



OPEN ACCESS

EDITED BY

Yunchao Tang,
Dongguan University of Technology, China

REVIEWED BY

János Botzheim,
Eötvös Loránd University, Hungary
Weikuan Jia,
Shandong Normal University, China
Jinhai Wang,
Foshan University, China
Xing Wang,
Commonwealth Scientific and Industrial
Research Organisation (CSIRO), Australia

*CORRESPONDENCE

Lei Ye
[✉ leoye1992@sgu.edu.cn](mailto:leoye1992@sgu.edu.cn)

RECEIVED 13 November 2023

ACCEPTED 22 April 2024

PUBLISHED 13 May 2024

CITATION

Ye L, Li J and Li P (2024) Improving path
planning for mobile robots in complex
orchard environments: the continuous
bidirectional Quick-RRT* algorithm.
Front. Plant Sci. 15:1337638.
doi: 10.3389/fpls.2024.1337638

COPYRIGHT

© 2024 Ye, Li and Li. This is an open-access
article distributed under the terms of the
[Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/).
The use, distribution or reproduction in other
forums is permitted, provided the original
author(s) and the copyright owner(s) are
credited and that the original publication in
this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted
which does not comply with these terms.

Improving path planning for mobile robots in complex orchard environments: the continuous bidirectional Quick-RRT* algorithm

Lei Ye*, Jin Li and Pu Li

School of Intelligent Engineering, Shaoguan University, Shaoguan, China

Efficient obstacle-avoidance path planning is critical for orchards with numerous irregular obstacles. This paper presents a continuous bidirectional Quick-RRT* (CBQ-RRT*) algorithm based on the bidirectional RRT (Bi-RRT) and Quick-RRT* algorithms and proposes an expansion cost function that evaluates path smoothness and length to overcome the limitations of the Quick-RRT* algorithm for non-holonomic mobile robot applications. To improve the zigzag between dual trees caused by the dual-tree expansion of the Bi-RRT algorithm, CBQ-RRT* proposes the CreateConnectNode optimization method, which effectively solves the path smoothness problem at the junction of dual trees. Simulations conducted on the ROS platform showed that the CBQ-RRT* outperformed the unidirectional Quick-RRT* in terms of efficiency for various orchard layouts and terrain conditions. Compared to Bi-RRT*, CBQ-RRT* reduced the average path length and maximum heading angle by 8.5% and 21.7%, respectively. In addition, field tests confirmed the superior performance of the CBQ-RRT*, as evidenced by an average maximum path lateral error of 0.334 m, a significant improvement over Bi-RRT* and Quick-RRT*. These improvements demonstrate the effectiveness of the CBQ-RRT* in complex orchard environments.

KEYWORDS

path planning, mobile robots, orchard environments, rapidly exploring random tree, bidirectional expansion

1 Introduction

Mobile robots are garnering increasing attention among researchers as intelligent devices. Owing to their high degree of automation, mobile robots can improve productivity and provide various conveniences. Mobile robots are currently being utilized in various scenarios (Ferguson and Stentz, 2006; Zhao et al., 2014; Ljungqvist et al., 2019; Reid et al., 2020; Strader

et al., 2020; Buchanan et al., 2021; Ye et al., 2023). As agricultural automation research has gained widespread attention in recent years, mobile robots have begun to make their way into agricultural settings (Xiong et al., 2017; Blok et al., 2019; Ren et al., 2020). Khajepour et al. (2020) proposed a particle filter algorithm using a laser beam model and Kalman filter algorithm based on line detection for the positioning and navigation of mobile robots in orchards. Qiu and Li (2022) implemented a path-tracking control for orchard mobile robots using a PID algorithm. Yang et al. (2022) used a dataset of orchard road data to train semantic segmentation networks to extract navigation paths and achieve path tracking for orchard robots. However, the studies above focused on orchard robots operating in structured environments with more prominent road features, with less consideration for path planning and obstacle avoidance. In unstructured orchard scenes, it is difficult for mobile robots to make decisions by obvious features or scene markers to realize specific navigation movements. The mobile robot in the orchard needs to perceive the unstructured orchard environment and plan the movement path for the mobile robot in the navigation operation stably and efficiently through the constructed environment map or by obtaining the real-time perception feedback. Therefore, in the field of mobile robot navigation in orchards, path planning is a fundamental and crucial aspect. Path planning refers to the process by which a mobile robot identifies, within the configuration space, an obstacle-free path from its initial configuration to its target configuration. The path must simultaneously satisfy environmental constraints and the kinematic constraints of the mobile robot itself (Siciliano et al., 2008).

In recent years, a substantial amount of research has been dedicated to the path planning conundrum for orchard robots. Jiang et al. (2022) used 3D and 2D LiDAR to map greenhouse orchards and used the Dijkstra algorithm to plan global paths for mobile robots. Tian et al. (2023) proposed an ant colony optimization algorithm to solve multitarget waypoint planning for unmanned aerial vehicles in orchards. Zhang et al. (2023) introduced an improved artificial potential field algorithm for weed-removal robot path planning, improving the safety of robot automation operations. Zhang et al. (2022) conducted LiDAR scans of greenhouse maps and employed the probabilistic roadmap technique to calculate the paths between all target points, consequently establishing comprehensive global paths for navigating multiple operational points. In addition to general-purpose path planning algorithms, researchers have extensively researched path planning algorithms for special orchard target guidance based on the perception of the orchard scene. Liu et al. (2023) proposed a real-time orchard mapping method based on visual odometry-centered position estimation fused with a LiDAR camera and achieved real-time apple global localization using ORB-SLAM3, YOLO-V5, Euclidean clustering filtering, and the local point cloud sliding window method. Xiong et al. (2023) combined the VINS-RGBD SLAM framework with the semantic segmentation algorithm BiSeNetV1 to propose a real-time localization and semantic map reconstruction method for unstructured citrus orchards. Chen et al. (2021) presented a global map-construction method tailored to the nature of orchard-picking tasks using the ORB-SLAM3 algorithm. Chen et al. (2024) proposed a set of vision algorithms for moving

target estimation, real-time self-localization, and dynamic harvesting. In addition, a reliable coordination mechanism was established for continuous motion and harvesting behavior.

In large and complex orchard environments, efficiently determining the navigation paths for mobile robots is of concern (Ye et al., 2023). The rapid-exploration random tree (RRT) algorithm is a framework for path planning algorithms widely used by mobile robots. This algorithm iteratively generates feasible paths by incorporating points obtained from random sampling of the state space, providing probabilistic completeness (LaValle et al., 2001). The efficiency of RRT algorithms for path planning has been highlighted in recent studies. Huan et al. (2023) employed the RRT*-connect algorithm to plan the navigation path of a microrobot in a plant-vein environment. Combined with the characteristics of the Dijkstra algorithm for path optimization, Wencheng et al. (2020) improved the RRT algorithm to solve the path planning problem of orchard spraying robots. Castro et al. (2023) proposed an online adaptive path planning solution fusing RRT and deep reinforcement learning algorithms. Despite these advantages, the RRT has inherent limitations due to its random sampling nature, including (1) reliance on a global uniform random sampling strategy, which results in slower convergence due to the lack of guided search (Li et al., 2020); (2) path generation from random sampling and iteration, resulting in non-optimality (Liu et al., 2023); and (3) unsuitability of RRT-generated paths for scenarios involving mobile robot tracking motion because it overlooks kinematic constraints (Webb and Van Den Berg, 2013).

Researchers have refined the basic RRT algorithm to accommodate various application environments and overcome limitations. Kuffner and LaValle (2000) proposed Bi-RRT, a bidirectional search tree that initiates two parallel trees from the start and target states to expedite algorithm convergence to improve the convergence speed of the RRT algorithm. Ye et al. (2021) enhanced the Bi-RRT by incorporating a gravitational algorithm with an adaptive step and gravitational coefficient adjustment into the AtBi-RRT algorithm, which they applied to a picking robot, achieving improved search efficiency and the ability to navigate complex obstacle regions. Tahir et al. (2018) proposed a potentially guided intelligent bidirectional RRT* algorithm that significantly improves convergence speed and memory utilization. To advance the optimization of paths generated through iterative random sampling in the RRT algorithm, Karaman and Frazzoli (2011) introduced the RRT* algorithm that incorporates key processes such as ChooseParent and Rewire and proved its asymptotic optimality. As the number of samples increased, the algorithm gradually approached the optimal solution, with the path acquisition cost converging to the optimum. However, the convergence is slow, especially in complex environments. The informed RRT* algorithm was introduced to enhance convergence efficiency (Gammell et al., 2014). This algorithm restricts the sampling to an elliptical region that narrows as the path length decreases after finding a feasible path. This strategy preserves the probabilistic completeness and asymptotic optimality of RRT*, speeds up convergence, and improves solution quality. Furthermore, Nasir et al. (2013) developed RRT*-Smart to further

speed up convergence. The algorithm starts from the sampled nodes and seeks direct, unobstructed connections to ancestral nodes, reducing path costs. Similarly, Jeong et al. (2019) proposed Quick-RRT*, which extends the set of parent nodes beyond the immediate neighbors to include their ancestors. This approach also influences the rewiring process, favoring the alignment of new nodes with nearby parent nodes, improving initial solutions, and achieving faster optimal convergence compared to RRT. Li et al. (2020) combined a potentially guided method and Quick-RRT* to propose the PQ-RRT* algorithm, which achieved faster convergence and better initial path quality than the Quick-RRT*. Moreover, researchers have proposed two approaches to apply RRT algorithms to kinematically constrained navigation scenarios. The first is the consideration of kinematic constraints during the RRT path search process. Webb and Van Den Berg (2013) introduced kinodynamic-RRT*, an asymptotically optimal path planning solution for robots with linear dynamics. Moon and Chung (2014) proposed a dual-tree RRT algorithm for path planning for mobile robots with differential wheels. Wang et al. (2021) proposed the KB-RRT* algorithm, incorporating kinematic constraints to prevent unnecessary node expansion and speed up the feasible path identification for differential-wheel mobile robots. The second approach involves post-processing to mitigate increasing computational complexity during sampling. Methods such as spline fitting maintain the planning speed while producing smooth paths. Elbanhawi et al. (2015) proposed a cubic B-spline smoothing algorithm that ensured curvature continuity while satisfying a robot's incompleteness constraint. Other smoothing methods, such as Bezier curves, have also been widely used to smooth the initial RRT paths, making them suitable for different motion planning scenarios (Liu et al., 2019; Li and Yang, 2020; Gan et al., 2021).

The conventional directional extension algorithm research pays relatively little attention to the smooth connection of paths at the conjunction between two trees; in the RRT sampling algorithm, processing the dynamics constraint during each iteration will occupy a significant amount of computational resources and increase the planning time, especially in complex environments; in path planning research combining RRT algorithm with spline fitting, relatively few studies consider the corner constraints of path points, resulting in large deviations between the actual trajectory and planned path, which can easily trigger accidental collisions in complex environments. Therefore, applying the existing path planning methods for the path planning problem of orchard mobile robots in complex environments is relatively difficult. Based on the limitations of the existing studies, our primary contributions are presented below.

- (1) The path length and maximum heading angle of the path are simultaneously adopted as the path cost according to the mobile robot. Hence, the algorithm is optimized for path smoothness and improved in terms of path executability during the extension process.
- (2) The path planning speed is improved by adding a directional extension method to the Quick-RRT* algorithm.
- (3) To improve the local path smoothness of the bidirectional extension algorithm in conjunction with the two trees, the CreateConnectNode optimization algorithm is proposed to

generate the optimal connection point directly in the conjunction region of the two trees.

The remainder of this paper is organized as follows: *Section 2* discusses the specific implementation of the proposed continuous bidirectional Quick-RRT* (CBQ-RRT*) algorithm. Simulation comparisons and field experiments for B-RRT*, Quick-RRT*, and CBQ-RRT* are described in *Section 3*, and the results are compared and analyzed. Finally, *Section 4* presents the conclusions and discusses the study.

2 Methods

In developing path planning algorithms for the autonomous navigation of mobile robots in orchards, the primary considerations are the executability and efficiency of the paths planned by the algorithm and the computational efficiency of the algorithm. This study improves the Quick-RRT* algorithm and introduces a CBQ-RRT* algorithm to meet these requirements. The pseudocode for the CBQ-RRT* algorithm is presented in [Algorithm 1](#). Initially, an extended cost function containing the kinematic constraints of the mobile robot is established, which is applied to lines 4 and 11 in [Algorithm 1](#) to constrain the extension direction of the nodes. The optimal parent nodes are further selected for X_{new} and X_{new2} in lines 7 and 14 using this cost function. Then, a bidirectional augmentation method is introduced to accelerate path planning. Finally, an optimization strategy is proposed for connecting dual trees to avoid abrupt connections that violate the kinematic constraints (line 19). Further details of the implementation are presented in the following section.

```

1  $V_a \leftarrow \{X_{start}\}; V_b \leftarrow \{X_{goal}\}; E_a \leftarrow \emptyset; E_b \leftarrow \emptyset; T_a \leftarrow (V_a, E_a);$ 
 $T_b \leftarrow (V_b, E_b);$ 
2 for  $i \leftarrow 0$  to  $N$  do
3    $X_{rand} \leftarrow \text{Sample}();$ 
4    $X_{nearest} \leftarrow \text{NearestVertex}(X_{rand}, T_a);$ 
5    $X_{new} \leftarrow \text{Extend}(X_{nearest}, X_{rand});$ 
6    $\sigma_a \leftarrow \text{Steer}(X_{nearest}, X_{new});$ 
7    $X_{parent\_Ta} \leftarrow \text{ChooseParent}(X_{new}, \text{Depth});$ 
8   if  $X_{parent\_Ta} \neq \emptyset$  then
9      $T_a \leftarrow \text{Rewire}(T_a, X_{new}, X_{ancestries\_Ta})$ 
10  end if
11   $X_{nearest\_Tb} \leftarrow \text{NearestVertex}(X_{new}, T_b);$ 
12   $X_{new2} \leftarrow \text{Extend}(X_{nearest\_Tb}, X_{Tb\_Extend});$ 
13   $\sigma_b \leftarrow \text{Steer}(X_{nearest\_Tb}, X_{new2});$ 
14   $X_{parent\_Tb} \leftarrow \text{ChooseParent}(X_{new2}, \text{Depth});$ 
15  if  $X_{parent\_Tb} \neq \emptyset$  then
16     $T_b \leftarrow \text{Rewire}(T_b, X_{new2}, X_{ancestries\_Tb})$ 
17  end if
18  if  $\text{distance}(X_{new2}, X_{new}) \leq \text{GoalRadius};$ 
19     $X_{create\_conn} \leftarrow \text{CreateConnectNode}(T_a, T_b, X_{new});$ 
20  end if
21 end for

```

Algorithm 1. CBQ-RRT*.

2.1 Cost function of smooth expansion

In the application of mobile robots in orchards, the efficacy of the RRT and its extended variants in path planning has been suboptimal. The heading angles of the generated path nodes have large variations in the path planning phase because of the random nature of the algorithm, which challenges the kinematic constraints of the mobile robot. This leads to significant lateral deviation in the navigation of the robot, contributing to navigation failure. This section proposes a novel path cost function based on the analysis of the kinematic constraints of a mobile robot that mitigates this problem.

Assuming a uniform terrain, the spatial position of a tracked mobile robot within an orchard environment can be defined as $X_i = (x_i, y_i, \theta_i)$, as illustrated in Figure 1. Here, x_i and y_i represent the coordinates in the global x_w - y_w coordinate system, while θ_i signifies the robot's orientation angle. The computation of θ_i is done using Equation (1).

$$\theta_i = \arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) \quad (1)$$

Additionally, by neglecting any track deformation during the motion of the tracked mobile robots, we can approximate their kinematics as that of a differentially driven mobile robot. According to the kinematic model formula, the kinematic constraints can be articulated using Equation (2).

$$\begin{aligned} \begin{bmatrix} x_{i+1} \\ y_{i+1} \\ \theta_{i+1} \end{bmatrix} &= \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} + \begin{bmatrix} \int_0^{\Delta t} v_c \cos(\theta_i + \omega t) dt \\ \int_0^{\Delta t} v_c \sin(\theta_i + \omega t) dt \\ \int_0^{\Delta t} \omega dt \end{bmatrix} \\ &= \begin{cases} \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} + \begin{bmatrix} -R \sin \theta_i + R \sin(\theta_i + \omega \Delta t) \\ R \cos \theta_i - R \cos(\theta_i + \omega \Delta t) \\ \omega \Delta t \end{bmatrix} & \omega \neq 0 \\ \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} + \begin{bmatrix} v_c \Delta t \cos \theta_i \\ v_c \Delta t \sin \theta_i \\ 0 \end{bmatrix} & \omega = 0 \end{cases} \quad (2) \end{aligned}$$

where v_c and ω are the linear and angular velocities of the robot, respectively; R is the turning radius of the robot, and $R = \frac{v_c}{\omega}$.

From Equation (2), the transformation between the robot's current configurations depends on the kinematic parameter constraints. When the planned path includes points with significant changes in yaw angles, it can result in unreachable points along the path. Consequently, significant lateral deviation errors occur during the robot's trajectory tracking process, as shown in Figure 2, where $X_2 = (x_2, y_2, \theta_2)$ is the path point of the planned path and $X_2' = (x_2', y_2', \theta_2')$ is the path point reached by the mobile robot under the kinematic constraints.

Thus, significant changes in the heading angle at the waypoint result in considerable lateral error in the path-following trajectory of the mobile robot during navigation. This section introduces a novel path cost function, denoted as $C(x_i, x_{i+1})$, designed to regulate the

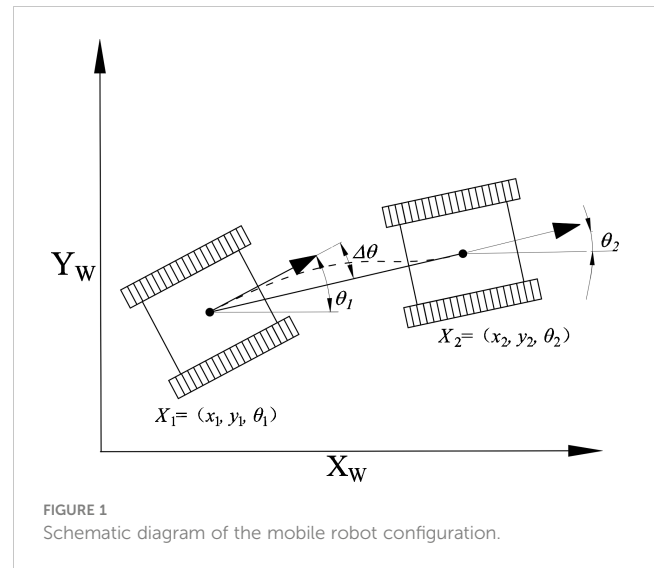


FIGURE 1 Schematic diagram of the mobile robot configuration.

expansion direction of the nodes, as shown in Equations (3) and (4).

$$C(X_i, X_{i+1}) = D(x_i, x_{i+1}) + A(X_i, X_{i+1}) \quad (3)$$

$$\begin{cases} D(X_i, X_{i+1}) = (X_{x_{-i}} - X_{x_{-i+1}})^2 + (X_{y_{-i}} - X_{y_{-i+1}})^2 \\ A(X_i, x_{i+1}) = \begin{cases} 0, D(X_i, X_{i+1}) + D(X_i, X_{i(\text{parent})}) < D(X_{i(\text{parent})}, X_{i+1}) \\ +\infty, \text{others} \end{cases} \end{cases} \quad (4)$$

where $X_{i(\text{parent})}$ denotes the parent node of X_i . The path cost function $C(X_i, X_{i+1})$ comprises a distance cost function $D(X_i, x_{i+1})$ and an angle penalty function $A(X_i, X_{i+1})$. The computation of $D(X_i, X_{i+1})$ involves using the Euclidean distance formula, whereas $A(X_i, X_{i+1})$ is calculated indirectly using the cosine formula. An angle cost penalty is introduced when the heading angle between path nodes exceeds $\pi/2$.

The CBQ-RRT* algorithm uses the cost function in lines 4–6 and 11–13 of Algorithm 1 to select the initial parents of X_{rand} and X_{new} in the two extended trees, ensuring that the paths between them and their respective parents satisfy the kinematic constraints. Thus, the CBQ-RRT* algorithm restricts the expansion direction of the nodes, ensuring compliance with the kinematic constraints of the mobile robot. Figure 3 illustrates the node expansion process. Additionally, the CBQ-RRT* algorithm incorporates the cost function into the ChooseParent and Rewire functions in lines 7–10 and 14–17 of Algorithm 1, which further maintains the path to satisfy the kinematic constraints during the path optimization process of the CBQ-RRT* algorithm. The pseudocode for implementing the ChooseParent and Rewire functions is shown in Algorithm 2 and Algorithm 3.

```

1  X_near ← NearVertices(X_new, T, R);
2  for X_near ∈ X_near do
3  X_ancestry ← X_near;
4  for i = 0 to Depth do
5  if X_ancestry ∉ X_ancestries;
6  X_ancestries ← X_ancestry ∪ X_ancestries;

```

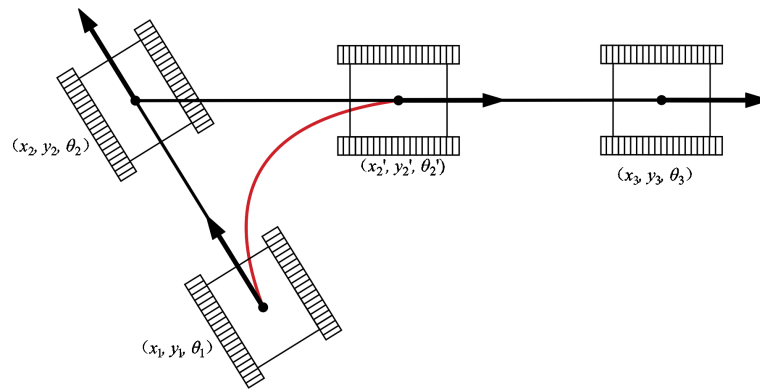


FIGURE 2 Schematic of mobile robot trajectory deviation.

```

7   end if
8   X_ancestry ← parent(X_ancestry);
9   end for
10  end for
11  X_parent_new ← findCost_min(X_ancestry, X_new)
12  Return X_parent_new
    
```

Algorithm 2. ChooseParent(X_{new} , Depth).

```

1  for all X_from ∈ X_new ∪ X_ancestries do
2    for all X_near ∈ X_near
3      if Cost(X_from) + Cost(X_from, X_near) ≤ Cost(X_near) &
CollisionFree then;
4        X_parent_near ← X_from;
5      end if
6    end for
    
```

Algorithm 3. Rewire(X_{new} , X_{near} , $X_{ancestries}$).

2.2 Dual-tree smooth connected method

Adding the Quick-RRT* algorithm to the bidirectional expansion method in Algorithm 1 improves the planning efficiency, but in the expansion direction; T_a expands in the X_{rand} direction and T_b expands in the X_{new} direction generated by T_a . Moreover, ChooseParent and Rewire optimization during the dual-tree expansion is performed independently, making it difficult to obtain the optimal connection point by random sampling in a complex environment. For example, in Figure 4, if the Quick-RRT* algorithm is added directly to the bidirectional expansion method, the iteration is stopped after the dual-tree connection. In addition, there is a higher probability of path points with large changes in the heading angle in the path connection, and a large redundant section of the path is added, as shown in Figure 4A. To enhance this situation, once a connection path is obtained between the two trees (line 18 in Algorithm 1), the CBQ-RRT* algorithm proposes the CreateConnectNode algorithm to generate a more optimal dual-tree connection point X_{creat_conn} (line 19 in Algorithm 1). The pseudocode implementation is displayed in Algorithm 4, and an example of the algorithm implementation is displayed in Figure 4B.

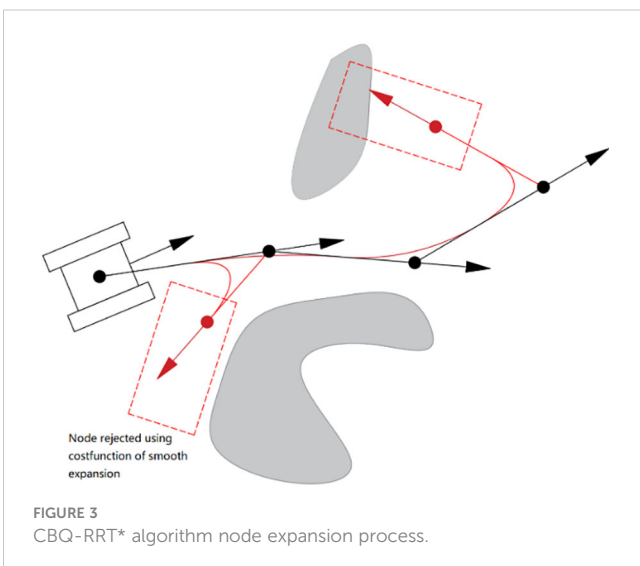


FIGURE 3 CBQ-RRT* algorithm node expansion process.

```

1  X_up ← X_conn
2  X_bottom ← X_parent-Ta
3  while distance(X_up, X_bottom) > dichotomy do
4  X_transition ← (X_up + X_bottom)/2;
5  if CollisionFree(X_transition, X_parent-x_new)
6    X_up ← X_transition;
7  else
8    X_bottom ← X_transition;
9  end if
10 end while
11 X_bottom ← X_parent-Tb;
12 while distance(X_up, X_bottom) > dichotomy do
13 X_creat_conn ← (X_up + X_bottom)/2;
14 if CollisionFree(X_creat_conn, X_parent-x_new)
15 X_up ← X_creat_conn;
16 else
    
```

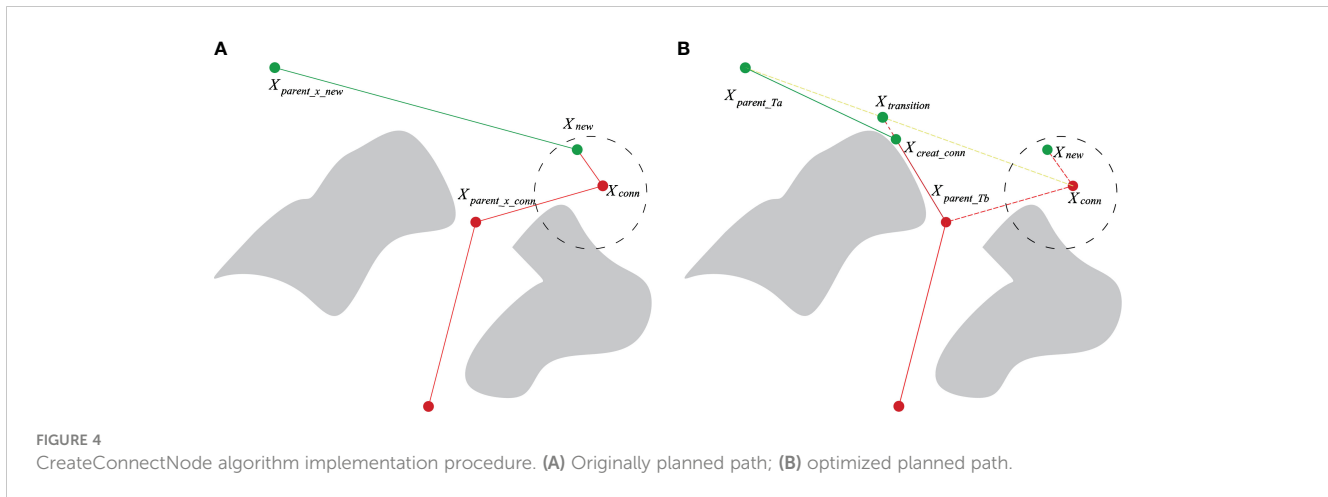


FIGURE 4
CreateConnectNode algorithm implementation procedure. (A) Originally planned path; (B) optimized planned path.

```

17    $X_{bottom} \leftarrow X_{creat\_conn}$ ;
18   end if
19 end while
20 return  $X_{creat\_conn}$ 

```

Algorithm 4. CreateConnectNode(T_a , T_b , X_{new}).

The CreateConnectNode algorithm operates as follows:

- (1) When the two expansion trees T_a and T_b are connected, the respective node sets $X_{ancestry_Ta}$ and $X_{ancestry_Tb}$ belonging to node X_{conn} are identified. The parent nodes X_{parent_Ta} and X_{parent_Tb} of X_{conn} are also located within this set.
- (2) Initially, the transition point $X_{transition}$ on the line linking X_{parent_Ta} and X_{conn} is ascertained. At the iteration onset, X_{parent_Ta} and X_{conn} are characterized as the lower and upper endpoints X_{bottom} and X_{top} , respectively. The midpoint between X_{bottom} and X_{top} is defined as X_{mid} . This is followed by connecting X_{parent_Tb} and X_{mid} . In the absence of a collision, X_{mid} is reassigned as X_{top} ; otherwise, X_{mid} is redefined as X_{bottom} . The iteration continues, and according to the dichotomy principle, the distance between X_{top} and X_{bottom} is progressively reduced until it falls below a pre-established dichotomy parameter. Using this approximation, $X_{transition}$ on the line connecting X_{parent_Ta} and X_{conn} is obtained. Applying the same principle, $X_{transition}$ and X_{parent_Tb} are designated as X_{up} and X_{bottom} respectively, with the optimal connection point X_{creat_conn} discovered on the line connecting $X_{transition}$ and X_{parent_Tb} .
- (3) Eventually, X_{creat_conn} serves as the link connecting the two expansion trees T_a and T_b , with X_{parent_Ta} and X_{parent_Tb} assigned as parent nodes, respectively, thereby combining the two trees and revealing the final path, as depicted in Figure 5B. The CreateConnectNode optimization algorithm produces an improved connection point, X_{creat_conn} , over the original. This resolves the smoothing issue encountered when connecting two expansion trees, thereby reducing both the trajectory's extensive angular change and its length.

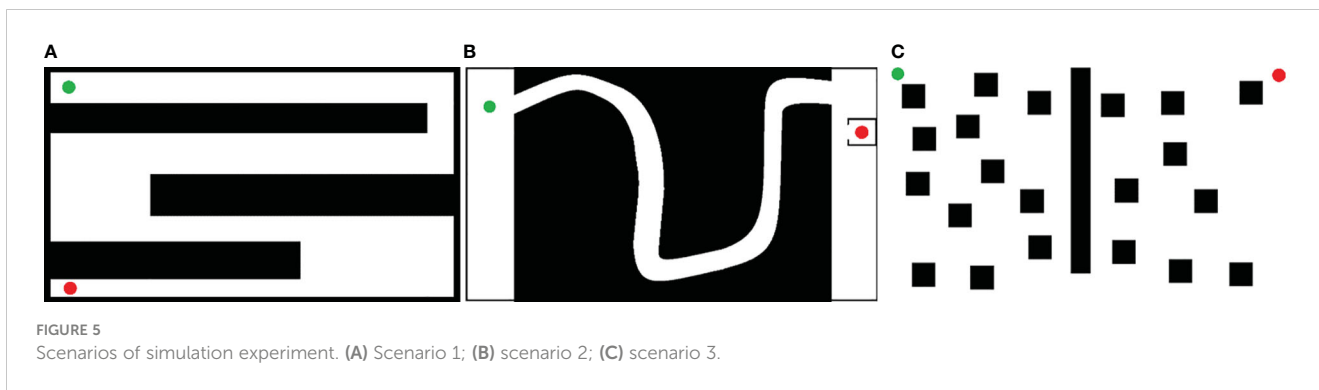
3 Simulations and experimentations

3.1 Path planning simulations

3.1.1 Simulation setting

To evaluate the performance of the proposed algorithm, three representative orchard scenarios (Figure 6) were used to construct the corresponding simulation maps (Figure 5). The white and black areas indicate the feasible and obstacle areas, respectively; the green and red points indicate the starting and target points of the scenario path plan, respectively. The maps have dimensions of 1,152×648 pixels and a resolution of 0.025 m/pixel. The starting and target configurations for all three scenarios are presented in Table 1. The simulation experiments were conducted on a computer powered by an Intel Core i5 processor with 16 GB of RAM utilizing the ROS-Melodic platform. For a comparative analysis, the proposed algorithm was pitted against the Bi-RRT* and Quick-RRT* algorithms, run with the depth parameter set to three during the simulation.

In the practical application of the orchard mobile robot navigation path planning, we focused on the executability, execution efficiency of the planned paths, and computational efficiency of the algorithm. Therefore, we systematically recorded the key parameters: the length and maximum steering angle of the algorithmically planned path, and the time taken by the algorithm for path formulation in the simulation. Path length serves as a central metric for evaluating the execution efficiency of an algorithmically planned path. A shorter path length directly corresponds to reduced time to execute the path, particularly for a given linear velocity of the mobile robot. Simultaneously, the maximum steering angle acts as an indicator of path smoothness. The conclusions drawn from the previous kinematic analysis indicate that a smaller maximum steering angle in the path correlates with a smaller lateral error generated by the mobile robot during path execution. The execution time of an algorithm is a direct indicator of its computational efficiency, which is related to its real-time responsiveness to dynamic scenarios.



In this section, each algorithm was executed 100 times in the three scenarios. The evaluation aimed to assess the efficiency of path planning, the feasibility of mobile robots in orchards, and overall algorithmic efficiency.

3.1.2 Analysis of simulation results

Examples of the simulation results are shown in Figures 7–9. The thin red lines indicate the randomly constructed trees, whereas the thick green lines indicate the final planned paths.

The path planning results highlight a notable reduction in the number of red lines for the Bi-RRT and CBQ-RRT* algorithms compared to the Quick-RRT* algorithm. This reduction implies that the Bi-RRT* and CBQ-RRT* algorithms can formulate planned paths with an expansion of a relatively modest number of nodes, indicating superior efficiency in path planning. Additionally, an examination of the final planned path, represented by the thick green line, reveals distinctive characteristics. The path of the Bi-RRT* algorithm at the dual-tree node, as indicated by the rectangular box, displays a noticeable curvature. In contrast, the path planned by the CBQ-

RRT* algorithm demonstrates remarkable smoothness, with a shorter length than that planned by the Bi-RRT* algorithm. This observation underscores the superior quality of the paths generated by the CBQ-RRT* algorithm compared to the Bi-RRT* algorithm. The simulation examples provided initially demonstrate that the CBQ-RRT* algorithm can achieve higher-quality path planning while maintaining superior algorithmic efficiency. More detailed simulation results and statistics are presented in Figure 10 and Tables 2–4.

First, Figure 10A and Table 2 demonstrate that the CBQ-RRT* algorithm consistently outperforms the Bi-RRT* and Quick-RRT* algorithms in generating more efficient paths across the three distinct environmental scenarios. During iteration, the CBQ-RRT* and Quick-RRT* algorithms enhance the potential parent node set in the ChooseParent and Rewire phases, finding shorter paths. Additionally, the CBQ-RRT* algorithm employs CreateConnectNode optimization, significantly improving path quality and yielding shorter paths than the Quick-RRT* algorithm, as shown in Table 2.

Second, unlike Bi-RRT* and Quick-RRT* algorithms, the CBQ-RRT* algorithm integrates a cost function that accounts for cornering constraints, significantly reducing sudden changes in path curvature. The CBQ-RRT* algorithm uses a smoothing connection technique in the bidirectional tree connection process to produce smoother paths and is further supported by the maximum path-turning angle data in Figure 10B and Table 3. Based on the constraints imposed by the robot kinematics, the smaller the maximum turning angle, the smaller the lateral deviation error of the mobile robot when executing the path.

TABLE 1 Parameters of the starting and target points of the simulation environment.

Scenarios	Starting configuration	Target configuration
1	[0 m, 0 m, 0°]	[0 m, -14 m, 180°]
2	[0 m, 0 m, 0°]	[3 m, 27 m, 90°]
3	[0 m, 0 m, 0°]	[26.5 m, -1.6 m, 0°]

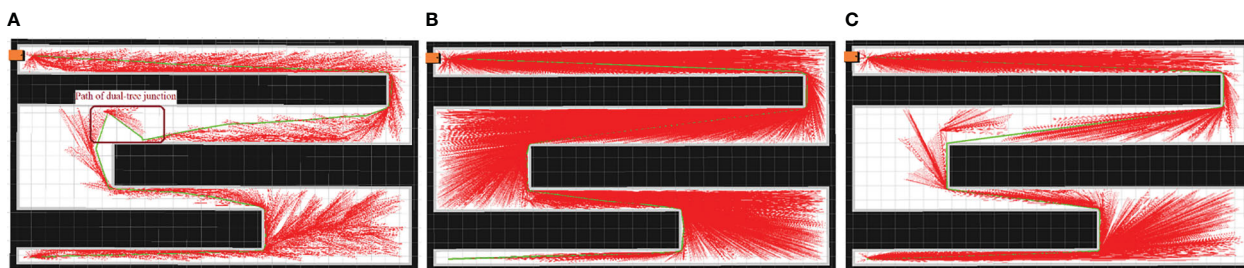


FIGURE 7 Example of simulation results for scenario 1. (A) Bi-RRT*; (B) Quick-RRT*; (C) CBQ-RRT*.

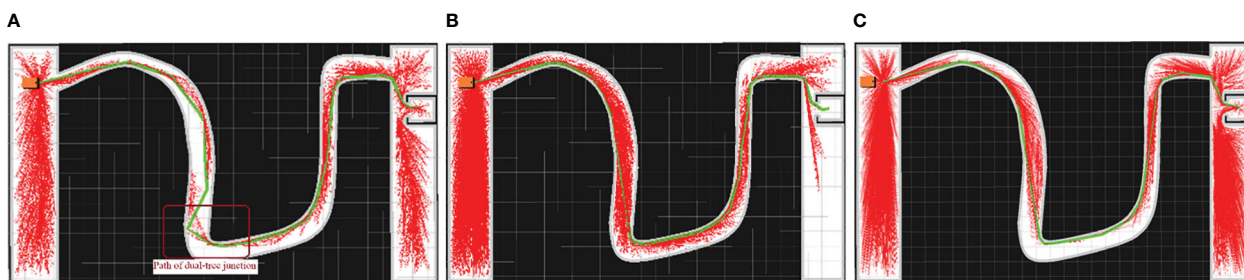


FIGURE 8 Example of simulation results for scenario 2. (A) Bi-RRT*; (B) Quick-RRT*; (C) CBQ-RRT*.

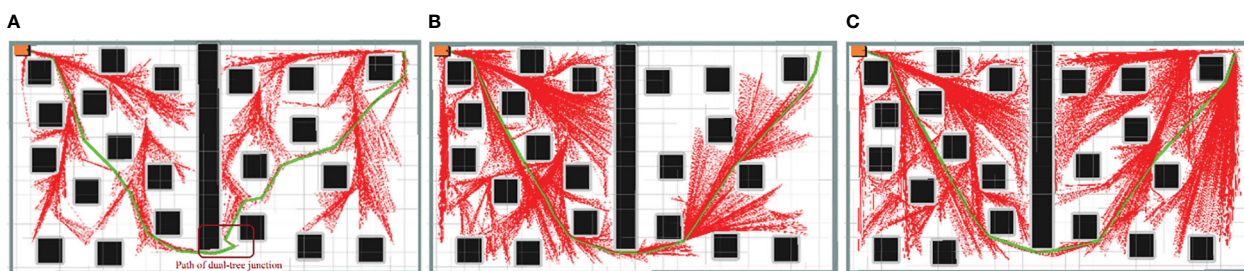


FIGURE 9 Example of simulation results for scenario 3. (A) Bi-RRT*; (B) Quick-RRT*; (C) CBQ-RRT*.

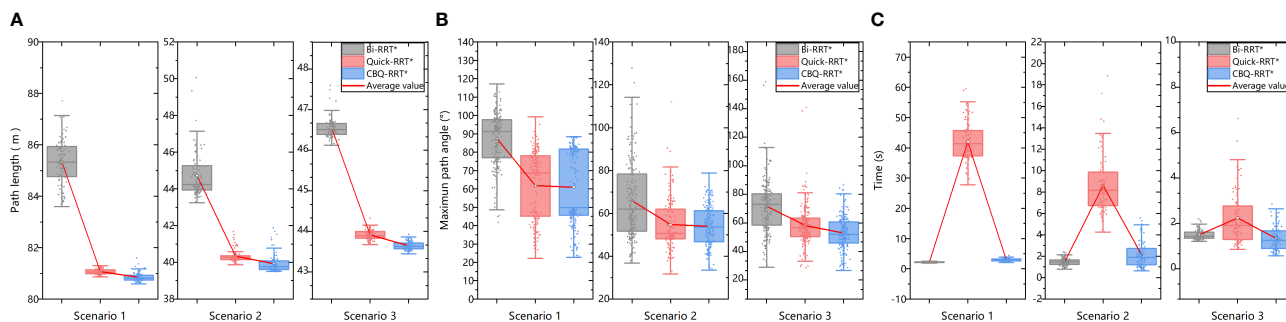


FIGURE 10 Statistics of simulation results. (A) The path length statistics results; (B) the maximum path angle statistics results; (C) the planning time statistics results.

TABLE 2 Path length statistics of simulation results.

Algorithm	Scenario 1	Scenario 2	Scenario 3
CBQ-RRT*	80.84 m	39.93 m	43.65 m
Quick-RRT*	81.06 m (+0.2%)	40.33 m (+1.0%)	43.92 m (+0.6%)
Bi-RRT*	85.33 m (+5.3%)	44.71 m (+10.9%)	46.55 m (+6.0%)

Finally, regarding algorithmic efficiency, Figure 10C and Table 4 show that Bi-RRT* and CBQ-RRT* algorithms have shorter path initialization times than the Quick-RRT* algorithm, particularly in Scenario 1, as characterized by the smaller feasible region and a more complex environment. This emphasizes the effectiveness of bidirectional RRT algorithms, including CBQ-RRT*, in meeting the efficiency needs of path planning for mobile robots in orchard settings.

The results confirm that the CBQ-RRT* algorithm is superior to Bi-RRT* and Quick-RRT* algorithms in terms of overall performance and demonstrates its ability to efficiently plan shorter collision-free paths that align better with the kinematic constraints of mobile robots in a shorter time.

3.2 Planned path tracking experiment

3.2.1 Experiment setting

The algorithmic performance of CBQ-RRT* was further evaluated in an orchard context using real-world experiments. The physical orchard mobile robot used is shown in Figure 11, and its motion parameters are listed in Table 5. The experimental environment consisted of an orchard with complex obstacles, as shown in Figure 12A. The mobile robot, equipped for picking, incorporated a three-dimensional LiDAR “Robosensen_16” for environmental sensing and a nine-axis IMU module to monitor its movement. From the resulting sensor data, an offline map is constructed using the cartographer algorithm proposed by Hess et al. (2016) to facilitate path planning. This map, shown in Figure 12B, consists of 712×659 pixels at a resolution of 0.05 m/pixel.

This map was utilized to plan paths using the Bi-RRT*, Quick-RRT*, and CBQ-RRT* algorithms, each originating at [6 m, 2 m, 0°] and terminating at [22.73 m, 1 m, 90°], over five trials per algorithm. The experiments assessed the executability of each path planning algorithm by comparing the lateral deviation error between the planned and actual robot trajectory. The results of these on-field experiments are represented in Figures 13, 14; Supplementary

TABLE 4 Planning time statistics of simulation results.

Algorithm	Scenario 1	Scenario 2	Scenario 3
CBQ-RRT*	2.96 s	2.14 s	1.33 s
Quick-RRT*	41.97 s	8.86 s	2.24 s
Bi-RRT*	2.15 s	1.47 s	1.49 s

TABLE 3 Maximum path angle of simulation results.

Algorithm	Scenario 1	Scenario 2	Scenario 3
CBQ-RRT*	60.89°	53.93°	52.89°
Quick-RRT*	61.74° (+1.4%)	54.78° (+1.6%)	58.26° (+1.0%)
Bi-RRT*	74.33° (+41.44%)	60.46° (+20.4%)	71.61° (+22.9%)

Figure 1 statistics regarding the maximum path deviation are summarized in Table 6.

3.2.2 Analysis of experiment results

Figure 13 shows the results of the paths planned by the Bi-RRT*, Quick-RRT*, and CBQ-RRT* algorithms in a real environment map. These results are consistent with those obtained in the simulation experiments, with the CBQ-RRT* algorithm demonstrating the ability to generate planned paths of relatively high quality using fewer extended nodes.

The real-time trajectory of the robot was monitored during path execution using the localization module of the Cartographer algorithm. Supplementary Figure 1 shows the paths planned by the three algorithms, represented by red line segments, along with the actual trajectories of the mobile robot during navigation, represented by blue curves. The planned and actual trajectories are length normalized to quantify the executability of the path during the navigation process. We select n observation points from them in equal proportions and calculate the lateral error of the mobile robot during the path-following process as Equation (5).

$$Lateral_err(i) = \sqrt{[x(i)_p - x(i)_r]^2 + [y(i)_p - y(i)_r]^2} \quad (5)$$

where $x(i)_p$, $y(i)_p$ and $x(i)_r$, $y(i)_r$ are the coordinates of the i th observation point in the planned path and actual trajectory, respectively. Figure 14 shows the lateral error between the actual and planned trajectories when the mobile robot executes the paths

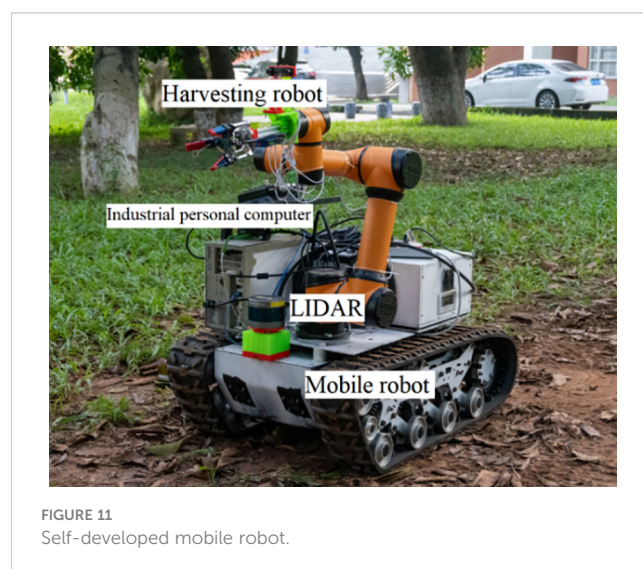


FIGURE 11
Self-developed mobile robot.

TABLE 5 The motion parameters of the mobile robot.

Parameter	Value
Minimum turning radius R_{min}	0.6 m
Maximum curvature k_{max}	1.6 m^{-1}
Maximum angular velocity ω_{max}	$20^\circ/\text{s}$
Maximum speed v_{c_max}	2 m/s

planned by these algorithms. [Supplementary Figure 2](#) illustrates the movement of a mobile robot when executing a planned path. The mobile robot exhibits comparatively smaller lateral errors in [Figure 14](#); [Supplementary Figures 1, 2](#) when executing the paths

planned by the CBQ-RRT* algorithm, suggesting superior executability of the paths planned by the CBQ-RRT* algorithm.

[Table 6](#) presents the maximum path lateral error statistics obtained from these experiments. CBQ-RRT* recorded an average maximum path lateral error of 0.334 m across five trials. In contrast, Quick-RRT* and Bi-RRT* recorded lateral errors of 0.455 and 0.796 m, respectively. The field experiments validated the superior performance of the CBQ-RRT* algorithm in challenging orchards.

4 Conclusions

When operating autonomously, orchard mobile robots, such as those used for weeding, spraying, and picking, must easily traverse

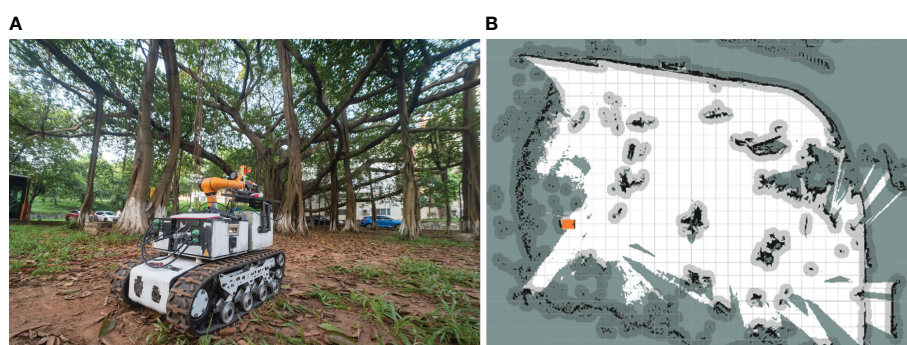


FIGURE 12 Experimental environment. (A) Experimental field scenario; (B) grid map of the experimental scenario.

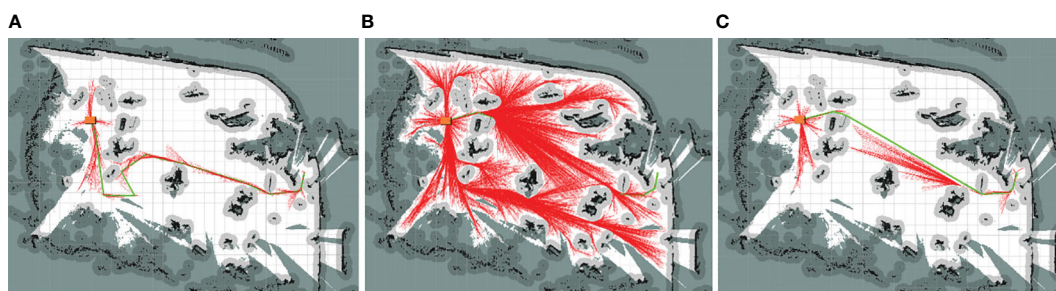


FIGURE 13 Planned paths result. (A) Bi-RRT*; (B) Quick-RRT*; (C) CBQ-RRT*.

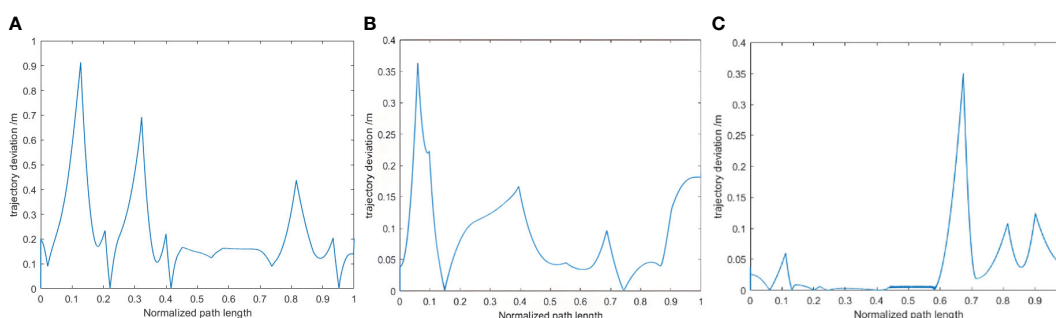


FIGURE 14 The lateral error records. (A) Bi-RRT*; (B) Quick-RRT*; (C) CBQ-RRT*.

TABLE 6 Maximum lateral error statistics of the path.

Sequence	Bi-RRT*	Quik-RRT*	CBQ-RRT*
1	0.77 m	0.60 m	0.32 m
2	0.52 m	0.40 m	0.36 m
3	0.92 m	0.37 m	0.42 m
4	1.05 m	0.46 m	0.31 m
5	0.72 m	0.45 m	0.26 m

through the orchard. Existing research on the navigation scheme of orchard mobile robots focuses on path extraction and fixed path following in a structured orchard. However, this type of research cannot be applied to unstructured orchard scenes where the road characteristics are fuzzy, and the robot needs to autonomously plan the navigation path according to environmental perception. Therefore, an efficient path planning algorithm is key to achieving improved automation for mobile orchard robots in unstructured orchard environments.

In this study, based on the Quick-RRT* algorithm, a bidirectional expansion method was introduced, and a computationally lightweight path cost function was proposed to maintain the smoothness of the paths between nodes and the efficiency of node expansion during random tree expansion. Moreover, the algorithm performs smoothness optimization during dual-tree link processing, which locally improves the kinematics of the paths at the dual-tree link. We prove the effectiveness of the CBQ-RRT* algorithm for the path planning problem of working mobile robots in an orchard scene by setting three types of orchard simulation maps: a standardized planting scene, an irregular road scene, and an irregular planting scene. The statistical simulation results of the experiment show that, compared with the Bi-RRT* algorithm, the CBQ-RRT* algorithm improves the length and smoothness of the planned paths under the premise of using a similar path planning time, and the planning efficiency is significantly improved compared with the Quick-RRT* algorithm. Path-following experiments were conducted in a real field environment to prove further the effectiveness of the CBQ-RRT* algorithm in practical work-robot applications. The experimental results show that the mobile robot can follow the path planned by the CBQ-RRT* algorithm more accurately. The experimental results show that the CBQ-RRT* algorithm can effectively plan a safe and feasible point-to-point navigation path for a mobile robot in a complex orchard environment and the mobile robot can follow the path planned by the CBQ-RRT* algorithm more accurately. Autonomous point-to-point navigation is integral in orchard robotics, particularly spraying, weeding, and inspection robots. These robots also share structural similarities with the tracked

chassis used in our experiment, leading us to believe that CBQ-RRT* can be directly applied to enhance operational efficiency.

Data availability statement

The original contributions presented in the study are included in the article/[Supplementary Material](#). Further inquiries can be directed to the corresponding author.

Author contributions

LY: Conceptualization, Methodology, Software, Writing – original draft. JL: Resources, Writing – review & editing. PL: Data curation, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research was supported by the the National Natural Science Foundation of China (52105268), Shaoguan Science and Technology Project (210725144530830 and 230403118030785), and the Key Research Project of Shaoguan University (SZ2023KJ14 and SZ2022KJ13).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fpls.2024.1337638/full#supplementary-material>

References

Blok, P. M., van Boheemen, K., van Evert, F. K., Ijsselmuiden, J., and Kim, G.-H. (2019). Robot navigation in orchards with localization based on Particle filter and

Kalman filter. *Comput. Electron. Agric.* 157, 261–269. doi: 10.1016/j.compag.2018.12.046

- Buchanan, R., Wellhausen, L., Bjelonic, M., Bandyopadhyay, T., Kottege, N., and Hutter, M. (2021). Perceptive whole-body planning for multilegged robots in confined spaces. *J. Field Robot.* 38, 68–84. doi: 10.1002/rob.21974
- Castro, G. G. D., Berger, G. S., Cantieri, A., Teixeira, M., Lima, J., Pereira, A. I., et al. (2023). Adaptive path planning for fusing rapidly exploring random trees and deep reinforcement learning in an agriculture dynamic environment UAVs. *Agriculture* 13, 354. doi: 10.3390/agriculture13020354
- Chen, M., Chen, Z., Luo, L., Tang, Y., Cheng, J., Wei, H., et al. (2024). Dynamic visual servo control methods for continuous operation of a fruit harvesting robot working throughout an orchard. *Comput. Electron. Agric.* 219, 108774. doi: 10.1016/j.compag.2024.108774
- Chen, M., Tang, Y., Zou, X., Huang, Z., Zhou, H., and Chen, S. (2021). 3D global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and SLAM. *Comput. Electron. Agric.* 187, 106237. doi: 10.1016/j.compag.2021.106237
- Elbanhawi, M., Simic, M., and Jazar, R. N. (2015). Continuous path smoothing for car-like robots using B-spline curves. *J. Intell. Robot. Syst.* 80, 23–56. doi: 10.1007/s10846-014-0172-0
- Ferguson, D., and Stentz, A. (2006). Using interpolation to improve path planning: The Field D* algorithm. *J. Field Robot.* 23, 79–101. doi: 10.1002/rob.20109
- Gammell, J. D., Srinivasa, S. S., and Barfoot, T. D. (2014). Informed RRT: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. *2014 IEEE/RSJ Int. Conf. Intell. Robots Syst.* 2997–3004. doi: 10.1109/IROS.2014.6942976
- Gan, Y., Zhang, B., Ke, C., Zhu, X., He, W., and Ihara, T. (2021). Research on robot motion planning based on RRT algorithm with nonholonomic constraints. *Neural Process. Lett.* 53, 3011–3029. doi: 10.1007/s11063-021-10536-4
- Hess, W., Kohler, D., Rapp, H., and Andor, D. (2016). “Real-time loop closure in 2D LIDAR SLAM,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, pp. 1271–1278. doi: 10.1109/ICRA.2016.7487258
- Huan, Z., Wang, J., Zhu, L., Zhong, Z., Ma, W., and Chen, Z. (2023). Navigation and closed-loop control of magnetic microrobot in plant vein mimic environment. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1133944
- Jeong, I.-B., Lee, S.-J., and Kim, J.-H. (2019). Quick-RRT*: Triangular inequality-based implementation of RRT* with improved initial solution and convergence rate. *Expert Syst. Appl.* 123, 82–90. doi: 10.1016/j.eswa.2019.01.032
- Jiang, S., Wang, S., Yi, Z., Zhang, M., and Lv, X. (2022). Autonomous navigation system of greenhouse mobile robot based on 3D Lidar and 2D Lidar SLAM. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.815218
- Karaman, S., and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* 30, 846–894. doi: 10.1177/0278364911406761
- Khajepour, A., Sheikhmohammady, M., and Nikbakhsh, E. (2020). Field path planning using capacitated arc routing problem. *Comput. Electron. Agric.* 173, 105401. doi: 10.1016/j.compag.2020.105401
- Kuffner, J. J., and LaValle, S. M. (2000). “RRT-connect: An efficient approach to single-query path planning,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, San Francisco, CA, USA, 2, 995–1001. doi: 10.1109/ROBOT.2000.844730
- LaValle, S. M., Kuffner, J. J., and Donald, B. R. (2001). Rapidly-exploring random trees: Progress and prospects. *Algorithmic Comput. robotics: New Dir.* 5, 293–308.
- Li, J., and Yang, C. (2020). “AUV path planning based on improved RRT and Bezier curve optimization,” in *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, Beijing, China, pp. 1359–1364. doi: 10.1109/ICMA49215.2020.9233842
- Li, Y., Wei, W., Gao, Y., Wang, D., and Fan, Z. (2020). PQ-RRT*: An improved path planning algorithm for mobile robots. *Expert Syst. Appl.* 152, 113425. doi: 10.1016/j.eswa.2020.113425
- Liu, H., Zhang, X., Wen, J., Wang, R., and Chen, X. (2019). Goal-biased bidirectional RRT based on curve-smoothing. *IFAC-PapersOnLine* 52, 255–260. doi: 10.1016/j.ifacol.2019.12.417
- Liu, T., Kang, H., and Chen, C. (2023). ORB-Livox: A real-time dynamic system for fruit detection and localization. *Comput. Electron. Agric.* 209, 107834. doi: 10.1016/j.compag.2023.107834
- Ljungqvist, O., Evestedt, N., Axehill, D., Cirillo, M., and Pettersson, H. (2019). A path planning and path-following control framework for a general 2-trailer with a car-like tractor. *J. Field Robot.* 36, 1345–1377. doi: 10.1002/rob.21908
- Moon, C.-B., and Chung, W. (2014). Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree. *IEEE Trans. Ind. Electron.* 62, 1080–1090. doi: 10.1109/TIE.2014.2345351
- Nasir, J., Islam, F., and Ayaz, Y. (2013). Adaptive rapidly-exploring-random-tree-star (RRT*)-smart: algorithm characteristics and behavior analysis in complex environments. *Asia Pac. J. Inf. Tech. Multimed.* 2, 39–51. doi: 10.17576/apjtm
- Qiu, Q., and Li, X. (2022). “LiDAR point-cloud odometer based mobile robot routine tracking in orchards,” in *2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, Baishan, China, pp. 735–739. doi: 10.1109/CYBER55403.2022.9907082
- Reid, W., Fitch, R., Göktoğan, A. H., and Sukkarieh, S. (2020). Sampling-based hierarchical motion planning for a reconfigurable wheel-on-leg planetary analogue exploration rover. *J. Field Robot.* 37, 786–811. doi: 10.1002/rob.21894
- Ren, G., Lin, T., Ying, Y., Chowdhary, G., and Ting, K. C. (2020). Agricultural robotics research applicable to poultry production: A review. *Comput. Electron. Agric.* 169, 105216. doi: 10.1016/j.compag.2020.105216
- Siciliano, B., Khatib, O., and Kröger, T. (2008). *Springer handbook of robotics* Vol. 200 (Berlin: Springer).
- Strader, J., Otsu, K., and Agha-mohammadi, A. A. (2020). Perception-aware autonomous motion planning for planetary exploration rovers. *J. Field Robot.* 37, 812–829. doi: 10.1002/rob.21925
- Tahir, Z., Qureshi, A. H., Ayaz, Y., and Nawaz, R. (2018). Potentially guided bidirectional RRT* for fast optimal path planning in cluttered environments. *Robot. Auton. Syst.* 108, 13–27. doi: 10.1016/j.robot.2018.06.013
- Tian, H., Mo, Z., Ma, C., Xiao, J., Jia, R., Lan, Y., et al. (2023). Design and validation of a multi-objective waypoint planning algorithm for UAV spraying in orchards based on improved ant colony algorithm. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1101828
- Wang, J., Li, B., and Meng, M. Q.-H. (2021). Kinematic Constrained Bi-directional RRT with Efficient Branch Pruning for robot path planning. *Expert Syst. Appl.* 170, 114541. doi: 10.1016/j.eswa.2020.114541
- Webb, D. J., and Van Den Berg, J. (2013). “Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics,” in *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, pp. 5054–5061. doi: 10.1109/ICRA.2013.6631299
- Wencheng, W., Gege, Z., Xinlin, C., and Weixian, W. (2020). Research on path planning of orchard spraying robot based on improved RRT algorithm. *Proc. 2020 2nd Int. Conf. Big Data Artif. Intell.* 311–316. doi: 10.1145/3436286.3436412
- Xiong, Y., Ge, Y., Liang, Y., and Blackmore, S. (2017). Development of a prototype robot and fast path-planning algorithm for static laser weeding. *Comput. Electron. Agric.* 142, 494–503. doi: 10.1016/j.compag.2017.11.023
- Xiong, J., Liang, J., Zhuang, Y., Hong, D., Zheng, Z., Liao, S., et al. (2023). Real-time localization and 3D semantic map reconstruction for unstructured citrus orchards. *Comput. Electron. Agric.* 213, 108217. doi: 10.1016/j.compag.2023.108217
- Yang, Z., Ouyang, L., Zhang, Z., Duan, J., Yu, J., and Wang, H. (2022). Visual navigation path extraction of orchard hard pavement based on scanning method and neural network. *Comput. Electron. Agric.* 197, 106964. doi: 10.1016/j.compag.2022.106964
- Ye, L., Duan, J., Yang, Z., Zou, X., Chen, M., and Zhang, S. (2021). Collision-free motion planning for the litchi-picking robot. *Comput. Electron. Agric.* 185, 106151. doi: 10.1016/j.compag.2021.106151
- Ye, L., Wu, F., Zou, X., and Li, J. (2023). Path planning for mobile robots in unstructured orchard environments: An improved kinematically constrained bidirectional RRT approach. *Comput. Electron. Agric.* 215, 108453. doi: 10.1016/j.compag.2023.108453
- Zhang, W., Zeng, Y., Wang, S., Wang, T., Li, H., Fei, K., et al. (2023). Research on the local path planning of an orchard mowing robot based on an elliptic repulsion scope boundary constraint potential field method. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1184352
- Zhang, X., Guo, Y., Yang, J., Li, D., Wang, Y., and Zhao, R. (2022). Many-objective evolutionary algorithm based agricultural mobile robot route planning. *Comput. Electron. Agric.* 200, 107274. doi: 10.1016/j.compag.2022.107274
- Zhao, Y. M., Lin, Y., Xi, F., and Guo, S. (2014). Calibration-based iterative learning control for path tracking of industrial robots. *IEEE Trans. Ind. Electron.* 62, 2921–2929. doi: 10.1109/TIE.2014.2364800