IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 27, 2019, accepted July 10, 2019, date of publication July 15, 2019, date of current version August 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2928846

Bidirectional Potential Guided RRT* for Motion Planning

WANG XINYU[1,2], LI XIAOJUAN[1,2], GUAN YONG[1,3], SONG JIADONG[1,4], AND WANG RUI[1,5]

[1] Information Engineering College, Capital Normal University, Beijing 100048, China
[2] Beijing Engineering Research Center of High Reliable Embedded System
[3] Beijing Advanced Innovation Center for Imaging Theory and Technology
[4] Machinery Industry Information Center, Beijing 100823, China
[5] Beijing Key Laboratory of Light Industrial Robots and Safety Verification

Corresponding authors: Wang Xinyu (18810371250@163.com) and Li Xiaojuan (lixj@cnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61472468, Grant 61572331, Grant 61602325, and Grant 61702348, in part by the National Key Research and Development Plan under Grant 2017YFC0806700, in part by the Project of the Beijing Municipal Science and Technology Commission under Grant LJ201607, in part by the Capacity Building for Sci-Tech Innovation–Fundamental Scientific Research Funds under Grant 025195305000, and in part by the Capital Normal University Major (key) Nurturing Project.

ABSTRACT Requirement for high accuracy and speed of grasping operation for motion planning is very important. Motion planning algorithms for avoiding obstacles in narrow channels play a vital role for robotic arm effectively operating grasp tasks. The potential function-based RRT*-connect (P-RRT*-connect) algorithm for motion planning is presented by combining the bidirectional artificial potential field into the rapidly exploring random tree star (RRT*) in order to enhance the performance of the RRT*. The motion path is found out by exploring two path trees from the start node and destination node, respectively, with the rapidly exploring random tree star. Two trees advance each other at the same time according to the attractive potential field and the repulsion potential field generated by the artificial potential field method of sampled nodes until they meet. The P-RRT*-connect algorithm is especially suitable for solving the problem of narrow channels. The simulation results prove that the P-RRT*-connect algorithm is more efficient than potential Function-based RRT* (P-RRT*) regardless of the number of iterations or the running time. The experimental data show that the time for the P-RRT*-connect to find the optimal path from the starting node to the target node is half than that of the P-RRT*, and the number of iterations of the P-RRT*-connect is also about one-third less than that of the P-RRT* which is useful for real time.

INDEX TERMS Potential function based RRT*-connect, RRT*, artificial potential field, two path trees, narrow channels.

I. INTRODUCTION

Robot grasping operation is an indispensable part in the process of robot performing tasks. A common grasping task is that the robot obtains the precise pose of the target object with machine vision, then moves the robotic arm (Figure 1) to the target position. The end-effector of robot arm grasps the target object stably and reliably in case of avoiding obstacles, and moves the object to another target position. When the robotic arm performs the grasping task, motion planning plays a crucial role, so it is necessary to study an efficient motion planning algorithm.

The associate editor coordinating the review of this manuscript and approving it for publication was Ludovico Minati.

Robot motion planning is to find a collision-free path between a given initial state and a target state, under the condition that the motion is constrained. After many years of development, motion planning has matured, and a series of planning methods have been derived, which are mainly divided into the following categories: graph-based search methods, artificial potential field-based methods, random sampling-based methods, intelligent optimization methods and many other different planning methods. The graph-based search methods are not suitable for motion planning of multi-degree-of-freedom robotic arms. When the environment is more complicated, the graph-based search methods are more complicated for the obstacle area. Therefore, this method can no longer be tested in a narrow channel environment. The method based on intelligent optimization has the problem

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

FIGURE 1. **Robotic arm in lab.**

of low computational efficiency. This paper only discusses the artificial potential field-based method and the random sampling-based method.

The *Artificial Potential Field* [1] method was first proposed by O. Katib in 1986. The basic idea is to construct the artificial potential field under the influence of the attractive potential field at the target position and the repulsive potential field around the obstacle, and then search the descending direction of the potential function to find the collision free path, so that the robotic arm will bypass the obstacle under the action of these two forces and move from the starting point to the target point. This method has been widely used because of its simple structure, high computational efficiency and real-time control. However, its disadvantage is that it is easy to fall into a local minimum point [2], where the resultant force of the robot is 0 and the target pose cannot be reached. To solve this problem, many scholars have proposed improvements, such as introducing virtual target points [3] or making robots try to move randomly [4], using the simulated annealing algorithm [5], using the adding extra control force method [6], applying the genetic algorithm into artificial potential field method [7], introducing the gain factor [8], applying a virtual obstacle concept [9] or virtual obstacle method [10], [11] to get rid of the current minimum. Jinseok Lee proposed an internal state model to solve the local minimum problem with computational cost which is low relatively [12]. Zhang Tao et al. proposed an improved wall-following approach and path memory [13]. Anugrah K et al. proposed the vector potential function [14]. Ya-Chun Chang et al. combined the *Artificial Potential Field* method with Voronoi diagram method to improve the moving quality of mobile robots [15]. Rahman proposed to plan a constant path for each agent, thus avoiding changing the trajectory [16]. Qinzhao Wang proposed a method of gravity field rotation and virtual obstacle filling [17]. Chen JinXin introduced a repulsion deflection model [18].

The motion planner based on random sampling method firstly conducts random sampling in the free configuration space of robot, then the connection graph is formed by sampling points, and then the collision-free path is obtained by graph searching. This method mainly includes *Probabilistic Roadmap Method* (*PRM*) [19] and *Rapidly Exploring Random Tree* (*RRT*) [20]. In 2000, [21] proposed the $RRT-connect$ algorithm to greatly increase the node expansion efficiency. In 2001, [22] proposed the concept of Bidirectional-*RRT*, and constructed a spanning tree search path from both ends to achieve the convergence of the algorithm. In 2002, E.Frazzoli proved that the *RRT* algorithm decreases exponentially with the exponential rate as the number of nodes increases [23]. In 2003, C. Urmson and R. Simmons proposed a heuristic algorithm to guide the expansion tree to the target region [24]. In 2006, D. Ferguson and A. Stentz proposed to run the *RRT* algorithm multiple times to gradually improve the quality of the solution [25]. In 2006, PENG studied a distance algorithm to reduce the sensitivity of the distance function to the environment during the *RRT* extension tree [26]. By this time, the *RRT* algorithm had been applied to robots and other fields [27]–[30]. In 2010, Karaman and Frazzoli first proposed the $RRT*$ algorithm to solve the problem that the probability of *RRT* algorithm is not optimal [20]. In 2015, Ahmed Hussain Qureshi proposed the $IB-RRT*$ algorithm, which quickly converges to the optimal solution through the heuristic function of intelligent sample insertion [31]. In 2018, Meng Li proposed a new node acceptance criterion for path planning of *RRT* algorithm in 3D environment [32]. In 2018, Byungchul proposed an adaptive step size *RRT* planning algorithm [33]. In 2019, Cao applied genetic algorithm and smooth processing to *RRT* algorithm [34]. In 2019, Cai Wentao proposed $I-RRT$ to improve the RRT algorithm by introducing target probability bias and step size control [35]. In 2019, Zhang Yakun proposed $MSB-RRT$, using multi-sampling methods to make *RRT* extensions target-oriented [36].

In order to solve the problems that the *Artificial Potential Field* method is easy to fall into the local minimum value and the $RRT*$ algorithm has many iterations and the long running time, Ahmed Hussain Qureshi et al. combined the two methods. *Potential Guided Directional* $RRT*(PGD-RRT*)$ [37], *Adaptive Potential Guided Directional* $RRT*(APGD-RRT*)$ [38] and *Potential Function Based* $RRT*(P-RRT*)$ [39] have been proposed successively. Among them, $P-RRT*$ is an extension of $PGD-RRT*$ and $APGD-RRT*$, which not only solves the situation that *Artificial Potential Field* method is prone to fall into local minimum. Moreover, it inherits the advantages of $RRT*$ to achieve asymptotic optimization, and reduces the time and the number of iterations to find the optimal path, thus reducing the memory utilization.

Both experiments and analysis prove that the $P-RRT*$ algorithm has the following characteristics: 1) has the same asymptotic computational complexity as $RRT*$; 2) inherits the asymptotic optimality from $RRT*$; 3) solves the local minimum problem; 4) can converge to the optimal path solution faster than $RRT*$; 5) reduce the number of iterations and time required to calculate a more optimized solution compared to $RRT*$, thereby using less memory.

The work of this paper is to further improve the $P-RRT*$ algorithm by introducing the idea of greed algorithm and puting forward the method of two-sides extension tree. $P-RRT*-connect$ not only solves the non-optimalism problem for $RRT*$ of complete probability, but also overcomes the situation that *APF* is liable to fall into local minimum. Because the algorithm constructs an extended tree from both the starting point and the target point, the algorithm can provide more explicit orientation than the other algorithms. So it is especially suitable for narrow channel environments. Based on the smooth arrival of the target point, the algorithm can also reduce the time and number of iterations for searching for the optimal path, thereby reducing memory utilization and improving the efficiency of the algorithm.

The remaining work of this paper is as follows: the second part introduces the related work which respectively introduces the idea of the *Rapidly Exploring Random Tree Star* and its advantages and disadvantages, and the idea of *Artificial Potential Field* method and its advantages and disadvantages; The third part introduces $P-RRT*-connect$, which is developed from $P-RRT*$ algorithm, and describes its basic idea and implementation process in detail. In the fourth part, the experiments prove that $P-RRT*-connect$ is indeed more efficient by comparing $P-RRT*$ with $P-RRT*-connect$ under the same experimental environment. The fifth part is the summary and prospect of the future work.

II. RELATED WORK

A. RRT*

The *Rapidly Exploring Random Tree Star*($RRT*$) is an improved algorithm of the *Rapidly Exploring Random Tree*(RRT). The basic idea of RRT is that taking an initial point as the root node and generating a random extension tree by randomly sampling and increasing leaf nodes. When leaf nodes in the random tree contain the target point or enter the target region, a path from the initial point to the target point can be found in the random tree. The idea of $RRT*$ is comparing the path cost by building a set of surrounding nodes near the new node on the basis of RRT, that is, walking through these surrounding nodes to check whether there is a better path, and if there is, to replace the existing path with this better path, so as to improve the existing search tree.

Ahmed Hussain Qureshi improved the original $RRT*$ algorithm to improve the efficiency of the algorithm by reducing the number of collision detection process. The following is the pseudo-code of the improved $RRT*$ algorithm:

The $RRT*$ algorithm finds an optimal path between start point and target point through collision detection. In this process, since the dynamic system, the algorithm will generate an input u:$[0,T] \in U$, the total consumption time is $t \in [0,T]$. $X \in R^d$ is the configuration space, X_{obs} is the obstacle area, U mentioned above is the input space, X_{free} =X/X_{obs} is the collision-free area. We define $T=(V,E)$ which aims to find path in the process of structure of the generated tree, $V \subset X$ is the vertex of the tree and E is the edge of the tree. The implementation process of $RRT*$ mainly consists of the following seven parts:

Algorithm 1 $RRT*$

$V \leftarrow start\_node; E \leftarrow \emptyset; T \leftarrow (V,E)$;
for $ind = 0 \rightarrow N$ **do**
  $new\_node \leftarrow sample(ind)$;
  $near\_node \leftarrow near\_node(new\_node, T)$;
  if $near\_node = \emptyset$ **then**
    $near\_node \leftarrow nearest\_node(new\_node, T)$;
  end if
  $L \leftarrow insert\_node(new\_node, near\_node)$;
  $min\_node \leftarrow chooseparent(L)$;
  if $min\_node = \emptyset$ **then**
    $T \leftarrow insert\_vertex(new\_node, min\_node, T)$;
    $T \leftarrow rewire(new\_node, L, E)$;
  end if
end for
return $T=(V,E)$;

Sampling: randomly select the sampling points in the free area with the sampling function, and assign the sampling points to the random variable $new\_node \in X_{free}$.

Near node: the function *near_node* of the adjacent node provides the node $near\_node \in V$ in the spherical region formed with r as the radius, where the radius r is expressed as follows:

$$r = \gamma(\frac{\log n}{n})^{\frac{1}{d}} \tag{1}$$

where γ is an independent constant, d is the dimension of the configuration space, and n is the number of vertices.

Distance: the improved algorithm adds the path cost, so the value returned by the distance function is the path cost between two nodes, namely the Euclidean distance.

Nearest nodes: this function returns the nearest nodes in the constructed extended tree T, which are computed from random states based on Euclidean distances.

Collision checking: the collision checking function checks whether each connected path will collide with an obstacle in the process of constructing the extended tree. If the collision occurs, the node will be abandoned. If no collision occurs, this node is added to the tree.

Listing and sorting: the element *near_node* in the set of adjacent nodes is ordered in ascending order by the cost function $c()$ under the function *insert_node* to produce a list L, where each element in the list L is composed of three variable elements($near\_node'$,c,ε).

Extending: this function enables the extension tree to find a smoother optimal path.

In view of the disadvantage which is complete but not optimal of RRT algorithm, $RRT*$ algorithm has been able to achieve progressive optimization, but at the same time, some new problems have emerged in $RRT*$:

1) $RRT*$ can achieve progressive optimization, but the implementation process is slow and takes a long time.
2) in order to achieve progressive optimization, $RRT*$ needs to use a large number of iterative processes and wastes resources;
3) in the process of searching for the optimal solution, $RRT*$ has certain limitations in cost comparison. It cannot guarantee that all the selected nodes are the most efficient, that is, some useful nodes may be neglected.

B. APF

In 1986, Oussama Khatib first proposed the *Artificial Potential Field* method (*APF*) for the motion planning of robots. As the name suggests, the *Artificial Potential Field* method is to artificially introduce the potential field. In the playground, the potential field does not really exist, but is assumed by people to describe the force of the target point and the obstacle on the robot. Where, the force of the target point on the robot is attractive, and the robot moves towards the target point under the attractive force. However, the obstacle exerts a repulsive force on the robot, which acts on the robot to avoid the obstacle. The robot moves under the combined force of these two forces. The forces that the robot receives at the obstacle and the target point is shown in Figure 2. G represents the target point and O represents the obstacle.The farther the robot is from the target, the more attractive force it is to the target, and vice versa. The closer the robot is to the obstacle, the greater the repulsive force will be, and vice versa. When both the attractive and repulsive forces on the robot are zero, we assume that the robot has reached the target point. However, when the resultant force of attractive and repulsive forces on the robot is zero, the robot also thinks it has reached the target point, which is a typical shortcoming of the *Artificial Potential Field* method that it is easy to fall into the local minimum.

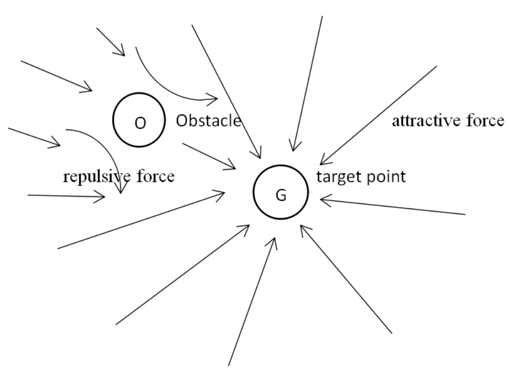


FIGURE 2. Schematic diagram of the *Artificial Potential Field* method.

The potential is generated by the *Artificial Potential Field*. According to the definition of the potential, the gradient descent method is generally adopted to define the attractive force generated by the attractive potential field on the mobile robot as the negative gradient of the attractive potential field, and the repulsive force generated by the repulsion force potential field on the mobile robot as the negative gradient of the repulsion force potential field. The force exerted on the electric charge in the electric field becomes the electric field force, whose magnitude and direction can be calculated by coulomb's law. When there are multiple charges acting simultaneously in the electric field, the magnitude and direction of the electric field force follow the vector operation rules. The potential field force is defined by the potential field model. The magnitude and direction of the potential field force received by the mobile robot also follow the vector operation rules. According to this principle, the target produces an attractive force on the mobile robot, and the obstacle produces a repulsive force to the mobile robot. The repulsive force generated by the obstacle and the attractive force generated by the target position are superimposed, and the resultant force is the total force of the mobile robot in the *Artificial Potential Field*. The robot moves under the action of the total potential force to generate a planned path.

Artificial Potential Field method is often used to solve local programming problems, but it can also solve global programming problems. The idea of the algorithm is based on the virtual potential field. The algorithm definition is intuitive and the model structure is simple. The planning process can avoid obstacles and complete the planning tasks in real time without a large amount of calculation. Since the trajectory of the robot is the final planning path, the planning path must be safe and smooth. Therefore, this algorithm is widely used in real time obstacle avoidance and smooth trajectory control.

However, the classical *Artificial Potential Field* method has two limitations in the long-term practical application, which limits the application scenarios of the planning algorithm and affects the planning efficiency of the *Artificial Potential Field* method. 1) The problem of target point cannot be reached, that is, when the robot travels to a position which is close to the target point, there is one or more obstacles around the target point, causing the robot to repeatedly squat near the target point and cannot continue the path planning. 2) The local minimum value problem, that is, a plurality of obstacles in the environment are distributed at a specific position, so that there are some local extreme value regions, and the mobile robot cannot leave the local region to continue the path planning.

C. P-RRT*

The basic idea of $P-RRT*$ is to obtain the random sample x_{rand} in free space. When the robot approaches the target node, the attractive force generated by the target node becomes larger and the negative gradient of the attraction force decreases. Under the action of the attractive potential field, a new node, x_{prand}, is generated and then connects this node as a new sample to the extension tree. In the process of extension, the $P-RRT*$ algorithm compares whether the path of the current node from the starting point is the optimal path, and if it is the optimal path, retains the current node;

if the path formed by the current node is not the optimal path, the $P-RRT*$ algorithm will find the new node again until it finds the optimal path from the starting point to the target point.

Although the $P-RRT*$ algorithm has improved many deficiencies, the algorithm is an one-way search. In order to improve the search speed in free space, next section proposes a greedy strategy applied to the $P-RRT*$ algorithm, and proposes two trees. Improving the search efficiency of the algorithm by letting both trees expand at the same time. Two trees, T_1 and T_2, are established. These two trees serve as heuristic guidance search for each other until the two trees meet. The specific implementation process is as follows.

III. P-RRT*-CONNECT

A. BASIC IDEA

Combining the characteristics of APF and $RRT*$, the path planning is mainly carried out by using $RRT*$ algorithm, which is assisted by APF algorithm. When entering the narrow channel environment, it is assisted by $RRT*$ algorithm to escape from the local minimum region in the environment. In the working space of the mobile robot, the mobile robot gradually approaches the target point due to the attractive force traction of the target point, and when it approaches the obstacle, it is repulsive and avoids the obstacle in real time. When there are specific obstacles and these obstacles cause the balance between the attractive force and repulsive forces on the mobile robot, the algorithm will detect that the mobile robot falls into a local minimum value. At this time, the mobile robot will switch to $RRT*$ algorithm for path planning. The obstacles to be avoided are around the mobile robot, which conforms to the applicable conditions of $RRT*$ algorithm. After random sampling for a certain distance, the mobile robot escapes from the local minimum and switches back to the APF path planning algorithm until it reaches the target point.

In order to avoid the local minimum problem caused by target orientation, the guidance factor is added to the node growth function of $RRT*$ algorithm. The idea of target attractive force of APF was introduced into the search tree expansion stage of $RRT*$ algorithm, and the attractive force factor based on the target state point was constructed. The trend expansion tree was biased towards target growth to reduce invalid search volume and improve expansion efficiency, and the probability of target deviation was small to avoid local minimum. In view of the problem that $RRT*$ algorithm lacks stability and has slow convergence speed, this paper uses the attractive potential field thought of traditional APF algorithm for reference to improve $RRT*$ algorithm, so that mobile robot can not only make normal planning, but also avoid falling into the local minimum region, and its expansion process is biased towards the target point to accelerate the convergence speed.

Next, we will introduce how to use the *Artificial Potential Field* method to guide the $RRT*$ algorithm to bias the sampling point to the target point.

U_{att} and U_{rep} are used to respectively represent the constructed attractive field and repulsive field. F_{att} and F_{rep} are used to respectively represent the attractive force and repulsive force by the robot. Using the gradient descent method, the magnitude and direction of the attractive force and the repulsive force are calculated under the condition of the known attractive force field and repulsive force field. $W = [X\ Y]^T$ is to represent the coordinates of the position of the robot in the two-dimensional motion space. The expression of the total potential field when the robot moves to a certain place is shown in formula (2):

$$U(W) = U_{att}(W) + U_{rep}(W) \tag{2}$$

On the basis of formula (2), the magnitude and direction of resultant force received by the robot here can be obtained. Negative gradients of two potential field quantities in formula (2), U_{att} and U_{rep}, can be calculated respectively, and the magnitude and direction of attractive force and repulsion force received by the robot in two-dimensional space can be obtained. That is:

$$\vec{F} = \vec{F_{att}} + \vec{F_{rep}} \tag{3}$$
$$\vec{F_{att}} = -grad[U_{att}(W)] \tag{4}$$
$$\vec{F_{rep}} = -grad[U_{rep}(W)] \tag{5}$$

When moving to a certain place, the attractive potential field of the robot is expressed as:

$$U_{att}(W) = \frac{1}{2}k\|W - W_g\| \tag{6}$$

where, k is the attractive gain coefficient of the target point on the robot, W is the coordinate position of the current movement of the robot, W_g is the coordinate of the target point, and $\|W - W_g\|$ is the distance between the current position of the robot and the target point. The attractive force expression is:

$$\vec{F_{att}} = -k\|W - W_g\| \tag{7}$$

The repulsive potential field of the robot at this point is:

$$U_{rep}(W) = \frac{1}{2}m(\frac{1}{\|W - W_0\|} - \rho_0), \quad \|W - W_0\| \leq \rho_0 \tag{8}$$
$$U_{rep}(W) = 0, \quad \|W - W_0\| > \rho_0 \tag{9}$$

where, m is the gain coefficient of the obstacle's repulsive force on the robot, and ρ_0 is the influence range of the obstacle's repulsive force. The repulsion force is expressed as:

$$\vec{F_{rep}} = m(\frac{1}{\|W - W_0\|})\frac{1}{\|W - W_0\|^2}, \quad \|W - W_0\| \leq \rho_0 \tag{10}$$
$$\vec{F_{rep}} = 0, \quad \|W - W_0\| > \rho_0 \tag{11}$$

Under the action of the attractive potential field and the repulsive potential field, the sampling function can only be sampled in the scope, and these sampling points gradually converge near the target point. The optimal path formed by the $RRT*$ algorithm approaches the target point.

Algorithm 2 $P-RRT*-Connect$

$V_1 \leftarrow X_{init}$;$E_1 \leftarrow \emptyset$;$T_1 \leftarrow (V_1, E_1)$;
$V_2 \leftarrow X_{goal}$;$E_2 \leftarrow \emptyset$;$T_2 \leftarrow (V_2, E_2)$;
while $i < N$ **do**
  $x_{rand} \leftarrow RandomSample(i)$;
  $x_{prand1} \leftarrow RGD(x_{rand})$;
  $X_{near1} \leftarrow NearbyNodes(T_1, x_{prand1}, n)$;
  if $X_{near1} \neq \emptyset$ **then**
    $X_{near1} \leftarrow NearestNode(x_{prand1}, T_1 = (V_1, E_1))$;
  end if
  $L_1 \leftarrow GetTuple(x_{prand1}, X_{near1})$;
  $X_{parent1} \leftarrow SelectBestParent(L_1)$;
  if $X_{parent1} \neq \emptyset$ **then**
    $T_1 = (V_1, E_1) \leftarrow InsertNode(x_{prand1}, x_{parent1}, T_1 = (V_1, E_1))$;
    $E_1 \leftarrow RewireNodes(x_{prand1}, L_1, E_1)$;
  end if
  $x_{prand2} \leftarrow RGD(x_{rand})$;
  $X_{near2} \leftarrow NearbyNodes(T_2, x_{prand2}, n)$;
  if $X_{near2} \neq \emptyset$ **then**
    $X_{near2} \leftarrow NearestNode(x_{prand2}, T_2 = (V_2, E_2))$;
  end if
  $L_2 \leftarrow GetTuple(x_{prand2}, X_{near2})$;
  $X_{parent2} \leftarrow SelectBestParent(L_2)$;
  if $X_{parent2} \neq \emptyset$ **then**
    $T_2 = (V_2, E_2) \leftarrow InsertNode(x_{prand2}, x_{parent2}, T_2 = (V_2, E_2))$;
    $E_2 \leftarrow RewireNodes(x_{prand2}, L_2, E_2)$;
  end if
  while $X_{near1} \neq X_{near2}$ **do**
    $X_{near3} \leftarrow NearbyNodes(T_2, x_{near2})$;
    if $X_{near3} \neq \emptyset$ **then**
      $X_{near3} \leftarrow NearestNode(x_{prand2}, T_2 = (V_2, E_2))$;
    end if
    $L_1 \leftarrow GetTuple(x_{prand2}, X_{near2})$;
    $X_{parent2} \leftarrow SelectBestParent(L_2)$;
    if $X_{parent2} \neq \emptyset$ **then**
      $T_2 = (V_2, E_2) \leftarrow InsertNode(x_{prand2}, x_{parent2}, T_2 = (V_2, E_2))$;
      $E_2 \leftarrow RewireNodes(x_{prand2}, L_2, E_2)$;
    end if
    if $X_{near1} \neq X_{near2}$ **then**
      return $T_1 = (V_1, E_1)$;
    end if
  end while
end while

B. P-RRT*-CONNECT

Before we introduce the implementation, we define some of the variables used in this article. The new algorithm is improved on the basis of *Potential Function Based* $-$ *RRT*$*$ ($P-RRT*$), so we named it as *Potential Function Based RRT* $*-$*connect* ($P-RRT*-connect$). The starting node and target node are respectively defined as $x_{init} \in X_{free}$ and $x_{goal} \in X_{free}$. $T_1 = (V_1,E_1)$ and $T_2 = (V_2,E_2)$ are the two trees generated from the starting node and target node in the process of finding the optimal path, V_1 and V_2 are the vertices of the extension tree, and E_1 and E_2 are the edges. A random sample $x_{rand} \in X_{free}$ is obtained by random sampling function. In the direction of gradient descent of the attractive potential field, it is separated from x_{rand} by a small distance from the walking length $\lambda \in R_+$, generating a new random sample $x_{prand} \in X_{free}$. According to the characteristics of *Artificial Potential Field* method, the closer the random sample is to the target region, the smaller the descending gradient of the attractive potential field will be.

Set the starting node x_{init} and the target node x_{goal}, and use the starting node as the root node of T_1 and the target node as the root node of T_2. Firstly, T_1 is extended, and the sampling point function *RandomSample* is called to obtain x_{rand}. In order to obtain a higher quality sampling point, x_{rand} is processed and x_{prand} is obtained under the action of *RGD* function and attractive potential field. *RGD*, or the *Randomized Gradient Descent Planning*, is used as a function of the previous state to perform iteration until the potential field difference approaches zero and calculate the next state. x_{prand}, the enlarged sampling point, is taken as a new random sample to conduct node expansion on this basis. The nearest x_{prand} node is obtained by using the *NearestNode* function, and the nearest parent node is found by using the nearest ordered sequence L. If there is no collision with obstacles and this node is not empty, the node is added to the tree and the second tree is expanded at the same time. T_2 takes the target node as the root node, and just like the node expansion process of T_1, new nodes are obtained. Until the nodes of the two trees meet, stop expanding, connect the two trees, and switch the order of the nodes of T_2, that's the path we're looking for.

Figure 3 shows that the simultaneous expansion of two different extension trees. The white circles represent the starting node and the target node respectively. The black circles represent the leaf node collected by the algorithm. The red circle represents the leaf node that the two trees meet in the process of expansion. The black line represents the edge of T_1, and the blue line represents the edge of T_2 which also represent the steps of the two trees. The pseudocode for the specific implementation process is shown as following.

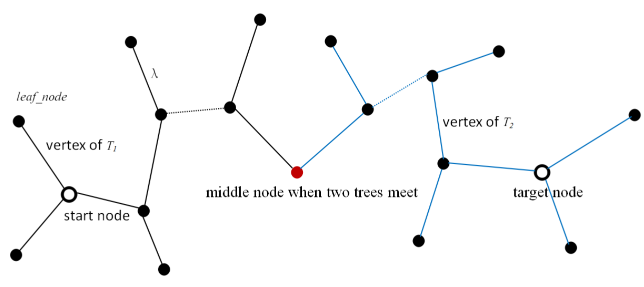
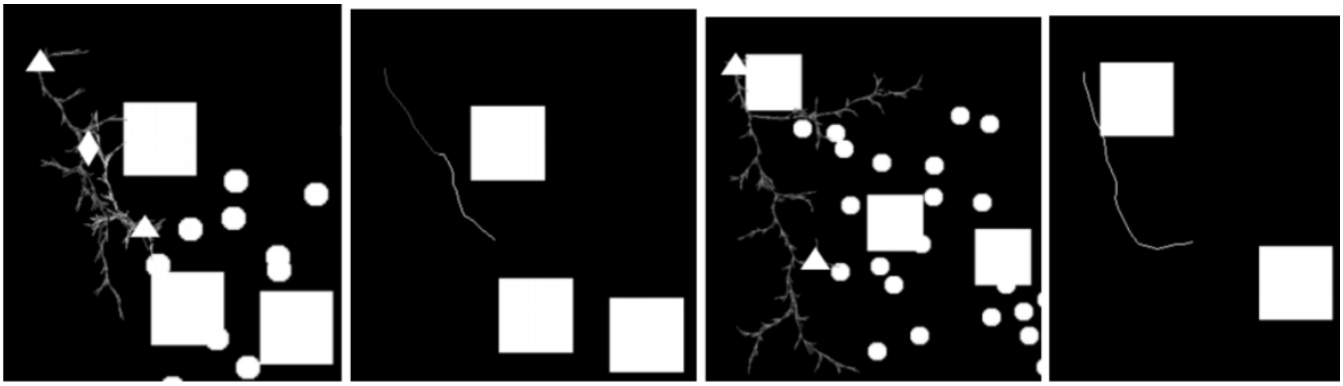
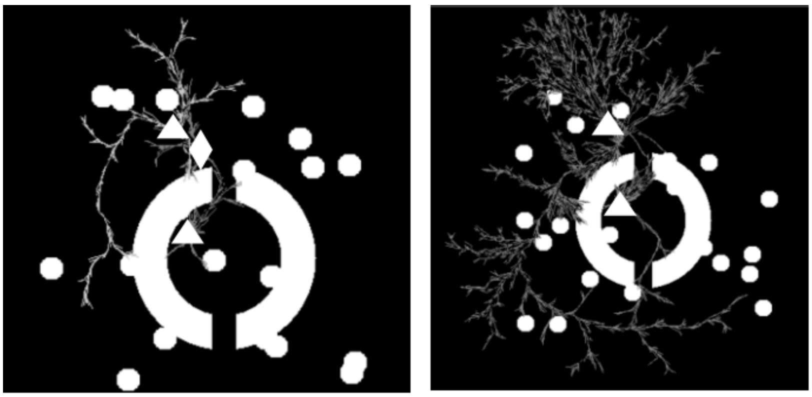


FIGURE 3. Extension process of two trees.



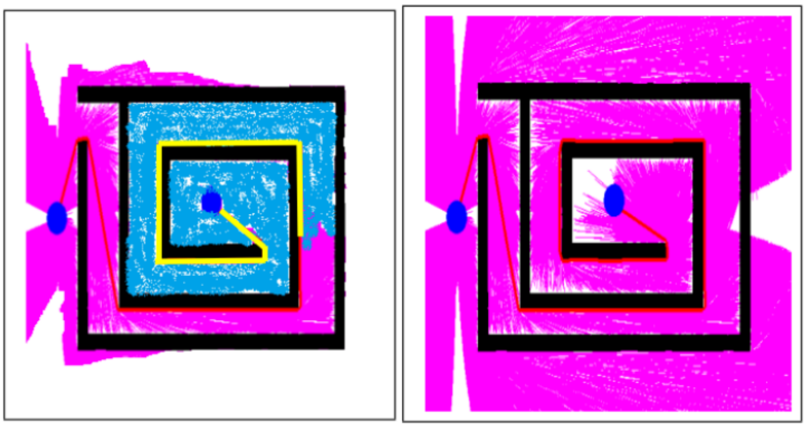
4.a 4.b 4.c 4.d

FIGURE 4. Implementation of $P-RRT*-connect$(4.a 4.b) and $P-RRT*$(4.c 4.d) algorithms under the same conditions.



5.a 5.b

FIGURE 5. Realization process of $P-RRT*-connect$(5.a) and $P-RRT*$(5.b) under narrow channel.



6.a 6.b

FIGURE 6. Realization process of $P-RRT*-connect$(6.a) and $P-RRT*$(6.b) in 2D environment.

IV. EXPERIMENT AND COMPARATIVE ANALYSIS

This experiment was run in the environment of VMware Workstations Pro+Ubuntu 16.04. By comparing $P-RRT*$ algorithm with $P-RRT*-connect$ algorithm in the same experimental environment, it is verified that $P-RRT*-connect$ algorithm can find the optimal path more quickly.

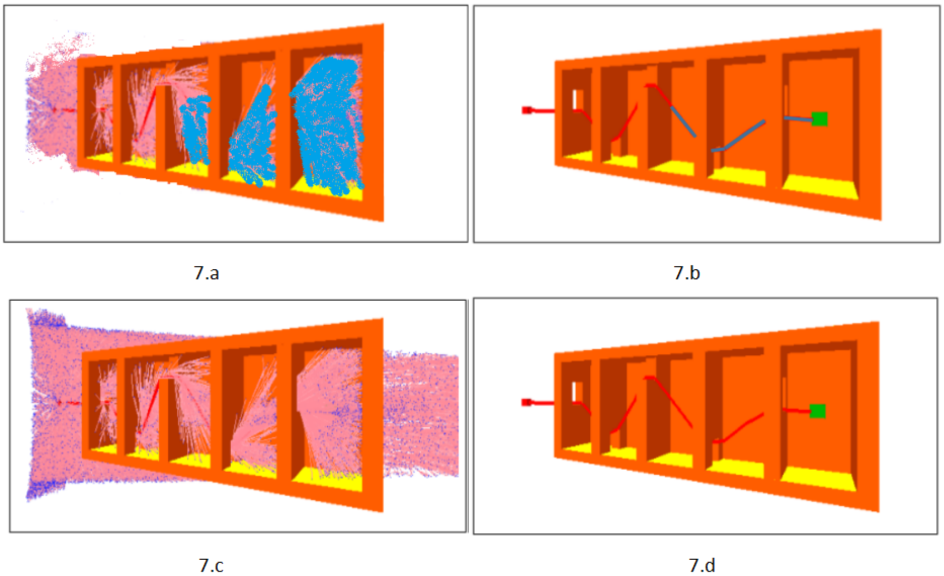


FIGURE 7. Realization process of $P-RRT*-connect$ (7.a 7.b) and $P-RRT*$ (7.c 7.d) in 3D environment with multiple barriers.

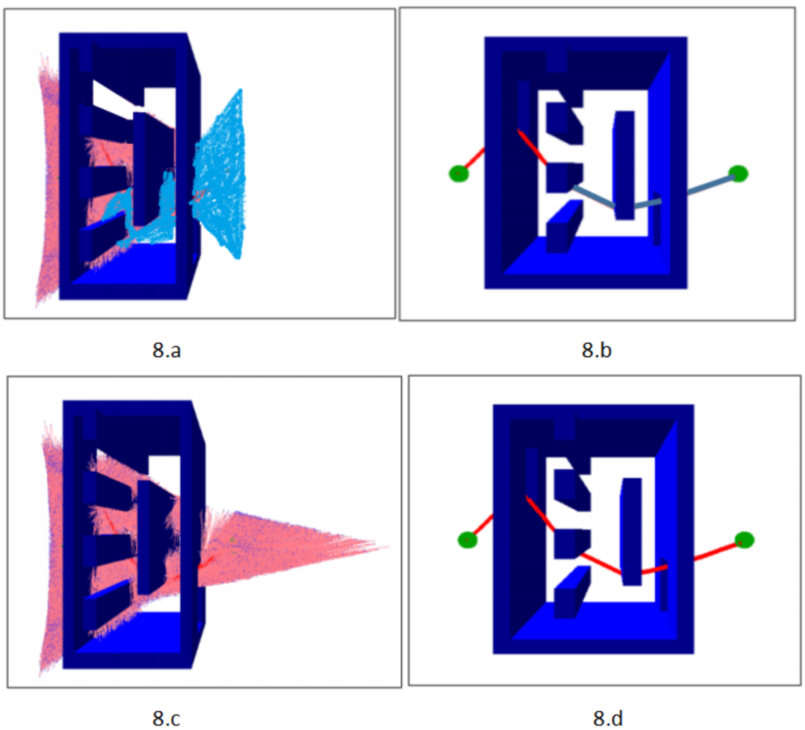


FIGURE 8. Realization process of $P-RRT*-connect$ (8.a 8.b) and $P-RRT*$ (8.c 8.d) in 3D environment with multiple narrow channel.

We will compare the efficiency of the two algorithms in terms of time and iteration times. Time t is the time required to complete the path search, that is, the number of seconds to complete all iterations. The number of iterations is set to n. In order to make the experimental results reliable and convincing, the experimental parameters and environment configuration space of the two algorithms are the same. Figure 4 shows the process of $P-RRT*-connect$ algorithm (figure 4.a) and $P-RRT*$ algorithm (figure 4.c) searching for the optimal path under the same experimental parameters and the configuration space size of the environment, that is, the process of extension tree extension. In Figure 4.a,the gray

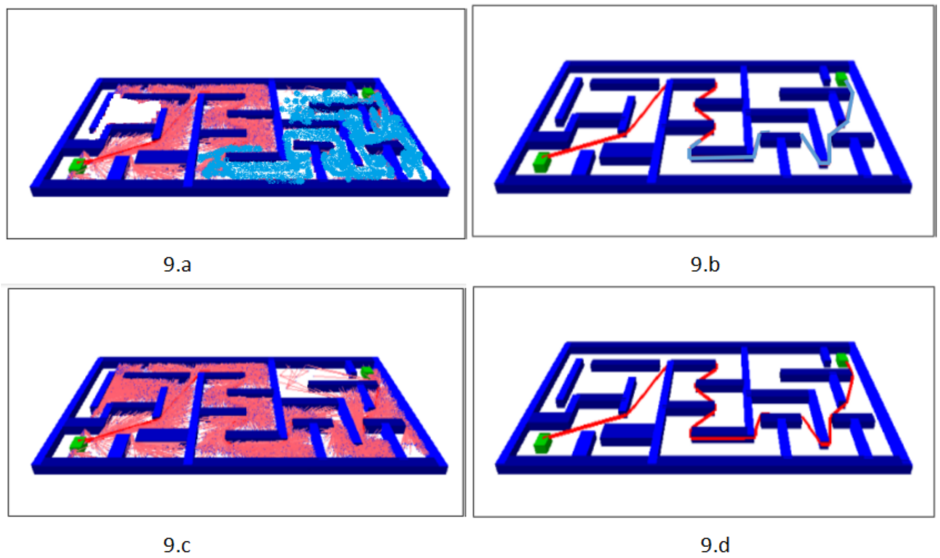


FIGURE 9. Realization process of $P - RRT * -connect$(9.a 9.b) and $P - RRT*$(9.c 9.d) in 3D maze environment.

part indicates the tree T_1 extended from the starting node to the target node, the white part represents the tree T_2 that extends from the target node to the starting node; Figure 4.c and figure 4.d respectively represent the optimal path which is found. All the pictures, we only intercepted the key parts. The box in the figure represents the obstacle, the circle represents the potential field of distribution, the triangles represent the starting node and the target node, and the diamond represents the node where the two trees meet. As we can see from figure 4, the number of iterations for $P - RRT*$ algorithm to complete this path planning is $n = 723$, and the elapsed time is $t = 4.01$s. The $P - RRT * -connect$ algorithm is implemented with fewer iterations $n = 402$, shorter time $t = 1.48$s, and smoother path.

Figure 5 is a simulation experiment conducted in the narrow channel environment. According to the iteration times $n = 1053$ and consumption time $t = 1.25$s of $P - RRT * -connect$ algorithm, the iteration times n =5739 and consumption time $t = 6.37$s of $P - RRT*$ algorithm, it can be seen that the efficiency of $P - RRT * -connect$ algorithm is also higher than that of $P - RRT*$ algorithm in the local minimum environment.

Figure 6 shows the performance tests of $P - RRT * -connect$ and $P - RRT*$ in 2D maze environment. In figure 6.a, the pink area and the blue area respectively represent the extension tree generated by $P-RRT*-connect$ algorithm sampling from both the starting point and the target point. In figure 6.b, the pink area represents the extension tree generated by $P - RRT*$ algorithm. The explanation also applies to Figure 7, Figure 8, and Figure 9. It can be seen intuitively from the two graphs that $P - RRT * -connect$ can find the optimal path from the starting point to the target point with reduced number of iterations, thus achieving higher efficiency than $P - RRT*$. Figures 7, 8 and 9 represent the

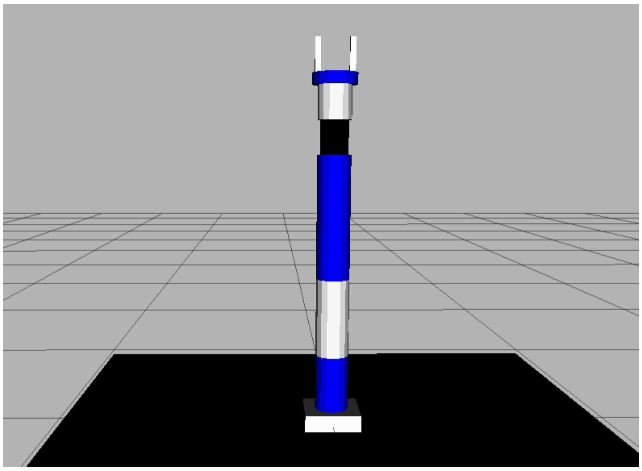


FIGURE 10. 6-dof robotic arm.

performance tests of the two algorithms in the 3D environment with multiple barriers, the 3D environment with narrow passages and the 3D maze environment, respectively. The specific results are presented in Table 1. Figure 7.b shows the optimal path found by the $P - RRT * -connect$ algorithm. The two lines with different colors represent the paths formed from the starting point and the target point. Figure 7.d shows the optimal path found by the $P - RRT*$ algorithm. Figure 8 and figure 9 are similar to figure 7.

Due to the large randomness of sample selection, we conducted 50 experiments, recorded the maximum number of iterations n_{max}, minimum number of iterations n_{min}, maximum consumption time t_{max} and minimum consumption time t_{min}, and calculated the mean value of n_{avg} and t_{avg}. See table 1. We limit the number of nodes to 5 million to reduce time overconsumption and define the distance to the long range $\lambda = 0.1$.

TABLE 1. Comparison of the two algorithms.

environments	algorithm	n_{max}	n_{min}	n_{avg}	t_{max}	t_{min}	t_{avg}
2D-Maze (Figure 6)	$P-RRT*-connect$	103728	101217	102645	15.9	15.3	15.7
	$P-RRT*$	152782	150686	151178	29.2	28.9	29.1
3D-Multiple barriers (Figure 7)	$P-RRT*-connect$	73089	69935	71748	9.3	8.5	8.9
	$P-RRT*$	91827	84528	87496	17.6	16.3	16.8
3D-Narrow channel (Figure 8)	$P-RRT*-connect$	33756	31879	32067	5.1	4.8	4.5
	$P-RRT*$	48726	46419	47981	9.48	8.92	9.2
3D-Maze (Figure 9)	$P-RRT*-connect$	39470	36956	38320	4.6	4.46	4.51
	$P-RRT*$	43861	41380	42931	8.27	8.03	8.12

n is the number of iterations and t is the running time.

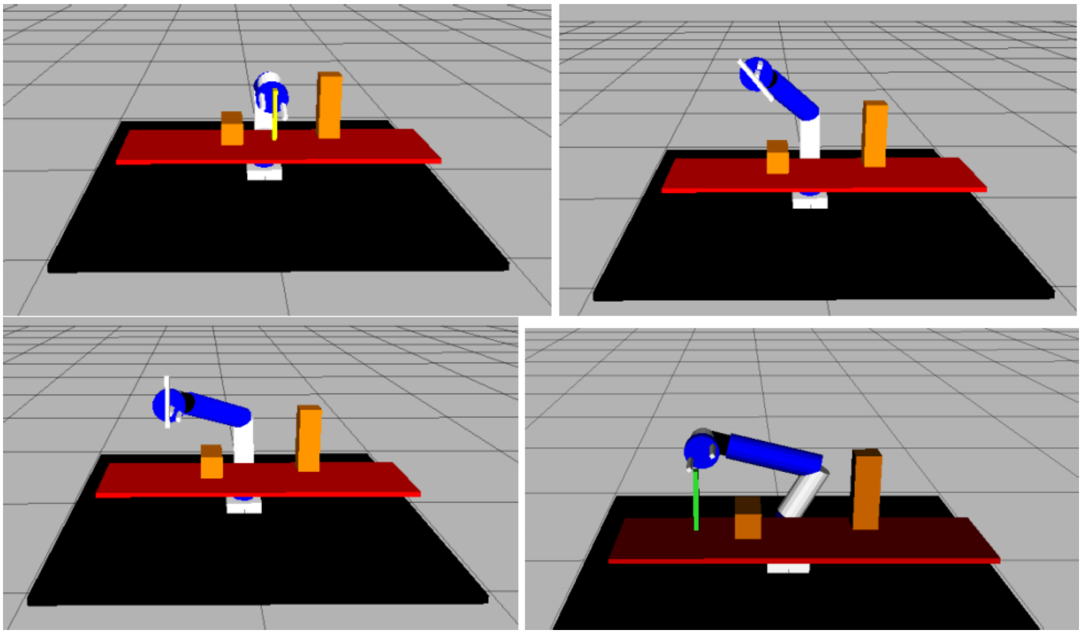


FIGURE 11. The process of robot arm grasping the object.

It can be seen from table 1 that the efficiency of $P-RRT*-connect$ algorithm in finding the starting node to the final node is higher than that of $P-RRT*$, both in terms of iteration times and time consumption. Therefore, it is proved that the new improved algorithm is very feasible and necessary.

In order to verify whether the $P-RRT*-connect$ algorithm is suitable for robotic arm grasping, we simulated a six-dof robotic arm in a virtual environment with two grippers as shown in figure 10. We built the simulation environment of the robotic arm based on *ROS* and displayed it in *Rviz*.

There are four objects: The red part is the table, the yellow object is the grasped target, and the other two are obstacles. The grasping operation we want to experiment with is to let the robotic arm grasp the target in the workspace and place the target in the specified position without touching the obstacles. When the target is placed in the specified position, it will automatically turn green. Several snapshots of the process of moving the target from the middle of the table to the edge of the table are shown in figure 11.

V. CONCLUSION AND FUTURE WORK

With the continuous development of robotics and technology, motion planning is not only applicable to robots, but also widely used in other fields, such as modern industry, surgical robot and many other fields. Therefore, it is necessary to study an efficient motion planning algorithm. In this paper, a simple and effective randomization algorithm, $P-RRT*-connect$, is proposed to solve the single-path query problem of $P-RRT*$. Based on the combination of $RRT*$ and APF, a two-sides extension tree is proposed. Experimental results show that this algorithm can achieve good performance in different environments, and its efficiency is higher than $P-RRT*$ both in iterations and running time. The main advantages of $P-RRT*-connect$ are as follows: 1)it can find a non-collision optimal path from the starting node to the target node with fewer iterations, which reduces the memory utilization; 2) running time to find optimal path is greatly reduced due to the simultaneous expansion from both ends; 3) this algorithm is especially suitable for the narrow channel environment, without falling into the local minimum. This algorithm can also be used to grasp objects accurately and quickly in a 6-dof robotic arm in a virtual environment. In our future work, we will apply $P-RRT*-connect$ algorithm to different practical environments to verify.

REFERENCES

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[2] Y. Koren, J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, Apr. 1991, pp. 1398–1404.

[3] W. Ji, J. Le Li, D. A. Zhao, and Y. Jun, "Obstacle avoidance method of apple harvesting robot manipulator," *Trans. Chin. Soc. Agricult. Mach.*, vol. 44, no. 11, pp. 253–259, Jan. 2013.

[4] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 2, pp. 224–241, Mar. 1992.

[5] Z. Peiyan and L. Tiansheng, "Soccer robot path planning based on artificial potential field approach with simulated annealing," *Mech. Sci. Technol.*, vol. 22, no. 4, pp. 547–558, 2003.

[6] Z. Jian-Ying, Z. Zhi-Ping, and L. Dun, "A path planning method for mobile robot based on artificial potential field," *J. Harbin Inst. Technol.*, vol. 38, no. 8, pp. 1306–1308, 2006.

[7] L. Qing, W. Lijun, C. Bo, Z. Zhou, and Y. Yixin, "An improved artificial potential field method with parameters optimization based on genetic algorithm," *J. Univ. Sci. Technol. Beijing*, vol. 34, no. 2, pp. 202–206, 2012.

[8] Z. Jian-Ying and L. Tun, "Optimized path planning of mobile robot based on artificial potential field," *Acta Aeronaut. Astronaut. Sinica*, vol. 28, no. 2, pp. 183–188, 2007.

[9] M. C. Lee and M. G. Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2003, pp. 735–740.

[10] M. G. Park, M. C. Lee, and K. Son, "Real-time path planning in unknown environments using a virtual hill," in *Proc. IFAC World Congr.*, 2004, pp. 2223–2228.

[11] A. Wahid, M. H. Mabrouk, and C. R. McInnes, "Potential field based navigation for planetary rovers using internal states," in *Proc. 59th Int. Astron. Congr.*, 2008, p. 14.

[12] J. Lee, Y. Nam, S. Hong, and W. Cho, "New potential functions with random force algorithms using potential field method," *J. Intell. Robot. Syst.*, vol. 66, no. 3, pp. 303–319, 2012.

[13] T. Zhang, Y. Zhu, and J. Song, "Real-time motion planning for mobile robots by means of artificial potential field method in unknown environment," *Ind. Robot, Int. J.*, vol. 37, no. 4, pp. 384–400, 2010.

[14] A. K. Pamosoaji and K.-S. Hong, "A path-planning algorithm using vector potential functions in triangular regions," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 4, pp. 832–842, Jul. 2013.

[15] Y.-C. Chang and Y. Yamamoto, "Path planning of wheeled mobile robot with simultaneous free space locating capability," *Intell. Service Robot.*, vol. 2, no. 1, pp. 9–22, 2009.

[16] A. Rahman and A. K. Azad, "To escape local minimum problem for multi-agent path planning using improved artificial potential field-based regression search method," in *Proc. Int. Conf. Inf. Technol.*, 2017, pp. 371–376.

[17] Q. Wang, J. Cheng, and X. Li, "Path planning of robot based on improved artificial potentional field method," in *Proc. Int. Conf. Artif. Intell., Automat. Control Technol.*, 2017, p. 34.

[18] C. Jinxin, D. Jiao, and L. Zhong, "Robot path planning based on improved artificial potential field method," in *Proc. Global Intell. Ind. Conf. Expo. Global Innov. Technol. Exchange Conf.*, 2018, pp. 1–11.

[19] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep., 1998.

[20] S. Karama and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," in *Proc. Robot. Sci. Syst.*, 2010.

[21] J. J. Kuffner and S. M. Lavalle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, Dec. 2000, p. 7.

[22] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[23] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *J. Guid., Control, Dyn.*, vol. 25, no. 1, pp. 116–129, May 2002.

[24] C. Urmson and R. Simmons, "Approaches for heuristically biasing RRT growth," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2003, pp. 1178–1183.

[25] D. Ferguson and A. Stentz, "Anytime RRTs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 5369–5375.

[26] P. Cheng, "Reducing RRT metric sensitivity for motion planning with differential constraints," Iowa State Univ., Ames, IA, USA, Tech. Rep., 2001.

[27] A. Bhatia and E. Frazzoli, "Incremental search methods for reachability analysis of continuous and hybrid systems," in *Hybrid Systems Computation Control*, vol. 2993. 2004, pp. 142–156.

[28] J. Cortes, L. Jaillet, and T. Simeon, "Molecular disassembly with RRT-like algorithms," in *Proc. IEEE Int. Conf. Robot. Automat.*, Apr. 2007, pp. 3301–3306.

[29] M. S. Branicky, M. M. Curtiss, J. Levine, and S. Morgan, "Sampling-based planning, control, and verification of hybrid systems," *IEE Proc. Control Theory Appl.*, vol. 153, no. 5, pp. 575–590, Sep. 2006.

[30] M. Zucker, J. J. Kuffner, and M. S. Branicky, "Multiple RRTs for rapid replanning in dynamic environments," in *Proc. IEEE Conf. Robot. Automat.*, 2007.

[31] A. H. Qureshi and Y. Ayaz, "Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments," *Robot. Auton. Syst.*, vol. 68, pp. 1–11, Jun. 2015.

[32] M. Li, Q. Sun, Q. Song, Z. Wang, and Y. Li, "Path planning of mobile robot based on RRT in rugged terrain," in *Proc. 2nd Int. Conf. Comput. Sci. Appl. Eng.*, 2018, p. 10.

[33] B. An, J. Kim, and F. C. Park, "An adaptive stepsize RRT planning algorithm for open-chain robots," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 312–319, Jun. 2018.

[34] X. Cao, X. Zou, C. Jia, M. Chen, and Z. Zeng, "RRT-based path planning for an intelligent litchi-picking manipulator," *Comput. Electron. Agricult.*, vol. 156, pp. 105–118, Jan. 2019.

[35] C. Wentao, D. Yi, Z. Jing, Z. Yongbo, R. Shuang, and Y. Kang, "Manipulator path planning based on improved RRT algorithm," *Transducer Microsyst. Technol.*, pp. 121–124, 2019.

[36] Z. Yakun, G. Zedong, C. Jie, and X. Yuqing, "Bidirectional RRT path planning algorithm with multiple sampling optimization," *Comput. Simul.*, pp. 319–324, 2019.

[37] A. H. Qureshi, K. F. Iqbal, S. M. Qamar, F. Islam, Y. Ayaz, and N. Muhammad, "Potential guided directional-RRT* for accelerated motion planning in cluttered environments," in *Proc. IEEE Int. Conf. Mechatronics Automat.*, Aug. 2013, pp. 519–524.

[38] A. H. Qureshi, S. Mumtaz, K. F. Iqbal, B. Ali, Y. Ayaz, F. Ahmed, M. S. Muhammad, O. Hasan, W. Y. Kim, and M. Ra, "Adaptive potential guided directional-RRT," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2013, pp. 1887–1892.

[39] A. H. Qureshi and Y. Ayaz, "Potential functions based sampling heuristic for optimal path planning," in *Autonomous Robots*. Springer, 2017, pp. 1079–1093.

WANG XINYU is currently pursuing the master's degree in computer science and technology with the College of Information Engineering, Capital Normal University, China. Her research interest includes the design of robot.

LI XIAOJUAN is currently a Professor with the College of Information Engineering, Capital Normal University, China. Her current research interests include formal verification and their applications in embedded systems.

GUAN YONG received the Ph.D. degree with the College of Mechanical Electronic and Information Engineering, China University of Mining and Technology, China, in 2004. He is currently a Professor with Capital Normal University. His research interest includes formal verification of embedded system design. He is a member of the Chinese Institute of Electronics Embedded Expert Committee.

SONG JIADONG received the M.S. degree from Capital Normal University, China, in 2011. He is currently a System Architect and the Project Manager of the Machinery Industry Information Center, China. His research interests include machine learning, pattern analysis, and system architecture analysis.

WANG RUI received the B.S. degree in computer science from Xi'an Jiaotong University, Xi'an, China, in 2004, and the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2012. She is currently an Associate Professor with the College of Information Engineering, Capital Normal University, China. Her current research interests include formal verification and their applications in embedded systems.