.
- [20] W. Sun, Y. F. Lu, H. W. Tang, and M. Xue, "Path rules for mobile robots based on an improved A* algorithm," *Journal of Hunan University (Natural Science Edition)*, vol. 44, no. 4, pp. 94–101, 2017.
- [21] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: progress and prospects," *Algorithmic and Computational Robotics: New Directions*, vol. 60, no. 3, pp. 293–308, 2011.
- [22] J. J. Kuffner and S. M. LaValle, "RRT-connect: an efficient approach to single-query path planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 995–1001, San Francisco, CA, USA, September 2000.
- [23] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for UAVs using rapidly-exploring

- random trees,” *Journal of Intelligent & Robotic Systems*, vol. 71, no. 2, pp. 231–253, 2013.
- [24] L. Jaillet and J. M. Porta, “Path planning under kinematic constraints by rapidly exploring manifolds,” *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 105–117, 2013.
- [25] A. M. Ladd and L. E. Kavraki, “Measure theoretic analysis of probabilistic path planning,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 229–242, 2004.
- [26] L. Jaillet, J. Hoffman, J. V. D. Berg, P. Abbeel, J. M. Porta, and K. Y. Goldberg, “EG-RRT: environment-guided random trees for kinodynamic motion planning with uncertainty and obstacles,” in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 2646–2652, San Francisco, CA, USA, July 2011.
- [27] D. Ghosh, G. Nandakumar, K. Narayanan, V. Honkote, and S. Sharma, “Kinematic constraints based Bi-directional RRT (KB-RRT) with parameterized trajectories for robot path planning in cluttered environment,” in *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, pp. 8627–8633, Montreal, Canada, May 2019.
- [28] X. Li, H. Jiang, W. Shi, S. Chen, and Y. Wang, “Path planning of multipoint region attraction RRT* algorithm in complex environment,” in *Proceedings of the 2019 Chinese Control Conference (CCC)*, pp. 4409–4414, Guangzhou, China, July 2019.
- [29] H. Liu, X. Zhang, J. Wen, R. Wang, and X. Chen, “Goal-biased bidirectional RRT based on curve-smoothing,” *IFAC-PapersOnLine*, vol. 52, no. 24, pp. 255–260, 2019.
- [30] X. Wu, L. Xu, R. Zhen, and X. Wu, “Biased sampling potentially guided intelligent bidirectional RRT* algorithm for UAV path planning in 3D environment,” *Mathematical Problems in Engineering*, vol. 2019, Article ID 5157403, 12 pages, 2019.
- [31] L. Palmieri and K. O. Arras, “Distance metric learning for RRT-based motion planning with constant-time inference,” in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 637–643, Seattle, WA, USA, October 2015.
- [32] X. Zhao and G. D. Hu, “Application of smooth AR A* algorithm in intelligent vehicle path planning,” *Mechanical Science and Technology*, vol. 36, no. 8, pp. 1272–1275, 2017.
- [33] X. Zhao, W. Wang, C. W. Huang, and Y. W. Zhao, “Path planning of mobile robot based on improved A* algorithm,” *Robot*, vol. 40, no. 6, pp. 903–910, 2018.
- [34] M. Duncan, “Applied geometry for computer graphics and CAD,” *Mathematical Gazette*, vol. 91, no. 520, 2007.